Cisco Nexus 9000 Series NX-OS VXLAN Configuration Guide, Release 10.1(x)

First Published: 2021-02-16

Last Modified: 2022-04-04

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 you supply the values.
[x]	Square brackets enclose an optional element (keyword or argument).
[x | y]	Square brackets enclosing keywords or arguments that are separated by a vertical bar indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments that are 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 includes 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 that 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 9000 Series Switches

The entire Cisco Nexus 9000 Series switch documentation set is available at the following URL:

http://www.cisco.com/en/US/products/ps13386/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 nexus9k-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 DevNet.
- 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 gateway to the Cisco bug-tracking system, which maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. The BST provides you with detailed defect information about your products and software.

Documentation Feedback

To provide feedback about Cisco technical documentation, use the feedback form available in the right pane of every online document.

CHAPTER 1

New and Changed Information

This chapter provides release-specific information for each new and changed feature in the *Cisco Nexus 9000 Series NX-OS VXLAN Configuration Guide Release 10.1(x)*.

New and Changed Information

Feature	Description	Changed in Release	Where Documented
Port Multi-VLAN Mapping	Introduced this feature. With this feature multiple VLANs are mapped on a trunk interface to a single global VLAN/VNI.	10.1(2)	About Port Multi-VLAN Mapping, on page 318
TRM Multisite with vPC Border Gateway	TRM Multisite support is added for vPC Border Gateway.	10.1(2)	TRM with Multi-Site, on page 240
ITD and ePBR over VXLAN	Added support for N9K-X9716D-GX TOR and N9K-C93180YC-FX3S platform switches.	10.1(1)	Guidelines and Limitations for VXLAN, on page 9
VXLAN EVPN Loop Detection and Mitigation	Added support for Cisco Nexus 9300-FX3 and -GX platform switches.	10.1(1)	Guidelines and Limitations for VXLAN EVPN Loop Detection and Mitigation, on page 211
VXLAN over Parent Interface that Carries Subinterfaces	Added support for Cisco Nexus 9300-FX3 platform switches.	10.1(1)	Guidelines and Limitations for VXLAN, on page 9
VXLAN Static Tunnels	Added support for Cisco Nexus 9300-FX3 platform switches.	10.1(1)	Guidelines and Limitations for VXLAN Static Tunnels, on page 35
Selective Q-in-VNI and VXLAN VLAN on Same Port	Added support for Cisco Nexus 9300-FX3 platform switches.	10.1(1)	Guidelines and Limitations for Q-in-VNI, on page 297

Feature	Description	Changed in Release	Where Documented
Selective Q-in-VNI and Advertise PIP on a VTEP	Added support for Cisco Nexus 9300-FX3 platform switches.	10.1(1)	Configuring Selective Q-in-VNI, on page 300
VXLAN Tunnel Encryption	Added support for Cisco Nexus 9300-FX3 platform switches.	10.1(1)	Guidelines and Limitations for Secure VXLAN EVPN Multi-Site Using CloudSec, on page 354
vPC Fabric Peering and FEX Support	Added support for Cisco Nexus 9000-EX/FX/FX2/FX3/GX platform switches.	10.1(1)	Guidelines and Limitations for vPC Fabric Peering , on page 390
VXLAN EVPN with Downstream VNI	Added support for Cisco Nexus 9300-FX3 platform switches and for N9K-C9316D-GX, N9K-C93600CD-GX, and N9K-C9364C-GX TOR switches.	10.1(1)	Guidelines and Limitations for VXLAN EVPN with Downstream VNI, on page 82
VXLAN PBR	Added support for N9K-C9316D-GX, N9K-C93600CD-GX, and N9K-C9364C-GX TOR switches.	10.1(1)	Guidelines and Limitations for VXLAN, on page 9
IPv6 Underlay	Added support for N9K-C9316D-GX, N9K-C93600CD-GX, and N9K-C9364C-GX TOR switches.	10.1(1)	Guidelines and Limitations for VXLAN with IPv6 in the Underlay (VXLANv6) , on page 131
VXLAN Flood and Learn	Added support for N9K-C9316D-GX, N9K-C93600CD-GX, and N9K-C9364C-GX TOR switches.	10.1(1)	Guidelines and Limitations for VXLAN, on page 9

CHAPTER 2

Overview

This chapter contains the following sections:

Licensing Requirements

For a complete explanation of Cisco NX-OS licensing recommendations and how to obtain and apply licenses, see the *Cisco NX-OS Licensing Guide* and the *Cisco NX-OS Licensing Options Guide*.

Supported Platforms

Starting with Cisco NX-OS release 7.0(3)I7(1), use the Nexus Switch Platform Support Matrix to know from which Cisco NX-OS releases various Cisco Nexus 9000 and 3000 switches support a selected feature.

VXLAN Overview

Virtual Extensible LAN (VXLAN) provides a way to extend Layer 2 networks across a Layer 3 infrastructure using MAC-in-UDP encapsulation and tunneling. This feature enables virtualized and multitenant data center fabric designs over a shared common physical infrastructure.

VXLAN has the following benefits:

- Flexible placement of workloads across the data center fabric.

It provides a way to extend Layer 2 segments over the underlying shared Layer 3 network infrastructure so that tenant workloads can be placed across physical pods in a single data center. Or even across several geographically divers data centers.

- Higher scalability to allow more Layer 2 segments.

  VXLAN uses a 24-bit segment ID, the VXLAN network identifier (VNID). This 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.

- Optimized utilization of available network paths in the underlying infrastructure.

  VXLAN packets are transferred through the underlying network based on their Layer 3 headers. They use equal-cost multipath (ECMP) routing and link aggregation protocols to use all available paths. In contrast, a Layer 2 network might block valid forwarding paths in order to avoid loops.

Cisco Nexus 9000 as Hardware-Based VXLAN Gateway

A Cisco Nexus 9000 Series switch can function as a hardware-based VXLAN gateway. It seamlessly connects VXLAN and VLAN segments as one forwarding domain across the Layer 3 boundary without sacrificing forwarding performance. The Cisco Nexus 9000 Series hardware-based VXLAN encapsulation and de-encapsulation provide line-rate performance for all frame sizes.

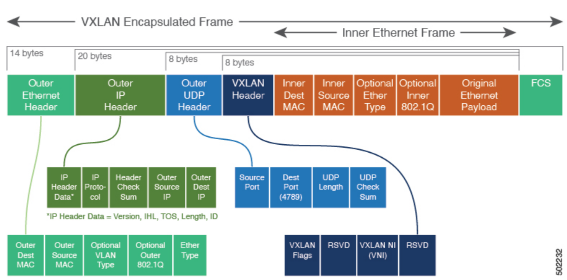
VXLAN Encapsulation and Packet Format

VXLAN is a Layer 2 overlay scheme over a Layer 3 network. It uses a MAC Address-in-User Datagram Protocol (MAC-in-UDP) encapsulation to provide a means to extend Layer 2 segments across the data center network. VXLAN is a solution to support a flexible, large-scale multitenant environment over a shared common physical infrastructure. The transport protocol over the physical data center network is IP plus UDP.

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 Layer 3 network.

VXLAN uses an 8-byte VXLAN header that consists of a 24-bit VNID and a few reserved bits. The VXLAN header, together with the original Ethernet frame, go inside the UDP payload. The 24-bit VNID is used to identify Layer 2 segments and to maintain Layer 2 isolation between the segments. With all 24 bits in the VNID, VXLAN can support 16 million LAN segments.

Figure 1:



VXLAN Tunnel

A VXLAN encapsulated communication between two devices where they encapsulate and decapsulate an inner Ethernet frame, is called a VXLAN tunnel. VXLAN tunnels are stateless since they are UDP encapsulated.

VXLAN Tunnel Endpoint

VXLAN tunnel endpoints (VTEPs) are devices that terminate VXLAN tunnels. They perform VXLAN encapsulation and de-encapsulation. Each VTEP has two interfaces. One is a Layer 2 interface on the local LAN segment to support a local endpoint communication through bridging. The other is a Layer 3 interface on the IP transport network.

The IP interface has a unique address that identifies the VTEP device in the transport network. The VTEP device uses this IP address to encapsulate Ethernet frames and transmit the packets on the transport network. A VTEP discovers other VTEP devices that share the same VNIs it has locally connected. It advertises the locally connected MAC addresses to its peers. It also learns remote MAC Address-to-VTEP mappings through its IP interface.

Underlay Network

The VXLAN segments are independent of the underlying physical network topology. Conversely, the underlying IP network, often referred to as the underlay network, is independent of the VXLAN overlay. The underlay network forwards the VXLAN encapsulated packets based on the outer IP address header. The outer IP address header has the initiating VTEP's IP interface as the source IP address and the terminating VTEP's IP interface as the destination IP address.

The primary purpose of the underlay in the VXLAN fabric is to advertise the reachability of the Virtual Tunnel Endpoints (VTEPs). The underlay also provides a fast and reliable transport for the VXLAN traffic.

Overlay Network

In broadcast terms, an overlay is a virtual network that is built on top of an underlay network infrastructure. In a VXLAN fabric, the overlay network is built of a control plane and the VXLAN tunnels. The control plane is used to advertise MAC address reachability. The VXLAN tunnels transport the Ethernet frames between the VTEPs.

Distributed Anycast Gateway

Distributed Anycast Gateway refers to the use of default gateway addressing that uses the same IP and MAC address across all the leafs that are a part of a VNI. This ensures that every leaf can function as the default gateway for the workloads directly connected to it. The distributed Anycast Gateway functionality is used to facilitate flexible workload placement, and optimal traffic forwarding across the VXLAN fabric.

Control Plane

There are two widely adopted control planes that are used with VXLAN:

Flood and Learn Multicast-Based Learning Control Plane

Cisco Nexus 9000 Series switches support the flood and learn multicast-based control plane method.

- When configuring VXLAN with a multicast based control plane, every VTEP configured with a specific VXLAN VNI joins the same multicast group. Each VNI could have its own multicast group, or several VNIs can share the same group.
- The multicast group is used to forward broadcast, unknown unicast, and multicast (BUM) traffic for a VNI.
- The multicast configuration must support Any-Source Multicast (ASM) or PIM BiDir.
- Initially, the VTEPs only learn the MAC addresses of devices that are directly connected to them.
- Remote MAC address to VTEP mappings are learned via conversational learning.

VXLAN MPBGP EVPN Control Plane

A Cisco Nexus 9000 Series switch can be configured to provide a Multiprotocol Border Gateway Protocol (MPBGP) ethernet VPN (EVPN) control plane. The control plane uses a distributed Anycast Gateway with Layer 2 and Layer 3 VXLAN overlay networks.

For a data center network, an MPBGP EVPN control plane provides:

- Flexible workload placement that is not restricted with physical topology of the data center network.
  - Place virtual machines anywhere in the data center fabric.
- Optimal East-West traffic between servers within and across data centers
  - East-West traffic between servers, or virtual machines, is achieved by most specific routing at the first hop router. First hop routing is done at the access layer. Host routes must be exchanged to ensure most specific routing to and from servers or hosts. Virtual machine (VM) mobility is supported by detecting new endpoint attachment when a new MAC address/IP address is seen directly connected to the local switch. When the local switch sees the new MAC/IP, it signals the new location to rest of the network.
- Eliminate or reduce flooding in the data center.
  - Flooding is reduced by distributing MAC reachability information via MP-BGP EVPN to optimize flooding relating to L2 unknown unicast traffic. Optimization of reducing broadcasts associated with ARP/IPv6 Neighbor solicitation is achieved by distributing the necessary information via MPBGP EVPN. The information is then cached at the access switches. Address solicitation requests can be responded locally without sending a broadcast to the rest of the fabric.
- A standards-based control plane that can be deployed independent of a specific fabric controller.
  - The MPBGP EVPN control plane approach provides:

    - IP reachability information for the tunnel endpoints associated with a segment and the hosts behind a specific tunnel endpoint.
    - Distribution of host MAC reachability to reduce/eliminate unknown unicast flooding.
    - Distribution of host IP/MAC bindings to provide local ARP suppression.
    - Host mobility.
    - A single address family (MPBGP EVPN) to distribute both L2 and L3 route reachability information.

- Segmentation of Layer 2 and Layer 3 traffic
    - Traffic segmentation is achieved with using VXLAN encapsulation, where VNI acts as segment identifier.

CHAPTER 3

Configuring VXLAN

This chapter contains the following sections:

Guidelines and Limitations for VXLAN

VXLAN has the following guidelines and limitations:

Table 1: ACL Options for VXLAN Traffic on Cisco Nexus 92300YC, 92160YC-X, 93120TX, 9332PQ, and 9348GC-FXP Switches

ACL Direction	ACL Type	VTEP Type	Port Type	Flow Direction	Traffic Type	Supported
Ingress	PACL	Ingress VTEP	L2 port	Access to Network [GROUP:encap direction]	Native L2 traffic [GROUP:inner]	YES
	VACL	Ingress VTEP	VLAN	Access to Network [GROUP:encap direction]	Native L2 traffic [GROUP:inner]	YES
Ingress	RACL	Ingress VTEP	Tenant L3 SVI	Access to Network [GROUP:encap direction]	Native L3 traffic [GROUP:inner]	YES

ACL Direction	ACL Type	VTEP Type	Port Type	Flow Direction	Traffic Type	Supported
Egress	RACL	Ingress VTEP	Uplink L3/L3-PO/SVI	Access to Network [GROUP:encap direction]	VXLAN encap [GROUP:outer]	NO
Ingress	RACL	Egress VTEP	Uplink L3/L3-PO/SVI	Network to Access [GROUP:decap direction]	VXLAN encap [GROUP:outer]	NO
Egress	PACL	Egress VTEP	L2 port	Network to Access [GROUP:decap direction]	Native L2 traffic [GROUP:inner]	NO
	VACL	Egress VTEP	VLAN	Network to Access [GROUP:decap direction]	Native L2 traffic [GROUP:inner]	NO
Egress	RACL	Egress VTEP	Tenant L3 SVI	Network to Access [GROUP:decap direction]	Post-decap L3 traffic [GROUP:inner]	YES

-
- For scale environments, the VLAN IDs related to the VRF and Layer-3 VNI (L3VNI) must be reserved with the **system vlan nve-overlay id** command.
- NLB in the unicast, multicast, and IGMP multicast modes is not supported on Cisco Nexus 9000 switch VXLAN VTEPs. The work-around is to move the NLB cluster behind the intermediary device (which supports NLB in the respective mode) and inject the cluster IP address as an external prefix into the VXLAN fabric.
- Support added for MultiAuth Change of Authorization (CoA). For more information, see the Cisco Nexus 9000 Series NX-OS Security Configuration Guide, Release 9.3(x).
- The **lacp vpc-convergence** command can be configured in VXLAN and non-VXLAN environments that have vPC port channels to hosts that support LACP.
- PIM BiDir for VXLAN underlay with and without vPC is supported.

  The following features are not supported when PIM BiDir for VXLAN underlay is configured:

  - Flood and Learn VXLAN
  - Tenant Routed Multicast (TRM)
  - VXLAN EVPN Multi-Site
  - VXLAN EVPN Multihoming
  - vPC attached VTEPs

For redundant RPs, use Phantom RP.

For transitioning from PIM ASM to PIM BiDir or from PIM BiDir to PIM ASM underlay, we recommend that you use the following example procedure:

```
no ip pim rp-address 192.0.2.100 group-list 230.1.1.0/8
clear ip mroute *
clear ip mroute date-created *
clear ip pim route *
clear ip igmp groups *
clear ip igmp snooping groups * vlan all
```

Wait for all tables to clean up.

```
ip pim rp-address 192.0.2.100 group-list 230.1.1.0/8 bidir
```

- When entering the **no feature pim** command, NVE ownership on the route is not removed so the route stays and traffic continues to flow. Aging is done by PIM. PIM does not age out entries having a VXLAN encap flag.
- Fibre Channel over Ethernet (FCoE) N-port Virtualization (NPV) can coexist with VXLAN on different fabric uplinks but on the same or different front-panel ports on Cisco Nexus 93180YC-EX and 93180YC-FX switches.

  Fibre Channel N-port Virtualization (NPV) can coexist with VXLAN on different fabric uplinks but on the same or different front-panel ports on Cisco Nexus 93180YC-FX switches. VXLAN can exist only on the Ethernet front-panel ports and not on the FC front-panel ports.
- VXLAN is supported on the Cisco Nexus 9348GC-FXP switch.
- VXLAN is not supported on the Cisco Nexus 92348GC switch.
- When SVI is enabled on a VTEP (flood and learn, or EVPN), make sure that ARP-ETHER TCAM is carved using the **hardware access-list tcam region arp-ether 256** command. This requirement does not apply to Cisco Nexus 9200, 9300-EX, 9300-FX/FX2/FX3, and 9300-GX platform switches and Cisco 9500 Series switches with 9700-EX line cards.
- For information regarding the **load-share** keyword usage for PBR with VXLAN, see the Guidelines and Limitations for Policy-Based Routing section of the *Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide, Release 9.3(x)*.
- Beginning with Cisco NX-OS Release 9.3(3), ARP suppression is supported for Cisco Nexus 9300-GX platform switches.
- Beginning with Cisco NX-OS Release 9.3(5), ARP suppression is supported with reflective relay for Cisco Nexus 9364C, 9300-EX, 9300-FX/FX2/FXP, and 9300-GX platform switches. For information on reflective relay, see the *Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide*.
- Beginning with Cisco NX-OS Release 9.3(5), the subinterfaces on VXLAN uplinks has the ability to carry non-VXLAN L3 IP traffic for Cisco Nexus 9332C, 9364C, 9300-EX, 9300-FX/FX2/FXP, and 9300-GX platform switches and Cisco Nexus 9500 platform switches with -EX/FX line cards. This feature is supported for VXLAN flood and learn and VXLAN EVPN, VXLAN EVPN Multi-Site, and DCI.
- Beginning with Cisco NX-OS Release 9.3(6), VXLAN flood and learn mode is supported for Cisco Nexus 9300-GX platform switches.
- Beginning with Cisco NX-OS Release 10.1(1), VXLAN flood and learn mode is supported for N9K-C9316D-GX, N9K-C93600CD-GX, and N9K-C9364C-GX TOR switches.

- For the Cisco Nexus 9504 and 9508 switches with -R line cards, VXLAN Layer 2 Gateway is supported on the 9636C-RX line card. VXLAN and MPLS cannot be enabled on the Cisco Nexus 9508 switch at the same time.
- For the Cisco Nexus 9504 and 9508 switches with -R line cards, if VXLAN is enabled, the Layer 2 Gateway cannot be enabled when there is any line card other than the 9636C-RX.
- For the Cisco Nexus 9504 and 9508 switches with -R line cards, PIM/ASM is supported in the underlay ports. PIM/Bidir is not supported. For more information, see the *Cisco Nexus 9000 Series NX_OS Multicast Routing Configuration Guide, Release 9.3(x)*.
- For the Cisco Nexus 9504 and 9508 switches with -R line cards, IPv6 hosts routing in the overlay is supported.
- For the Cisco Nexus 9504 and 9508 switches with -R line cards, ARP suppression is supported.
- For the Cisco Nexus 9504 and 9508 switches with -R line cards, VXLAN with ingress replication is not supported.
- Beginning with Cisco NX-OS Release 10.1(1), ITD and ePBR over VXLAN feature is supported on N9K-X9716D-GX TOR and N9K-C93180YC-FX3S platform switches.
- Beginning with Cisco NX-OS Release 10.1(1), PBR over VXLAN feature is supported on N9K-C9316D-GX, N9K-C93600CD-GX, and N9K-C9364C-GX TOR switches.
- The **load-share** keyword has been added to the Configuring a Route Policy procedure for the PBR over VXLAN feature.

  For more information, see the Cisco Nexus 9000 Series NX_OS Unicast Routing Configuration Guide, Release 9.x.
- The **lacp vpc-convergence** command is added for better convergence of Layer 2 EVPN VXLAN:

```
interface port-channel10
  switchport
  switchport mode trunk
  switchport trunk allowed vlan 1001-1200
  spanning-tree port type edge trunk
  spanning-tree bpdufilter enable
  lacp vpc-convergence
  vpc 10

interface Ethernet1/34  <- The port-channel member-port is configured with LACP-active
 mode (for example, no changes are done at the member-port level.)
  switchport
  switchport mode trunk
  switchport trunk allowed vlan 1001-1200
  channel-group 10 mode active
  no shutdown
```

- Port-VLAN with VXLAN is supported on Cisco Nexus 9300-EX and 9500 Series switches with 9700-EX line cards with the following exceptions:
  - Only Layer 2 (no routing) is supported with port-VLAN with VXLAN on these switches.
  - No inner VLAN mapping is supported.
- The **system nve ipmc** CLI command is not applicable to the Cisco 9200 and 9300-EX platform switches and Cisco 9500 platform switches with 9700-EX line cards.

- 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. This best practice should be applied not only for the vPC VXLAN deployment, but for all VXLAN deployments.
- To remove configurations from an NVE interface, we recommend manually removing each configuration rather than using the **default interface nve** command.
- **show** commands with the **internal** keyword are not supported.
- FEX ports do not support IGMP snooping on VXLAN VLANs.
- VXLAN is supported for the Cisco Nexus 93108TC-EX and 93180YC-EX switches and for Cisco Nexus 9500 Series switches with the X9732C-EX line card.
- DHCP snooping (Dynamic Host Configuration Protocol snooping) is not supported on VXLAN VLANs.
- 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.
- The QoS buffer-boost feature is not applicable for VXLAN traffic.
- The following limitations apply to releases prior to Cisco NX-OS Release 9.3(5):
  - VTEPs do not support VXLAN-encapsulated traffic over subinterfaces, regardless of VRF participation or IEEE 802.1Q encapsulation.
  - VTEPs do not support VXLAN-encapsulated traffic over parent interfaces if subinterfaces are configured, regardless of VRF participation.
  - Mixing subinterfaces for VXLAN and non-VXLAN VLANs is not supported.
- Beginning with Cisco NX-OS Release 10.1(1), VXLAN-encapsulated traffic over Parent Interface that Carries Subinterfaces is supported on Cisco Nexus 9300-FX3 platform switches.
- Beginning with Cisco NX-OS Release 9.3(5), VTEPs support VXLAN-encapsulated traffic over parent interfaces if subinterfaces are configured. This feature is supported for VXLAN flood and learn, VXLAN EVPN, VXLAN EVPN Multi-Site, and DCI. As shown in the following configuration example, VXLAN traffic is forwarded on the parent interface (eth1/1) in the default VRF, and L3 IP (non-VXLAN) traffic is forwarded on subinterfaces (eth1/1.10) in the tenant VRF.

```
interface ethernet 1/1
 description VXLAN carrying interface
 no switchport
 ip address 10.1.1.1/30

interface ethernet 1/1.10
 description NO VXLAN
 no switchport
 vrf member Tenant10
 encapsulation dot1q 10
  ip address 10.10.1.1/30
```

- Tenant VRF (VRF with VNI on it) cannot be used on an SVI that has no VNI binding into it (underlay infra VRF).
- Point-to-multipoint Layer 3 and SVI uplinks are not supported.

- SVI and subinterfaces as uplinks are not supported.
- A FEX HIF (FEX host interface port) is supported for a VLAN that is extended with VXLAN.
- In an ingress replication vPC setup, Layer 3 connectivity is needed between vPC peer devices.
- Rollback is not supported on VXLAN VLANs that are configured with the port VLAN mapping feature.
- 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 is supported on Cisco Nexus 9500 platform switches with the following line cards:
  - 9500-R
  - 9700-EX
  - 9700-FX
- Cisco Nexus 9300 Series switches with 100G uplinks only support VXLAN switching/bridging.

  Cisco Nexus 9200, Cisco Nexus 9300-EX, and Cisco Nexus 9300-FX, and Cisco Nexus 9300-FX2 platform switches do not have this restriction.

  Note For VXLAN routing support, a 40G uplink module is required.

- MDP is not supported for VXLAN configurations.
- Consistency checkers are not supported for VXLAN tables.
- ARP suppression is supported for a VNI only if the VTEP hosts the First-Hop Gateway (Distributed Anycast Gateway) for this VNI. The VTEP and SVI for this VLAN must be properly configured for the Distributed Anycast Gateway operation (for example, global anycast gateway MAC address configured and anycast gateway with the virtual IP address on the SVI).
- ARP suppression is a per-L2VNI fabric-wide setting in the VXLAN fabric. Enable or disable this feature consistently across all VTEPs in the fabric. Inconsistent ARP suppression configuration across VTEPs is not supported.
- The VXLAN network identifier (VNID) 16777215 is reserved and should not be configured explicitly.
- VXLAN supports In-Service Software Upgrades (ISSUs). However, VXLAN ISSU is not supported for Cisco Nexus 9300-GX platform switches.
- VXLAN does not support coexistence with the GRE tunnel feature or the MPLS (static or segment-routing) feature.
- VTEP connected to FEX host interface ports is not supported.
- If multiple VTEPs use the same multicast group address for underlay multicast but have different VNIs, the VTEPs should have at least one VNI in common. Doing so ensures that NVE peer discovery occurs and underlay multicast traffic is forwarded correctly. For example, leafs L1 and L4 could have VNI 10 and leafs L2 and L3 could have VNI 20, and both VNIs could share the same group address. When leaf L1 sends traffic to leaf L4, the traffic could pass through leaf L2 or L3. Because NVE peer L1 is not learned on leaf L2 or L3, the traffic is dropped. Therefore, VTEPs that share a group address need to

have at least one VNI in common so that peer learning occurs and traffic is not dropped. This requirement applies to VXLAN bud-node topologies.

- VXLAN does not support coexistence with MVR and MPLS for Cisco Nexus 9504 and 9508 with -R line cards.
- 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.

- For Cisco Nexus 9504 and 9508 switches with -R line cards, the L3VNI's VLAN must be added on the vPC peer-link trunk's allowed VLAN list.
- Native VLANs are supported as transit traffic over a VXLAN fabric on Cisco Nexus 9300-EX/FX/FX2/FX3/GX Series switches.
- To refresh the frozen duplicate host during fabric forwarding, use only "**fabric forwarding dup-host-recovery-timer**" command and do not use "**fabric forwarding dup-host-unfreeze-timer**" command, as it is deprecated.
- For traceroute through a VXLAN fabric when using L3VNI, the following scenario is the expected behavior:

  If L3VNI is associated with a VRF and an SVI, the associated SVI does not have an L3 address that is configured but instead has the "ip forward" configuration command. Due to this interface setup it cannot respond back to the traceroute with its own SVI address. Instead, when a traceroute involving the L3VNI is run through the fabric, the IP address reported will be the lowest IP address of an SVI that belongs to the corresponding tenant VRF.
- Routing protocol adjacencies using Anycast Gateway SVIs is not supported.

Considerations for VXLAN Deployment

- For scale environments, the VLAN IDs related to the VRF and Layer-3 VNI (L3VNI) must be reserved with the **system vlan nve-overlay id** command.

  This is required to optimize the VXLAN resource allocation to scale the following platforms:

  - Cisco Nexus 9300 platform switches
  - Cisco Nexus 9500 platform switches with 9500 line cards

  The following example shows how to reserve the VLAN IDs related to the VRF and the Layer-3 VNI:

```
system vlan nve-overlay id 2000

        vlan 2000
          vn-segment 50000

        interface Vlan2000
          vrf member MYVRF_50000
          ip forward
          ipv6 forward
```

```
vrf context MYVRF_50000
  vni 50000
```

Note The **system vlan nve-overlay id** command should be used for a VRF or a Layer-3 VNI (L3VNI) only. Do not use this command for regular VLANs or Layer-2 VNIs (L2VNI).

- When configuring VXLAN BGP EVPN, the "System Routing Mode: Default" is applicable for the following hardware platforms:
  - Cisco Nexus 9200 platform switches
  - Cisco Nexus 9300 platform switches
  - Cisco Nexus 9300-EX platform switches
  - Cisco Nexus 9300-FX/FX2/FX3 platform switches
  - Cisco Nexus 9300-GX platform switches
  - Cisco Nexus 9500 platform switches with X9500 line cards
  - Cisco Nexus 9500 platform switches with X9700-EX/FX line cards
- The “System Routing Mode: template-vxlan-scale” is not applicable.
- When using VXLAN BGP EVPN in combination with Cisco NX-OS Release 7.0(3)I4(x) or NX-OS Release 7.0(3)I5(1), the “System Routing Mode: template-vxlan-scale” is required on the following hardware platforms:
  - Cisco Nexus 9300-EX Switches
  - Cisco Nexus 9500 Switches with X9700-EX line cards
- Changing the “System Routing Mode” requires a reload of the switch.
- 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, 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.

Note Though the loopback is still down, the traffic is encapsulated and sent to fabric.

- 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 protocol.

- In VXLAN flood and learn mode, the default gateway for VXLAN VLAN is recommended to be a centralized gateway on a pair of vPC devices with FHRP (First Hop Redundancy Protocol) running between them.

  In BGP EVPN, it is recommended to use the anycast gateway feature on all VTEPs.

- For flood and learn mode, only a centralized Layer 3 gateway is supported. Anycast gateway is not supported. The recommended Layer 3 gateway design would be a pair of switches in vPC to be the Layer 3 centralized gateway with FHRP protocol running on the SVIs. The same SVI's cannot span across multiple VTEPs even with different IP addresses used in the same subnet.

Note When configuring SVI with flood and learn mode on the central gateway leaf, it is mandatory to configure **hardware access-list tcam region arp-ether** *size* **double-wide**. (You must decrease the size of an existing TCAM region before using this command.)

For example:

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

Note Configuring the **hardware access-list tcam region arp-ether** *size* **double-wide** is not required on Cisco Nexus 9200 Series switches.

- 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.)

Note This step is required for Cisco Nexus 9300 switches (NFE/ALE) and Cisco Nexus 9500 switches with N9K-X9564PX, N9K-X9564TX, and N9K-X9536PQ line cards. This step is not needed with Cisco Nexus 9200 switches, Cisco Nexus 9300-EX switches, or Cisco Nexus 9500 switches with N9K-X9732C-EX line cards.

- VXLAN tunnels cannot have more than one underlay next hop on a given underlay port. For example, on a given output underlay port, only one destination MAC address can be derived as the outer MAC on a given output port.

  This is a per-port limitation, not a per-tunnel limitation. This means that two tunnels that are reachable through the same underlay port cannot drive two different outer MAC addresses.

- When changing the IP address of a VTEP device, you must shut the NVE interface before changing the IP address.

- As a best practice, when migrating any sets of VTEP to a multisite BGW, NVE interface must be shut on all the VTEPs where this migration is being performed. NVE interface should be brought back up once the migration is complete and all necessary configurations for multisite are applied to the VTEPs.

- 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.

The following is an example of an anycast RP configuration on spines:

```
ip pim rp-address 1.1.1.10 group-list 224.0.0.0/4
ip pim anycast-rp 1.1.1.10 1.1.1.1
ip pim anycast-rp 1.1.1.10 1.1.1.2
```

Note

- 1.1.1.10 is the anycast RP IP address that is configured on all RPs participating in the anycast RP set.
- 1.1.1.1 is the local RP IP.
- 1.1.1.2 is the peer RP IP.

- Static ingress replication and BGP EVPN ingress replication do not require any IP Multicast routing in the underlay.

vPC Considerations for VXLAN Deployment

- As a best practice, when **feature vpc** is enabled or disabled on a VTEP, the NVE interfaces on both the vPC primary and the vPC secondary must be shut down before the change is made. Enabling **feature vpc** without the vPC domain being properly configured will result in the NVE loopback being held administratively down until the configuration is completed and the vPC peer-link is brought up.
- 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.
- On vPC VXLAN, it is recommended to increase the **delay restore interface-vlan** timer under the vPC configuration, if the number of SVIs are scaled up. For example, if there are 1000 VNIs with 1000 SVIs, we recommend to increase the **delay restore interface-vlan** timer to 45 seconds.
- If a ping is initiated to the attached hosts on VXLAN VLAN from a vPC VTEP node, the source IP address used by default is the anycast IP that is configured on the SVI. This ping can fail to get a response from the host in case the response is hashed to the vPC peer node. This issue can happen when a ping is initiated from a VXLAN vPC node to the attached hosts without using a unique source IP address. As a workaround for this situation, use VXLAN OAM or create a unique loopback on each vPC VTEP and route the unique address via a backdoor path.
- 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.
- 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, 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 or VLAN 4046).

- When the 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 the vPC domain is shut, the loopback interface used by NVE on the VTEP with shutdown vPC domain 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 other vPC VTEP.
- 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 two 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 to facilitate NVE RMAC/VMAC programming on both peers. 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 peer-link in his case. In BUD node topologies, the backup SVI needs to be added as a static OIF for each underlay multicast group.

```
switch# sh ru int vlan 2

interface Vlan2
 description backupl_svi_over_peer-link
 no shutdown
 ip address 30.2.1.1/30
 ip router ospf 1 area 0.0.0.0
 ip pim sparse-mode
 ip igmp static-oif route-map match-mcast-groups

 route-map match-mcast-groups permit 1
  match ip multicast group 225.1.1.1/32
```

Note In BUD node topologies, the backup SVI needs to be added as a static OIF for each underlay multicast group.

The SVI must be configured on bot vPC peers and requires PIM to be enabled.

- 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. 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.
- 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 must be shut before the IP changes are made.
- When SVI is enabled on a VTEP (flood and learn, or EVPN) regardless of ARP suppression, make sure that ARP-ETHER TCAM is carved using the **hardware access-list tcam region arp-ether 256 double-wide** command. This requirement does not apply to Cisco Nexus 9200, 9300-EX, and 9300-FX/FX2/FX3 and 9300-GX platform switches and Cisco Nexus 9500 platform switches with 9700-EX line cards.
- The **show** commands with the **internal** keyword are not supported.
- DHCP snooping (Dynamic Host Configuration Protocol snooping) is not supported on VXLAN VLANs.
- 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.

  See the Cisco Nexus 9000 Series NX-OS Security Configuration Guide, Release 9.3(x) for other guidelines and limitations for the VXLAN ACL feature.

- QoS classification is not supported for VXLAN traffic in the network to access direction on the Layer 3 uplink interface.

  See the Cisco Nexus 9000 Series NX-OS Quality of Service Configuration Guide, Release 9.3(x) for other guidelines and limitations for the VXLAN QoS feature.

- The QoS buffer-boost feature is not applicable for VXLAN traffic.
- Beginning with Cisco NX-OS Release 9.3(5), VTEPs support VXLAN-encapsulated traffic over parent interfaces if subinterfaces are configured.
- VTEPs do not support VXLAN encapsulated traffic over subinterfaces. This is regardless of VRF participation or IEEE802.1Q encapsulation.
- Mixing subinterfaces for VXLAN and non-VXLAN VLANs is not supported.
- Point-to-multipoint Layer 3 and SVI uplinks are not supported.
- Using the **ip forward** command enables the VTEP to forward the VXLAN de-capsulated packet destined to its router IP to the SUP/CPU.
- Before configuring it as an SVI, the backup VLAN needs to be configured on Cisco Nexus 9200, 9300-EX, and 9300-FX/FX2/FX3 and 9300-GX platform switches as an infra-VLAN with the **system nve infra-vlans** command.
- VXLAN is supported on Cisco Nexus 9500 platform switches with the following line cards:
  - 9700-EX
  - 9700-FX
- When Cisco Nexus 9500 platform switches are used as VTEPs, 100G line cards are not supported on Cisco Nexus 9500 platform switches. This limitation does not apply to a Cisco Nexus 9500 switch with 9700-EX or -FX line cards.
- Cisco Nexus 9300 platform switches with 100G uplinks only support VXLAN switching/bridging. Cisco Nexus 9200 and Cisco Nexus 9300-EX/FX/FX2 platform switches do not have this restriction.

Note For VXLAN routing support, a 40 G uplink module is required.

- 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.
- For Cisco Nexus 9200 platform switches that have the Application Spine Engine (ASE2). There exists a Layer 3 VXLAN (SVI) throughput issue. There is a data loss for packets of sizes 99 - 122.
- The VXLAN network identifier (VNID) 16777215 is reserved and should not be configured explicitly.
- VXLAN supports In Service Software Upgrade (ISSU).
- VXLAN ISSU is not supported on the Cisco Nexus 9300-GX platform switches.
- VXLAN does not support coexistence with the GRE tunnel feature or the MPLS (static or segment routing) feature.
- 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.

- When ARP suppression is enabled or disabled in a vPC setup, a down time is required because the global VXLAN vPC consistency checker will fail and the VLANs will be suspended if ARP suppression is disabled or enabled on only one side.

Note For information about VXLAN BGP EVPN scalability, see the *Cisco Nexus 9000 Series NX-OS Verified Scalability Guide, Release 9.3(x)*.

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 9000 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 9000 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:
  - Create and configure a loopback interface with a /32 IP address.

    (For vPC VTEPs, you must configure primary and secondary /32 IP addresses.)
  - Advertise the loopback interface /32 addresses through the routing protocol (static route) that runs in the transport network.
- Throughout the transport network:

For Cisco Nexus 9200, 9300-EX, and 9300-FX/FX2/FX3 and 9300-GX platform switches, the use of the **system nve infra-vlans** command is required. Otherwise, VXLAN traffic (IP/UDP 4789) is actively treated by the switch. The following scenarios are a non-exhaustive list but most commonly seen, where the need for a **system nve infra-vlans** definition is required.

Every VLAN that is not associated with a VNI (vn-segment) is required to be configured as a **system nve infra-vlans** in the following cases:

In the case of VXLAN flood and learn as well as VXLAN EVPN, the presence of non-VXLAN VLANs could be related to:

- An SVI related to a non-VXLAN VLAN is used for backup underlay routing between vPC peers via a vPC peer-link (backup routing).
- An SVI related to a non-VXLAN VLAN is required for connecting downstream routers (external connectivity, dynamic routing over vPC).
- An SVI related to a non-VXLAN VLAN is required for per Tenant-VRF peering (L3 route sync and traffic between vPC VTEPs in a Tenant VRF).
- An SVI related to a non-VXLAN VLAN is used for first-hop routing toward endpoints (Bud-Node).

In the case of VXLAN flood and learn, the presence of non-VXLAN VLANs could be related to:

- An SVI related to a non-VXLAN VLAN is used for an underlay uplink toward the spine (Core port).

The rule of defining VLANs as **system nve infra-vlans** can be relaxed for special cases such as:

- An SVI related to a non-VXLAN VLAN that does not transport VXLAN traffic (IP/UDP 4789).
- Non-VXLAN VLANs that are not associated with an SVI or not transporting VXLAN traffic (IP/UDP 4789).

Note You must not configure certain combinations of infra-VLANs. For example, 2 and 514, 10 and 522, which are 512 apart. This is specifically but not exclusive to the "Core port" scenario that is described for VXLAN flood and learn.

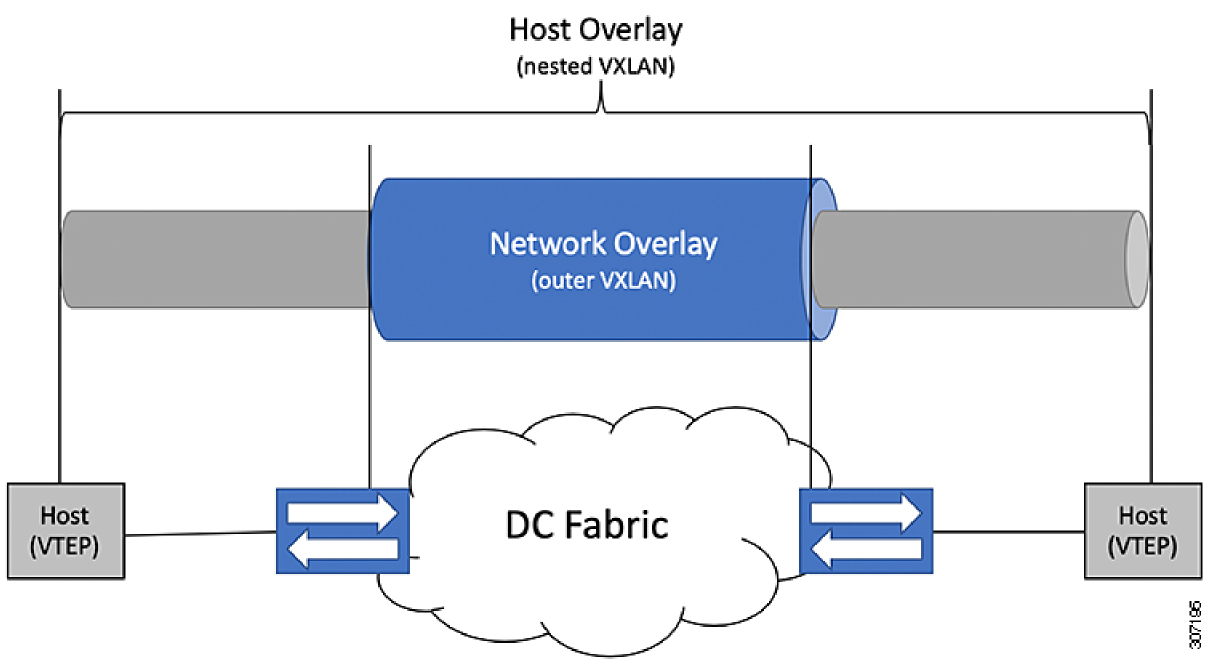
Considerations for Tunneling VXLAN

DC Fabrics with VXLAN BGP EVPN are becoming the transport infrastructure for overlays. These overlays, often originated on the server (Host Overlay), require integration or transport over the top of the existing transport infrastructure (Network Overlay).

Nested VXLAN (Host Overlay over Network Overlay) support has been added starting with Cisco NX-OS Release 7.0(3)I7(4) and Cisco NX-OS Release 9.2(2) on the Cisco Nexus 9200, 9300-EX, 9300-FX, 9300-FX2, 9500-EX, 9500-FX platform switches. It is also supported for Cisco Nexus 9300-FX3 platform switches starting with Cisco NX-OS Release 9.3(5).

Nested VXLAN is not supported on a Layer 3 interface or a Layer 3 port-channel interface in Cisco NX-OS Release 9.3(4) and prior releases. It is supported on a Layer 3 interface or a Layer 3 port-channel interface from Cisco NX-OS Release 9.3(5) onwards.

Figure 2: Host Overlay



To provide Nested VXLAN support, the switch hardware and software must differentiate between two different VXLAN profiles:

- VXLAN originated behind the Hardware VTEP for transport over VXLAN BGP EVPN (nested VXLAN)
- VXLAN originated behind the Hardware VTEP to integrated with VXLAN BGP EVPN (BUD Node)

The detection of the two different VXLAN profiles is automatic and no specific configuration is needed for nested VXLAN. As soon as VXLAN encapsulated traffic arrives in a VXLAN enabled VLAN, the traffic is transported over the VXLAN BGP EVPN enabled DC Fabric.

The following attachment modes are supported for Nested VXLAN:

- Untagged traffic (in native VLAN on a trunk port or on an access port)
- Tagged traffic Layer 2 ports (tagged VLAN on a IEEE 802.1Q trunk port)
- Untagged and tagged traffic that is attached to a vPC domain
- Untagged traffic on a Layer 3 interface or a Layer 3 port-channel interface
- Tagged traffic on Layer 3 interface or a Layer 3 port-channel interface

Configuring VXLAN

Enabling VXLANs

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **feature nv overlay**
3. [**no**] **feature vn-segment-vlan-based**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

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

Mapping VLAN to VXLAN VNI

SUMMARY STEPS

1. **configure terminal**
2. **vlan** *vlan-id*
3. **vn-segment** *vnid*
4. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**vlan** *vlan-id*	Specifies VLAN.
Step 3	**vn-segment** *vnid*	Specifies VXLAN VNID (Virtual Network Identifier)
Step 4	**exit**	Exit configuration mode.

Creating and Configuring an NVE Interface and Associate VNIs

An NVE interface is the overlay interface that terminates VXLAN tunnels.

You can create and configure an NVE (overlay) interface with the following:

SUMMARY STEPS

1. **configure terminal**
2. **interface nve** *x*
3. **source-interface** *src-if*
4. **member vni** *vni*
5. **mcast-group** *start-address* [*end-address*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**interface nve** *x*	Creates a VXLAN overlay interface that terminates VXLAN tunnels. **Note** Only 1 NVE interface is allowed on the switch.
Step 3	**source-interface** *src-if*	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 transient devices in the transport network and the remote VTEPs. This is accomplished by advertising it through a dynamic routing protocol in the transport network.
Step 4	**member vni** *vni*	Associate VXLAN VNIs (Virtual Network Identifiers) with the NVE interface.
Step 5	**mcast-group** *start-address* [*end-address*]	Assign a multicast group to the VNIs. **Note** Used only for BUM traffic

Configuring a VXLAN VTEP in vPC

You can configure a VXLAN VTEP in a vPC.

SUMMARY STEPS

1. Enter global configuration mode.
2. Enable the vPC feature on the device.
3. Enable the interface VLAN feature on the device.
4. Enable the LACP feature on the device.
5. Enable the PIM feature on the device.
6. Enables the OSPF feature on the device.
7. Define a PIM RP address for the underlay multicast group range.
8. Define a non-VXLAN enabled VLAN as a backup routed path.
9. Create the VLAN to be used as an infra-VLAN.
10. Create the SVI used for the backup routed path over the vPC peer-link.
11. Create primary and secondary IP addresses.
12. Create a primary IP address for the data plane loopback interface.
13. Create a vPC domain.
14. Configure the IPv4 address for the remote end of the vPC peer-keepalive link.
15. Enable Peer-Gateway on the vPC domain.
16. Enable Peer-switch on the vPC domain.
17. Enable IP ARP synchronize under the vPC domain to facilitate faster ARP table population following device reload.
18. (Optional) Enable IPv6 nd synchronization under the vPC domain to facilitate faster nd table population following device reload.
19. Create the vPC peer-link port-channel interface and add two member interfaces.
20. Modify the STP hello-time, forward-time, and max-age time.
21. (Optional) Enable the delay restore timer for SVI's.

DETAILED STEPS

Step 1 Enter global configuration mode.

```
switch# configure terminal
```

Step 2 Enable the vPC feature on the device.

```
switch(config)# feature vpc
```

Step 3 Enable the interface VLAN feature on the device.

```
switch(config)# feature interface-vlan
```

Step 4 Enable the LACP feature on the device.

```
switch(config)# feature lacp
```

Step 5 Enable the PIM feature on the device.

```
switch(config)# feature pim
```

Step 6 Enables the OSPF feature on the device.

```
switch(config)# feature ospf
```

Step 7 Define a PIM RP address for the underlay multicast group range.

```
switch(config)# ip pim rp-address 192.168.100.1 group-list 224.0.0/4
```

Step 8 Define a non-VXLAN enabled VLAN as a backup routed path.

```
switch(config)# system nve infra-vlans 10
```

Step 9 Create the VLAN to be used as an infra-VLAN.

```
switch(config)# vlan 10
```

Step 10 Create the SVI used for the backup routed path over the vPC peer-link.

```
switch(config)# interface vlan 10
switch(config-if)# ip address 10.10.10.1/30
switch(config-if)# ip router ospf UNDERLAY area 0
switch(config-if)# ip pim sparse-mode
switch(config-if)# no ip redirects
switch(config-if)# mtu 9216
(Optional)switch(config-if)# ip igmp static-oif route-map match-mcast-groups
switch(config-if)# no shutdown
(Optional)switch(config)# route-map match-mcast-gropus permit 10
(Optional)switch(config-route-map)# match ip multicast group 225.1.1.1/32
```

Step 11 Create primary and secondary IP addresses.

```
switch(config)# interface loopback 0
switch(config-if)# description Control_plane_Loopback
switch(config-if)# ip address x.x.x.x/32
switch(config-if)# ip router ospf process tag area area id
switch(config-if)# ip pim sparse-mode
switch(config-if)# no shutdown
```

Step 12 Create a primary IP address for the data plane loopback interface.

```
switch(config)# interface loopback 1
switch(config-if)# description Data_Plane_loopback
switch(config-if)# ip address z.z.z.z/32
switch(config-if)# ip address y.y.y.y/32 secondary
switch(config-if)# ip router ospf process tag area area id
switch(config-if)# ip pim sparse-mode
switch(config-if)# no shutdown
```

Step 13 Create a vPC domain.

```
switch(config)# vpc domain 5
```

Step 14 Configure the IPv4 address for the remote end of the vPC peer-keepalive link.

```
switch(config-vpc-domain)# peer-keepalive destination 172.28.230.85
```

Note The system does not form the vPC peer link until you configure a vPC peer-keepalive link

The management ports and VRF are the defaults.

Note We recommend that you configure a separate VRF and use a Layer 3 port from each vPC peer device in that VRF for the vPC peer-keepalive link. For more information about creating and configuring VRFs, see the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide.

Step 15 Enable Peer-Gateway on the vPC domain.

```
switch(config-vpc-domain)# peer-gateway
```

Note Disable IP redirects on all interface-vlans of this vPC domain for correct operation of this feature.

Step 16 Enable Peer-switch on the vPC domain.

```
switch(config-vpc-domain)# peer-switch
```

Note Disable IP redirects on all interface-vlans of this vPC domain for correct operation of this feature.

Step 17 Enable IP ARP synchronize under the vPC domain to facilitate faster ARP table population following device reload.

```
switch(config-vpc-domain)# ip arp synchronize
```

Step 18 (Optional) Enable IPv6 nd synchronization under the vPC domain to facilitate faster nd table population following device reload.

```
switch(config-vpc-domain)# ipv6 nd synchronize
```

Step 19 Create the vPC peer-link port-channel interface and add two member interfaces.

```
switch(config)# interface port-channel 1
switch(config-if)# switchport
switch(config-if)# switchport mode trunk
switch(config-if)# switchport trunk allowed vlan 1,10,100-200
switch(config-if)# mtu 9216
switch(config-if)# vpc peer-link
switch(config-if)# no shutdown
switch(config-if)# interface Ethernet 1/1 , 1/21
switch(config-if)# switchport
switch(config-if)# mtu 9216
switch(config-if)# channel-group 1 mode active
switch(config-if)# no shutdown
```

Step 20 Modify the STP hello-time, forward-time, and max-age time.

As a best practice, we recommend changing the **hello-time** to four seconds to avoid unnecessary TCN generation when the vPC role change occurs. As a result of changing the **hello-time**, it is also recommended to change the **max-age** and **forward-time** accordingly.

```
switch(config)# spanning-tree vlan 1-3967 hello-time 4
switch(config)# spanning-tree vlan 1-3967 forward-time 30
switch(config)# spanning-tree vlan 1-3967 max-age 40
```

Step 21 (Optional) Enable the delay restore timer for SVI's.

We recommend that you tune this value when the SVI or VNI scale is high. For example, when the SVI count is 1000, we recommended setting the delay restore for interface-vlan to 45 seconds.

```
switch(config-vpc-domain)# delay restore interface-vlan 45
```

Configuring Static MAC for VXLAN VTEP

Static MAC for VXLAN VTEP is supported on Cisco Nexus 9300 Series switches with flood and learn. This feature enables the configuration of static MAC addresses behind a peer VTEP.

Note Static MAC cannot be configured for a control plane with a BGP EVPN-enabled VNI.

SUMMARY STEPS

1. **configure terminal**
2. **mac address-table static** *mac-address* **vni** *vni-id* **interface nve** *x* **peer-ip** *ip-address*
3. **exit**
4. (Optional) **copy running-config startup-config**
5. (Optional) **show mac address-table static interface nve** *x*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**mac address-table static** *mac-address* **vni** *vni-id* **interface nve** *x* **peer-ip** *ip-address*	Specifies the MAC address pointing to the remote VTEP.
Step 3	**exit**	Exits global configuration mode.
Step 4	(Optional) **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.
Step 5	(Optional) **show mac address-table static interface nve** *x*	Displays the static MAC addresses pointing to the remote VTEP.

Example

The following example shows the output for a static MAC address configured for VXLAN VTEP:

```
switch# show mac address-table static interface nve 1

Legend:
        * - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
        age - seconds since last seen,+ - primary entry using vPC Peer-Link,
        (T) - True, (F) - False
   VLAN     MAC Address      Type      age     Secure NTFY Ports
---------+-----------------+--------+---------+------+----+------------------
*  501     0047.1200.0000   static   -         F      F    nve1(33.1.1.3)
*  601     0049.1200.0000   static   -         F      F    nve1(33.1.1.4)
```

Disabling VXLANs

SUMMARY STEPS

1. **configure terminal**
2. **no feature vn-segment-vlan-based**

3. **no feature nv overlay**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

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

Configuring BGP EVPN Ingress Replication

The following enables BGP EVPN with ingress replication for peers.

SUMMARY STEPS

1. **configure terminal**
2. **interface nve** *x*
3. **source-interface** *src-if*
4. **member vni** *vni*
5. **ingress-replication protocol bgp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**interface nve** *x*	Creates a VXLAN overlay interface that terminates VXLAN tunnels. **Note** Only 1 NVE interface is allowed on the switch.
Step 3	**source-interface** *src-if*	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 transient devices in the transport network and the remote VTEPs. This is accomplished by advertising it through a dynamic routing protocol in the transport network.
Step 4	**member vni** *vni*	Associate VXLAN VNIs (Virtual Network Identifiers) with the NVE interface.
Step 5	**ingress-replication protocol bgp**	Enables BGP EVPN with ingress replication for the VNI.

Configuring Static Ingress Replication

The following enables static ingress replication for peers.

SUMMARY STEPS

1. **configuration terminal**
2. **interface nve** *x*
3. **member vni** [*vni-id* | *vni-range*]
4. **ingress-replication protocol static**
5. **peer-ip** *n.n.n.n*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configuration terminal**	Enters global configuration mode.
Step 2	**interface nve** *x*	Creates a VXLAN overlay interface that terminates VXLAN tunnels. **Note** Only 1 NVE interface is allowed on the switch.
Step 3	**member vni** [*vni-id* | *vni-range*]	Maps VXLAN VNIs to the NVE interface.
Step 4	**ingress-replication protocol static**	Enables static ingress replication for the VNI.
Step 5	**peer-ip** *n.n.n.n*	Enables peer IP.

VXLAN and IP-in-IP Tunneling

Cisco NX-OS Release 9.3(6) and later releases support the coexistence of VXLAN and IP-in-IP tunneling.

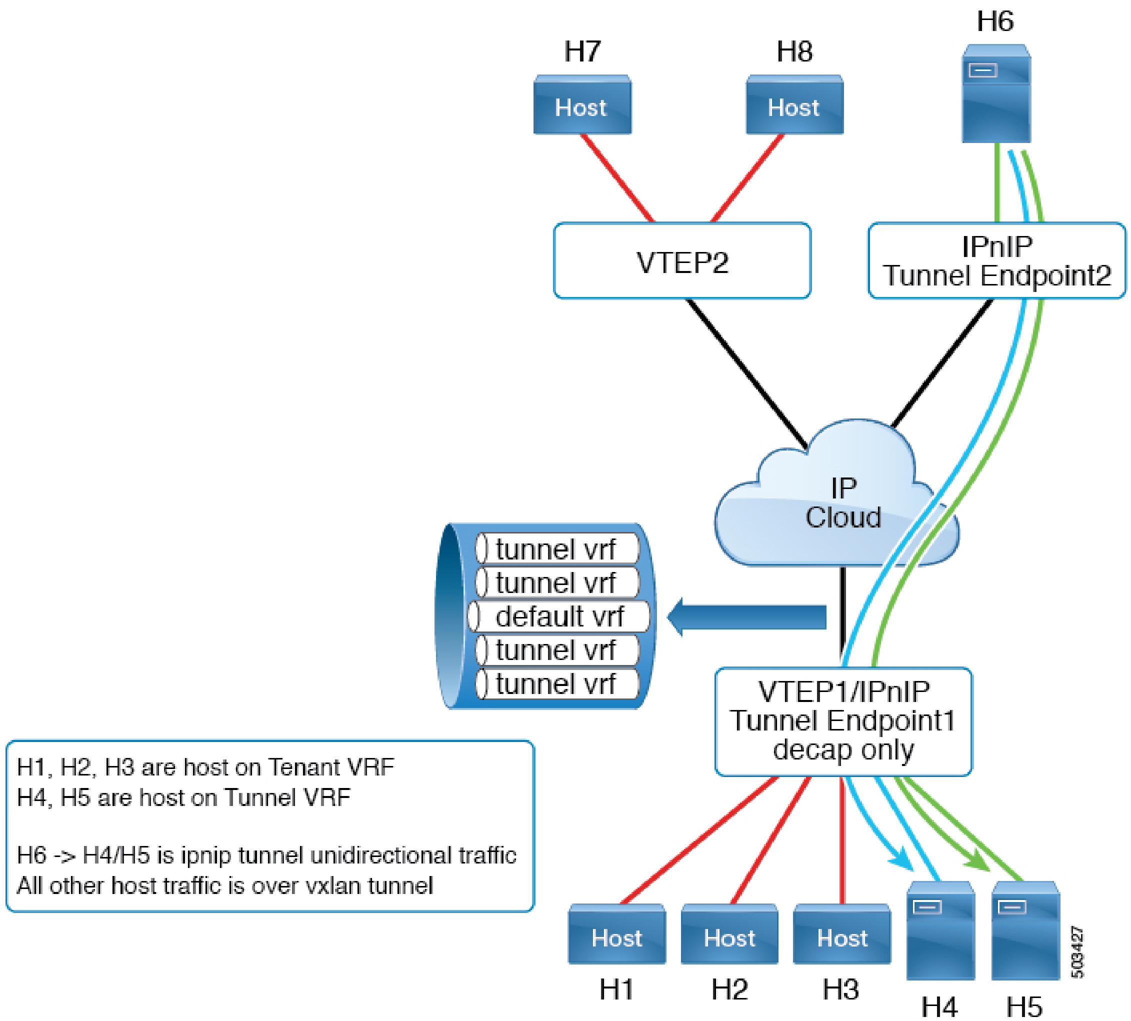
Coexistence of these features requires isolating IP-in-IP tunnels and VXLAN within their own VRFs. By isolating the VRFs, both VXLAN and the tunnels operate independently. VXLAN tunnel termination isn't reencapsulated as an IP-in-IP tunnel (or conversely) on the same or different VRFs.

By configuring subinterfaces under the interface to isolate VRFs, the same uplinks can be used to carry both VXLAN and IP-in-IP tunnel traffic. The parent port can be on the default VRF and subinterfaces on the non-default VRFs.

To terminate IP-in-IP encapsulated packets received on port-channel sub-interfaces, these sub-interfaces must be configured under the same non-default VRF as the tunnel interface, and can only be member of ***one*** non-default VRF.

Multiple port-channel sub interfaces from a different parent PC can still be configured under the same non-default VRF to terminate IP-in-IP encapsulation. The limitation only applies for sub-interfaces under one port-channel. This limitation is not applicable for L3 ports.

As the following example shows, VXLAN traffic is forwarded on the parent interface (eth1/1) in the default VRF, and IP-in-IP (non-VXLAN) traffic is forwarded on subinterfaces (eth1/1.10) in the tunnel VRF.



Cisco Nexus 9300-FX2 platform switches support the coexistence of VXLAN and IP-in-IP tunneling with the following limitations:

- VXLAN must be configured in the default VRF.
- Coexistence is supported on VXLAN with the EVPN control plane.
- IP-in-IP tunneling must be configured in the non-default VRF and is supported only in decapsulate-any mode.

Note If you try to enable VXLAN when a decapsulate-any tunnel is configured in the default VRF, an error message appears. It states that VXLAN and IP-in-IP tunneling can coexist only for a decapsulate-any tunnel in the non-default VRF and to remove the configuration.

- Point-to-point GRE tunnels are not supported. If you try to configure point-to-point tunnels, an error message appears indicating that VXLAN and IP-in-IP tunneling can coexist only for a decapsulate-any tunnel.

- Typically to configure a tunnel, you need to provide the two endpoints. However, decapsulate-any is a receive-only tunnel, so you need to provide only the source IP address or source interface name. The tunnel terminates on any IP interface in the same VRF.
- Tunnel statistics don't support egress counters.
- VXLAN and IP-in-IP tunnels can't share the same source loopback interface. Each tunnel must have its own source loopback interface.

The following example shows a sample configuration:

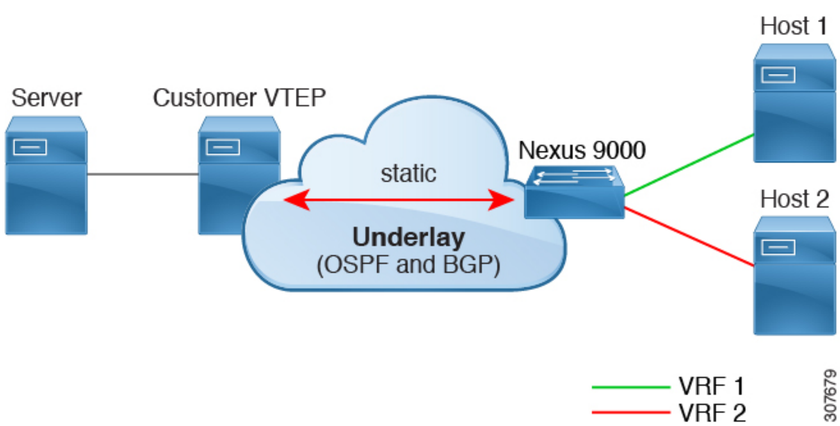
```
feature vn-segment-vlan-based
feature nv overlay
feature tunnel
nv overlay evpn

interface ethernet 1/1
    description VXLAN carrying interface
    no switchport
    ip address 10.1.1.1/30

interface ethernet 1/1.10
    description IPinIP carrying interface
    no switchport
    vrf member tunnel
    encapsulation dot1q 100
    ip address 10.10.1.1/30

interface loopback 0
    description VXLAN-loopback
    ip address 125.125.125.125/32

interface loopback 100
    description Tunnel_loopback
    vrf member tunnel
    ip address 5.5.5.5/32

interface Tunnel1
    vrf member tunnel
    ip address 55.55.55.1/24
    tunnel mode ipip decapsulate-any ip
    tunnel source loopback100
    tunnel use-vrf tunnel
    no shutdown

interface nve1
    host-reachability protocol bgp
    source-interface loopback0
    global mcast-group 224.1.1.1 L2
    global mcast-group 225.3.3.3 L3
    member vni 10000
    suppress-arp
    ingress-replication protocol bgp
    member vni 55500 associate-vrf
```

Configuring VXLAN Static Tunnels

About VXLAN Static Tunnels

Beginning with Cisco NX-OS Release 9.3(3), some Cisco Nexus switches can connect to a customer-provided software VTEP over static tunnels. Static tunnels are customer defined and support VXLAN-encapsulated traffic between hosts without requiring a control plane protocol such as BGP EVPN. You can configure static tunnels manually from the Nexus switch or programmatically, such as through a NETCONF client in the underlay.

Figure 3: VXLAN Static Tunnel Connecting Software VTEP



Static tunnels are supported per VRF. Each VRF can have a dedicated L3VNI to transport a packet with proper encapsulation and decapsulation on the switch and the software VTEP, the static peer. Typically, the static peer is a Cisco Nexus 1000V or bare-metal server with one or more VMs terminating one or more VNIs. However, a static peer can be any customer-developed device that complies with RFC 7348, *Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks*. Because the customer provides the static peer and a control plane protocol is not present, you must ensure that the static peer forwards the VXLAN-related configuration and routes to the correct hosts.

Beginning with Cisco NX-OS Release 9.3(5), this feature supports the handling of packets coming in and going out of the tunnel. Specifically, it allows the Nexus switch to send packets to the hosts or other switches over the tunnel. In Cisco NX-OS Releases 9.3(3) and 9.3(4), VXLAN static tunnels support communication only from the local host to the remote host.

Guidelines and Limitations for VXLAN Static Tunnels

The VXLAN static tunnels feature has the following guidelines and limitations:

- The Cisco Nexus 9332C, 9364C, 9300-EX, and 9300-FX/FX2/FX3, 9300-GX and 9300-FX3platform switches support VXLAN static tunnels.
- Beginning with Cisco NX-OS Release 10.1(1), VXLAN Static Tunnels are supported on Cisco Nexus 9300-FX3 platform switches.
- The following guidelines apply to software VTEPs:

    - The software VTEP must be configured as needed to determine how to forward traffic from the VNI.
    - The software VTEP must be compliant with RFC 7348.

- The underlay can be OSPFv2, BGP, IS-IS, or IPv4.
- The overlay can be IPv4 only.
- Additional VXLAN features (such as TRM, Multi-Site, OAM, Cross Connect, and VXLAN QoS), IGMP snooping, MPLS handoff, static MPLS, SR, and SRv6 are not supported.
- Pings across the overlay from local tenant VRF loopback to a host behind the software VTEP is not supported.
- Static tunnels do not support ECMP configuration.
- Static tunnels cannot be configured in the same fabric as traditional flood and learn or BGP EVPN fabrics.
- Local hosts are not supported for VNI-enabled VLANs. Therefore, you cannot have a host in the same VLAN where you configured the VNI.
- Fabric forwarding is supported with static tunnels. When fabric forwarding is enabled, be aware that it affects how SVIs and MAC addresses are used. Consider the following example configuration.

```
feature fabric forwarding
fabric forwarding anycast-gateway-mac 0000.0a0a.0a0a

interface Vlan802
no shutdown
vrf member vrfvxlan5201
ip address 103.33.1.1/16
fabric forwarding mode anycast-gateway
```

  When fabric forwarding is enabled:

    - all SVIs where **fabric forwarding mode anycast-gateway** is configured (for example, Vlan802) are used.
    - the MAC address configured with **fabric forwarding anycast-gateway-mac anycast-mac-address** (0000.0a0a.0a0a) is used.

Enabling VXLAN Static Tunnels

Enable the following features to enable VXLAN Static Tunnels.

SUMMARY STEPS

1. **config terminal**
2. **feature vn-segment**
3. **feature ofm**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**feature vn-segment**
	Example:
	`switch(config)# feature vn-segment`
	`switch(config)#`
		Enable VLAN-based VXLAN.

Step 3	**feature ofm**
	Example:
	`switch(config)# feature ofm`
	`switch(config)#`
		Enable static VXLAN tunnels.

What to do next

Configure the VRF overlay VLAN for VXLAN routing over Static Tunnels.

Configuring VRF Overlay for Static Tunnels

A VRF overlay must be configured for the VXLAN Static Tunnels.

SUMMARY STEPS

1. **vlan** *number*
2. **vn-segment** *number*

DETAILED STEPS

	Command or Action	Purpose

Step 1	**vlan** *number*
	Example:
	`switch(config)# vlan 2001`
	`switch(config-vlan)#`
		Specify the VLAN.

Step 2	**vn-segment** *number*
	Example:
	`switch(config-vlan)# vn-segment 20001`
	`switch(config-vlan)#`
		Specify the VN segment.

What to do next

Configure the VRF for VXLAN Routing over the Static Tunnel.

Configuring a VRF for VXLAN Routing

Configure the tenant VRF.

SUMMARY STEPS

1. **vrf context** *vrf-name*
2. **vni** *number*

DETAILED STEPS

	Command or Action	Purpose

Step 1	**vrf context** *vrf-name*
	Example:
	`switch(config-vlan)# vrf context cust1`
	`switch(config-vrf)#`
		Configure the tenant VRF.

Step 2	**vni** *number*
	Example:
	`switch(config-vrf)# vni 20001`
	`switch(config-vrf)#`
		Specify the VNI for the tenant VRF.

What to do next

Configure the L3 VNI for the host.

Configuring the L3 VNI for Static Tunnels

Configure the L3 VNI for the VTEPs.

Before you begin

The VLAN interface feature must be enabled. Use **feature interface-vlan** if needed.

SUMMARY STEPS

1. **vlan** *number*
2. **interface** *vlan-number*
3. **vrf member** *vrf-name*
4. **ip forward**
5. **no shutdown**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**vlan** *number* **Example:**	Specify the VLAN number

	Command or Action	Purpose
	`switch(config-vrf)# vlan 2001` `switch(config-vlan)#`	

Step 2	**interface** *vlan-number*
	Example:
	`switch(config)# interface vlan2001`
	`switch(config-if)#`
		Specify the VLAN interface.

Step 3	**vrf member** *vrf-name*
	Example:
	`switch(config-if)# vrf member cust1`
	`Warning: Deleted all L3 config on interface Vlan2001`
	`switch(config-if)#`
		Assign the VLAN interface to the tenant VRF.

Step 4	**ip forward**
	Example:
	`switch(config-if)# ip forward`
	`switch(config-if)#`
		Enable IPv4 traffic on the interface.

Step 5	**no shutdown**
	Example:
	`switch(config-if)# no shutdown`
	`switch(config-if)#`
		Enables the interface.

What to do next

Configure the tunnel profile.

Configuring the Tunnel Profile

To configure static tunnels, you create a tunnel profile that specifies the interface on the Nexus switch, the MAC address of the static peer, and the interface on the static peer.

Before you begin

To configure VXLAN static tunnels, the underlay must be completely configured and operating correctly.

SUMMARY STEPS

1. **tunnel-profile** *profile-name*
2. **encapsulation {***VXLAN* | *VXLAN-GPE* | *SRv6***}**
3. **source-interface loopback** *virtual-interface-number*
4. **route vrf** *tenant-vrf destination-host-prefix destination-vtep-ip-address* **next-hop-vrf** *destination-vtep-vrf* **vni** *vni-number* **dest-vtep-mac** *destination-vtep-mac-address*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**tunnel-profile** *profile-name* **Example:** `switch(config)# ` **`tunnel-profile test`** `switch(config-tnl-profile)#`	Create and name the tunnel profile.
Step 2	**encapsulation {***VXLAN* | *VXLAN-GPE* | *SRv6***}** **Example:** `switch(config-tnl-profile)# ` **`encapsulation vxlan`** `switch(config-tnl-profile)#`	Set the appropriate encapsulation type for the tunnel profile. **Note** In NX-OS release 9.3(3), only encapsulation type **vxlan** is supported.
Step 3	**source-interface loopback** *virtual-interface-number* **Example:** `switch(config-tnl-profile)# ` **`source-interface loopback 1`** `switch(config-tnl-profile)#`	Configure the loopback interface as the source interface for the tunnel profile, where the virtual interface number is from 0 to 1023.
Step 4	**route vrf** *tenant-vrf destination-host-prefix destination-vtep-ip-address* **next-hop-vrf** *destination-vtep-vrf* **vni** *vni-number* **dest-vtep-mac** *destination-vtep-mac-address* **Example:** `switch(tunnel-profile)# ` **`route vrf cust1 101.1.1.2/32 7.7.7.1 next-hop-vrf default vni 20001 dest-vtep-mac f80f.6f43.036c`** `switch(tunnel-profile)#`	Create the tunnel route by specifying the destination software VTEP and entering the route information for the VNI and destination VTEP MAC address. **Note** The **route vrf** command accepts one *destination-vtep-mac-address* per *destination-vtep-ip-address* across all the routes. If you configure additional routes, they are cached as errored routes and a error syslog is generated for each.

Verifying VXLAN Static Tunnels

VXLAN static tunnels remain configured if one end of the tunnel goes down. While one end of the tunnel is down, packets are dropped because that VTEP is unreachable. When the down VTEP comes back online, traffic can resume across the tunnel after the underlay relearns connectivity.

You can use **show** commands to check the state of the tunnel profile and tunnel route.

Before you begin

SUMMARY STEPS

1. **show tunnel-profile**
2. **show ip route** *tenant-vrf-name*
3. **show running-config ofm**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show tunnel-profile**	Shows information about the tunnel profile for the software.
Step 2	**show ip route** *tenant-vrf-name*	Shows route information for the VRF connecting to the software VTEP. For example, you can use this command when a `route unreachable` error occurs to verify that a route exists for a VRF's tunnel.
Step 3	**show running-config ofm**	Shows the running config for the OFM feature and static tunnels. You can use this command when a `route unreachable` error occurs to check whether the route information for the destination VTEP is present.

What to do next

In addition to VXLAN verification, you can use SPAN to check the ports and source VLANs for packets traversing the switch.

Example Configurations for VXLAN Static Tunnels

The following configuration examples shows VXLAN static tunnel configurations through the supported methods.

NX-OS CLI

```
vlan 2001
vlan 2001
  vn-segment 20001

interface Vlan2001
  no shutdown
  vrf member cust1
  ip forward

vrf context cust1
  vni 20001


feature ofm

tunnel-profile test
  encapsulation vxlan
  source-interface loopback1
  route vrf cust1 101.1.1.2/32 7.7.7.1 next-hop-vrf default vni 20001 dest-vtep-mac
f80f.6f43.036c
```

CHAPTER 4

Configuring the Underlay

This chapter contains the following sections:

IP Fabric Underlay

Underlay Considerations

Unicast Underlay:

The primary purpose of the underlay in the VXLAN EVPN fabric is to advertise the reachability of Virtual Tunnel End Points (VTEPs) and BGP peering addresses. The primary criterion for choosing an underlay protocol is fast convergence in the event of node failures. Other criteria are:

- Simplicity of configuration.
- Ability to delay the introduction of a node into the network on boot up.

This document details the two primary protocols supported and tested by Cisco, IS-IS and OSPF. It will also illustrate the use of the eBGP protocol as an underlay for the VXLAN EVPN fabric.

From an underlay/overlay perspective, the packet flow from a server to another over the Virtual Extensible LAN (VXLAN) fabric as mentioned below:

1. The server sends traffic to the source VXLAN tunnel endpoint (VTEP). The VTEP performs Layer-2 or Layer-3 communication based on the destination MAC and derives the nexthop (destination VTEP).

> **Note** When a packet is bridged, the target end host's MAC address is stamped in the DMAC field of the inner frame. When a packet is routed, the default gateway's MAC address is stamped in the DMAC field of the inner frame.

2. The VTEP encapsulates the traffic (frames) into VXLAN packets (overlay function – see Figure 1) and signals the underlay IP network.
3. Based on the underlay routing protocol, the packet is sent from the source VTEP to destination VTEP through the IP network (underlay function – see *Underlay Overview* figure).

4. The destination VTEP removes the VXLAN encapsulation (overlay function) and sends traffic to the intended server.

The VTEPs are a part of the underlay network as well since VTEPs need to be reachable to each other to send VXLAN encapsulated traffic across the IP underlay network.

The *Overlay Overview* and *Underlay Overview* images (below) depict the broad difference between an overlay and underlay. Since the focus is on the VTEPs, the spine switches are only depicted in the background. Note that, in real time, the packet flow from VTEP to VTEP traverses through the spine switches.

Figure 4: Overlay Overview

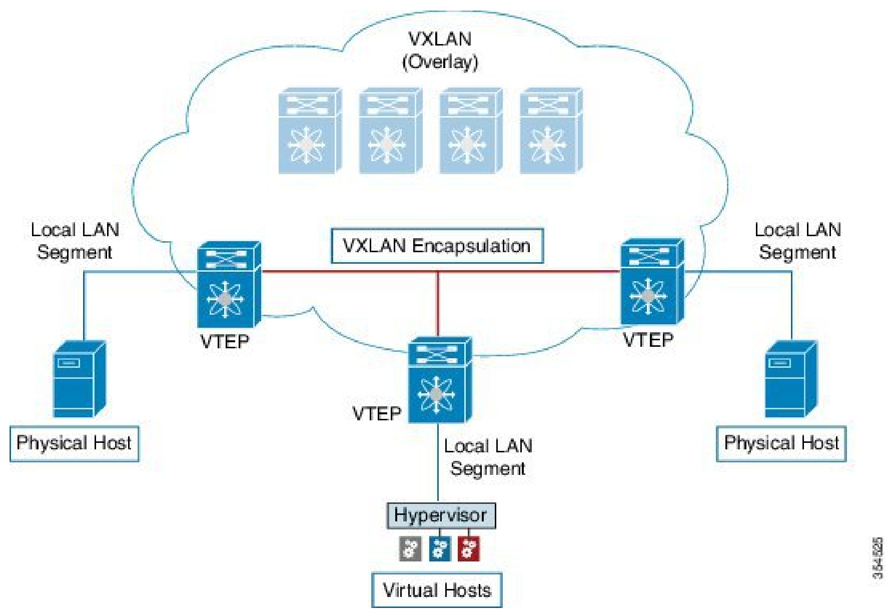
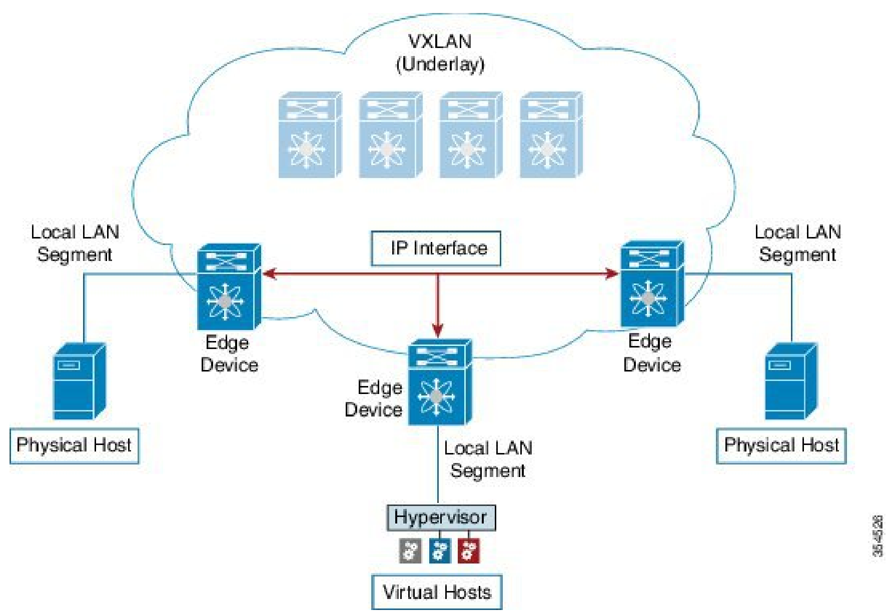


Figure 5: Underlay Overview



Deployment considerations for an underlay IP network in a VXLAN EVPN Programmable Fabric

The deployment considerations for an underlay IP network in a VXLAN EVPN Programmable Fabric are given below:

- Maximum transmission unit (MTU) – Due to VXLAN encapsulation, the MTU requirement is larger and we must avoid potential fragmentation.
  - An MTU of 9216 bytes on each interface on the path between the VTEPs accommodates maximum server MTU + VXLAN overhead. Most data center server NICs support up to 9000 bytes. So, no fragmentation is needed for VXLAN traffic.
  - The VXLAN IP fabric underlay supports the IPv4 address family.
- Unicast routing - Any unicast routing protocol can be used for the VXLAN IP underlay. You can implement OSPF, IS-IS, or eBGP to route between the VTEPs.

Note As a best practice, use a simple IGP (OSPF or IS-IS) for underlay reachability between VTEPs with iBGP for overlay information exchange.

- IP addressing – Point-to-point (P2P) or IP unnumbered links. For each point-to-point link, as example between the leaf switch nodes and spine switch nodes, typically a /30 IP mask should be assigned. Optionally a /31 mask or IP unnumbered links can be assigned. The IP unnumbered approach is leaner from an addressing perspective and consumes fewer IP addresses. The IP unnumbered option for the OSPF or IS-IS protocol underlay will minimize the use of IP addresses.

  /31 network - An OSPF or IS-IS point-to-point numbered network is only between two switch (interfaces), and there is no need for a broadcast or network address. So, a /31 network suffices for this network. Neighbors on this network establish adjacency and there is no designated router (DR) for the network.

Note IP Unnumbered for VXLAN underlay is supported starting with Cisco NX-OS Release 7.0(3)I7(2). Only a single unnumbered link between the same devices (for example, spine - leaf) is supported. If multiple physical links are connecting the same leaf and spine, you must use the single L3 port-channel with unnumbered link.

- Multicast protocol for multi-destination (BUM) traffic – Though VXLAN has the BGP EVPN control plane, the VXLAN fabric still requires a technology for Broadcast/Unknown unicast/Multicast (BUM) traffic to be forwarded.
- PIM Bidir is supported on Cisco Nexus 9300-EX/FX/FX2 platform switches.
- vPC configuration — This is documented in **Configuring vPCs** of *Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide*.

Unicast routing and IP addressing options

Each unicast routing protocol option (OSPF, IS-IS, and eBGP) and sample configurations are given below. Use an option to suit your setup's requirements.

Important All routing configuration samples are from an IP underlay perspective and are not comprehensive. For complete configuration information including routing process, authentication, Bidirectional Forwarding Detection (BFD) information, and so on, see *Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide*.

OSPF Underlay IP Network

Some considerations are given below:

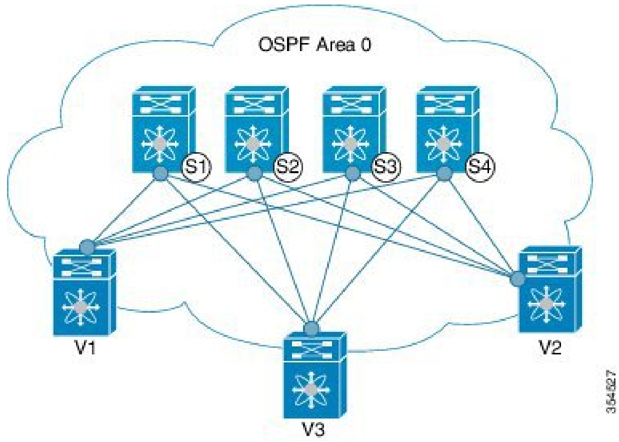
- For IP addressing, use P2P links. Since only two switches are directly connected, you can avoid a Designated Router/Backup Designated Router (DR/BDR) election.
- Use the *point-to-point* network type option. It is ideal for routed interfaces or ports, and is optimal from a Link State Advertisements (LSA) perspective.
- Do not use the broadcast type network. It is suboptimal from an LSA database perspective (LSA type 1 – Router LSA and LSA type 2 – Network LSA) and necessitates a DR/BDR election, thereby creating an additional election and database overhead.

Note You can divide OSPF networks into areas when the size of the routing domain contains a high number of routers and/or IP prefixes.. The same general well known OSPF best practice rules in regards of scale and configuration are applicable for the VXLAN underlay too. For example, LSA type 1 and type 2 are never flooded outside of an area. With multiple areas, the size of the OSPF LSA databases can be reduced to optimize CPU and memory consumption.

Note
- For ease of use, the configuration mode from which you need to start configuring a task is mentioned at the beginning of each configuration.
- Configuration tasks and corresponding show command output are displayed for a part of the topology in the image. For example, if the sample configuration is shown for a leaf switch and connected spine switch, the show command output for the configuration displays corresponding configuration.

OSPF configuration sample – P2P and IP unnumbered network scenarios

Figure 6: OSPF as the underlay routing protocol



OSPF – P2P link scenario with /31 mask

In the above image, the leaf switches (V1, V2, and V3) are at the bottom of the image. They are connected to the 4 spine switches (S1, S2, S3, and S4) that are depicted at the top of the image. For P2P connections between a leaf switch (also having VTEP function) and each spine, leaf switches V1, V2, and V3 should each be connected to each spine switch.

For V1, we should configure a P2P interface to connect to each spine switch.

A sample P2P configuration between a leaf switch (V1) interface and a spine switch (S1) interface is given below:

OSPF global configuration on leaf switch V1

(config) #

```
feature ospf
router ospf UNDERLAY
 router-id 10.1.1.54
```

OSPF leaf switch V1 P2P interface configuration

(config) #

```
interface Ethernet 1/41
   description Link to Spine S1
   no switchport
   ip address 198.51.100.1/31
   mtu 9192
   ip router ospf UNDERLAY area 0.0.0.0
   ip ospf network point-to-point
```

The **ip ospf network point-to-point** command configures the OSPF network as a point-to-point network

The OSPF instance is tagged as UNDERLAY for better recall.

OSPF loopback interface configuration (leaf switch V1)

Configure a loopback interface so that it can be used as the OSPF router ID of leaf switch V1.

(config) #

```
interface loopback 0
   ip address 10.1.1.54/32
   ip router ospf UNDERLAY area 0.0.0.0
```

The interface will be associated with the OSPF instance UNDERLAY and OSPF area 0.0.0.0

OSPF global configuration on spine switch S1

(config) #

```
feature ospf
router ospf UNDERLAY
 router-id 10.1.1.53
```

(Corresponding) OSPF spine switch S1 P2P interface configuration

(config) #

```
interface Ethernet 1/41
   description Link to VTEP V1
   ip address 198.51.100.2/31
   mtu 9192
   ip router ospf UNDERLAY area 0.0.0.0
   ip ospf network point-to-point
   no shutdown
```

Note MTU size of both ends of the link should be configured identically.

OSPF loopback Interface Configuration (spine switch S1)

Configure a loopback interface so that it can be used as the OSPF router ID of spine switch S1.

(config) #

```
interface loopback 0
   ip address 10.1.1.53/32
   ip router ospf UNDERLAY area 0.0.0.0
```

The interface will be associated with the OSPF instance UNDERLAY and OSPF area 0.0.0.0

.

.

To complete OSPF topology configuration for the 'OSPF as the underlay routing protocol' image, configure the following

- *3 more V1 interfaces (or 3 more P2P links) to the remaining 3 spine switches.*
- *Repeat the procedure to connect P2P links between V2,V3 and V4 and the spine switches.*

OSPF - IP unnumbered scenario

A sample OSPF IP unnumbered configuration is given below:

OSPF leaf switch V1 configuration

OSPF global configuration on leaf switch V1

(config) #

```
feature ospf
router ospf UNDERLAY
  router-id 10.1.1.54
```

The OSPF instance is tagged as UNDERLAY for better recall.

OSPF leaf switch V1 P2P interface configuration

(config) #

```
interface Ethernet1/41
  description Link to Spine S1
  mtu 9192
  ip ospf network point-to-point
  ip unnumbered loopback0
  ip router ospf UNDERLAY area 0.0.0.0
```

The **ip ospf network point-to-point** command configures the OSPF network as a point-to-point network.

OSPF loopback interface configuration

Configure a loopback interface so that it can be used as the OSPF router ID of leaf switch V1.

(config) #

```
interface loopback0
  ip address 10.1.1.54/32
  ip router ospf UNDERLAY area 0.0.0.0
```

The interface will be associated with the OSPF instance UNDERLAY and OSPF area 0.0.0.0

OSPF spine switch S1 configuration:

OSPF global configuration on spine switch S1

(config) #

```
feature ospf
router ospf UNDERLAY
  router-id 10.1.1.53
```

(Corresponding) OSPF spine switch S1 P2P interface configuration

(config) #

```
interface Ethernet1/41
  description Link to VTEP V1
  mtu 9192
  ip ospf network point-to-point
  ip unnumbered loopback0
  ip router ospf UNDERLAY area 0.0.0.0
```

OSPF loopback interface configuration (spine switch S1)

Configure a loopback interface so that it can be used as the OSPF router ID of spine switch S1.

(config) #

```
interface loopback0
  ip address 10.1.1.53/32
  ip router ospf UNDERLAY area 0.0.0.0
```

The interface will be associated with the OSPF instance UNDERLAY and OSPF area 0.0.0.0

.

.

To complete OSPF topology configuration for the ‘OSPF as the underlay routing protocol’ image, configure the following:

- *3 more VTEP V1 interfaces (or 3 more IP unnumbered links) to the remaining 3 spine switches.*
- *Repeat the procedure to connect IP unnumbered links between VTEPs V2,V3 and V4 and the spine switches.*

OSPF Verification

Use the following commands for verifying OSPF configuration:

```
Leaf-Switch-V1# show ip ospf

 Routing Process UNDERLAY with ID 10.1.1.54 VRF default
 Routing Process Instance Number 1
 Stateful High Availability enabled
 Graceful-restart is configured
   Grace period: 60 state: Inactive
   Last graceful restart exit status: None
 Supports only single TOS(TOS0) routes
 Supports opaque LSA
 Administrative distance 110
 Reference Bandwidth is 40000 Mbps
 SPF throttling delay time of 200.000 msecs,
   SPF throttling hold time of 1000.000 msecs,
   SPF throttling maximum wait time of 5000.000 msecs
 LSA throttling start time of 0.000 msecs,
   LSA throttling hold interval of 5000.000 msecs,
   LSA throttling maximum wait time of 5000.000 msecs
 Minimum LSA arrival 1000.000 msec
 LSA group pacing timer 10 secs
 Maximum paths to destination 8
 Number of external LSAs 0, checksum sum 0
 Number of opaque AS LSAs 0, checksum sum 0
 Number of areas is 1, 1 normal, 0 stub, 0 nssa
 Number of active areas is 1, 1 normal, 0 stub, 0 nssa
 Install discard route for summarized external routes.
 Install discard route for summarized internal routes.
   Area BACKBONE(0.0.0.0)
        Area has existed for 03:12:54
        Interfaces in this area: 2 Active interfaces: 2
        Passive interfaces: 0  Loopback interfaces: 1
        No authentication available
        SPF calculation has run 5 times
         Last SPF ran for 0.000195s
        Area ranges are
        Number of LSAs: 3, checksum sum 0x196c2
```

```
Leaf-Switch-V1# show ip ospf interface

loopback0 is up, line protocol is up
    IP address 10.1.1.54/32
    Process ID UNDERLAY VRF default, area 0.0.0.0
```

```
    Enabled by interface configuration
    State LOOPBACK, Network type LOOPBACK, cost 1
    Index 1
 Ethernet1/41 is up, line protocol is up
    Unnumbered interface using IP address of loopback0 (10.1.1.54)
    Process ID UNDERLAY VRF default, area 0.0.0.0
    Enabled by interface configuration
    State P2P, Network type P2P, cost 4
    Index 2, Transmit delay 1 sec
    1 Neighbors, flooding to 1, adjacent with 1
    Timer intervals: Hello 10, Dead 40, Wait 40, Retransmit 5
      Hello timer due in 00:00:07
    No authentication
    Number of opaque link LSAs: 0, checksum sum 0
```

```
Leaf-Switch-V1# show ip ospf neighbors

OSPF Process ID UNDERLAY VRF default
 Total number of neighbors: 1
 Neighbor ID     Pri State            Up Time  Address         Interface
 10.1.1.53         1 FULL/ -          06:18:32 10.1.1.53        Eth1/41
```

For a detailed list of commands, refer to the Configuration and Command Reference guides.

IS-IS Underlay IP Network

Some considerations are given below:

- Because IS-IS uses Connectionless Network Service (CLNS) and is independent of the IP, full SPF calculation is avoided when a link changes.
- Net ID - Each IS-IS instance has an associated network entity title (NET) ID that uniquely identifies the IS-IS instance in the area. The NET ID is comprised of the IS-IS system ID, which uniquely identifies this IS-IS instance in the area, and the area ID. For example, if the NET ID is 49.0001.0010.0100.1074.00, the system ID is 0010.0100.1074 and the area ID is 49.0001.

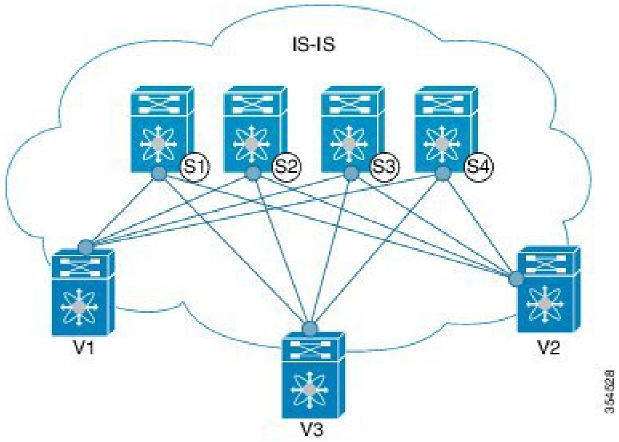
Important **Level 1 IS-IS in the Fabric**—Cisco has validated the use of IS-IS Level 1 only and IS-IS Level 2 only configuration on all nodes in the programmable fabric. The fabric is considered a stub network where every node needs an optimal path to every other node in the fabric. Cisco NX-OS IS-IS implementation scales well to support a number of nodes in a fabric. Hence, there is no anticipation of having to break up the fabric into multiple IS-IS domains.

Note

- For ease of use, the configuration mode from which you need to start configuring a task is mentioned at the beginning of each configuration.
- Configuration tasks and corresponding show command output are displayed for a part of the topology in the image. For example, if the sample configuration is shown for a leaf switch and connected spine switch, the show command output for the configuration displays corresponding configuration.

IS-IS configuration sample - P2P and IP unnumbered network scenarios

Figure 7: IS-IS as the underlay routing protocol



In the above image, the leaf switches (V1, V2, and V3, having the VTEP function) are at the bottom of the image. They are connected to the 4 spine switches (S1, S2, S3, and S4) that are depicted at the top of the image.

IS-IS – P2P link scenario with /31 mask

A sample P2P configuration between V1 and spine switch S1 is given below:

For P2P connections between a leaf switch and each spine switch, V1, V2, and V3 should each be connected to each spine switch.

For V1, we must configure a loopback interface and a P2P interface configuration to connect to S1. A sample P2P configuration between a leaf switch (V1) interface and a spine switch (S1) interface is given below:

IS-IS configuration on leaf switch V1

IS-IS global configuration

(config) #

```
feature isis
router isis UNDERLAY
  net 49.0001.0010.0100.1074.00
  is-type level-1
  set-overload-bit on-startup 60
```

Setting the overload bit - You can configure a Cisco Nexus switch to signal other devices not to use the switch as an intermediate hop in their shortest path first (SPF) calculations. You can optionally configure the overload bit temporarily on startup. In the above example, the **set-overload-bit** command is used to set the overload bit on startup to 60 seconds.

IS-IS P2P interface configuration (leaf switch V1)

(config) #

```
interface Ethernet 1/41
   description Link to Spine S1
   mtu 9192
   ip address 209.165.201.1/31
```

```
   ip router isis UNDERLAY
```

IS-IS loopback interface configuration (leaf switch V1)

Configure a loopback interface so that it can be used as the IS-IS router ID of leaf switch V1.

(config) #

```
interface loopback 0
   ip address 10.1.1.74/32
   ip router isis UNDERLAY
```

The IS-IS instance is tagged as UNDERLAY for better recall.

(Corresponding) IS-IS spine switch S1 configuration

IS-IS global configuration

(config) #

```
feature isis
router isis UNDERLAY
 net 49.0001.0010.0100.1053.00
 is-type level-1
 set-overload-bit on-startup 60
```

IS-IS P2P interface configuration (spine switch S1)

(config) #

```
interface Ethernet 1/1
   description Link to VTEP V1
   ip address 209.165.201.2/31
   mtu 9192
   ip router isis UNDERLAY
```

IS-IS loopback interface configuration (spine switch S1)

(config) #

```
interface loopback 0
   ip address 10.1.1.53/32
   ip router isis UNDERLAY
.
.
```

To complete IS-IS topology configuration for the above image, configure the following:

- 3 more leaf switch V1's interfaces (or 3 more P2P links) to the remaining 3 spine switches.
- Repeat the procedure to connect P2P links between leaf switches V2, V3 and V4 and the spine switches.

IS-IS - IP unnumbered scenario

IS-IS configuration on leaf switch V1

IS-IS global configuration

(config)#

```
feature isis
router isis UNDERLAY
  net 49.0001.0010.0100.1074.00
  is-type level-1
  set-overload-bit on-startup 60
```

IS-IS interface configuration (leaf switch V1)

(config) #

```
interface Ethernet1/41
  description Link to Spine S1
  mtu 9192
  medium p2p
  ip unnumbered loopback0
  ip router isis UNDERLAY
```

IS-IS loopback interface configuration (leaf switch V1)

(config)

```
interface loopback0
  ip address 10.1.1.74/32
  ip router isis UNDERLAY
```

IS-IS configuration on the spine switch S1

IS-IS global configuration

(config)#

```
feature isis
router isis UNDERLAY
  net 49.0001.0010.0100.1053.00
  is-type level-1
  set-overload-bit on-startup 60
```

IS-IS interface configuration (spine switch S1)

(config)#

```
interface Ethernet1/41
  description Link to V1
  mtu 9192
  medium p2p
  ip unnumbered loopback0
  ip router isis UNDERLAY
```

IS-IS loopback interface configuration (spine switch S1)

(config)#

```
interface loopback0
  ip address 10.1.1.53/32
  ip router isis UNDERLAY
```

IS-IS Verification

Use the following commands for verifying IS-IS configuration on leaf switch V1:

```
Leaf-Switch-V1# show isis

ISIS process : UNDERLAY
 Instance number :  1
 UUID: 1090519320
 Process ID 20258
VRF: default
  System ID : 0010.0100.1074  IS-Type : L1
  SAP : 412  Queue Handle : 15
  Maximum LSP MTU: 1492
  Stateful HA enabled
  Graceful Restart enabled. State: Inactive
  Last graceful restart status : none
  Start-Mode Complete
  BFD IPv4 is globally disabled for ISIS process: UNDERLAY
  BFD IPv6 is globally disabled for ISIS process: UNDERLAY
  Topology-mode is base
  Metric-style : advertise(wide), accept(narrow, wide)
  Area address(es) :
    49.0001
 Process is up and running
  VRF ID: 1
  Stale routes during non-graceful controlled restart
  Interfaces supported by IS-IS :
    loopback0
    loopback1
    Ethernet1/41
  Topology : 0
  Address family IPv4 unicast :
    Number of interface : 2
    Distance : 115
  Address family IPv6 unicast :
    Number of interface : 0
    Distance : 115
  Topology : 2
  Address family IPv4 unicast :
    Number of interface : 0
    Distance : 115
  Address family IPv6 unicast :
    Number of interface : 0
    Distance : 115
   Level1
  No auth type and keychain
  Auth check set
  Level2
  No auth type and keychain
  Auth check set
  L1 Next SPF: Inactive
  L2 Next SPF: Inactive

Leaf-Switch-V1# show isis interface

IS-IS process: UNDERLAY VRF: default
loopback0, Interface status: protocol-up/link-up/admin-up IP address: 10.1.1.74, IP subnet:
 10.1.1.74/32
IPv6 routing is disabled Level1
No auth type and keychain Auth check set
Level2
No auth type and keychain Auth check set
Index: 0x0001, Local Circuit ID: 0x01, Circuit Type: L1 BFD IPv4 is locally disabled for
Interface loopback0 BFD IPv6 is locally disabled for Interface loopback0 MTR is disabled
```

```
Level Metric 1  1
2 1
Topologies enabled:
    L  MT  Metric  MetricCfg  Fwdng IPV4-MT  IPV4Cfg  IPV6-MT  IPV6Cfg
    1  0        1       no   UP    UP       yes      DN       no
    2  0        1       no   DN    DN       no       DN       no

loopback1, Interface status: protocol-up/link-up/admin-up
  IP address: 10.1.2.74, IP subnet: 10.1.2.74/32
  IPv6 routing is disabled
  Level1
    No auth type and keychain
    Auth check set
  Level2
    No auth type and keychain
    Auth check set
  Index: 0x0002, Local Circuit ID: 0x01, Circuit Type: L1
  BFD IPv4 is locally disabled for Interface loopback1
  BFD IPv6 is locally disabled for Interface loopback1
  MTR is disabled
  Passive level: level-2
  Level      Metric
  1              1
  2              1
  Topologies enabled:
    L  MT  Metric  MetricCfg  Fwdng IPV4-MT  IPV4Cfg  IPV6-MT  IPV6Cfg
    1  0        1       no   UP    UP       yes      DN       no
    2  0        1       no   DN    DN       no       DN       no

Ethernet1/41, Interface status: protocol-up/link-up/admin-up
  IP unnumbered interface (loopback0)
  IPv6 routing is disabled
    No auth type and keychain
    Auth check set
  Index: 0x0002, Local Circuit ID: 0x01, Circuit Type: L1
  BFD IPv4 is locally disabled for Interface Ethernet1/41
  BFD IPv6 is locally disabled for Interface Ethernet1/41
  MTR is disabled
  Extended Local Circuit ID: 0x1A028000, P2P Circuit ID: 0000.0000.0000.00
  Retx interval: 5, Retx throttle interval: 66 ms
  LSP interval: 33 ms, MTU: 9192
  P2P Adjs: 1, AdjsUp: 1, Priority 64
  Hello Interval: 10, Multi: 3, Next IIH: 00:00:01
  MT    Adjs   AdjsUp  Metric   CSNP  Next CSNP  Last LSP ID
  1          1        1       4     60  00:00:35   ffff.ffff.ffff.ff-ff
  2          0        0       4     60  Inactive   ffff.ffff.ffff.ff-ff
  Topologies enabled:
    L  MT  Metric  MetricCfg  Fwdng IPV4-MT  IPV4Cfg  IPV6-MT  IPV6Cfg
    1  0        4       no   UP    UP       yes      DN       no
    2  0        4       no   UP    DN       no       DN       no

Leaf-Switch-V1# show isis adjacency

IS-IS process: UNDERLAY VRF: default
IS-IS adjacency database:
Legend: '!': No AF level connectivity in given topology
System ID          SNPA    Level  State Hold Time  Interface
Spine-Switch-S1    N/A     1      UP    00:00:23   Ethernet1/41
```

For a detailed list of commands, refer to the Configuration and Command Reference guides.

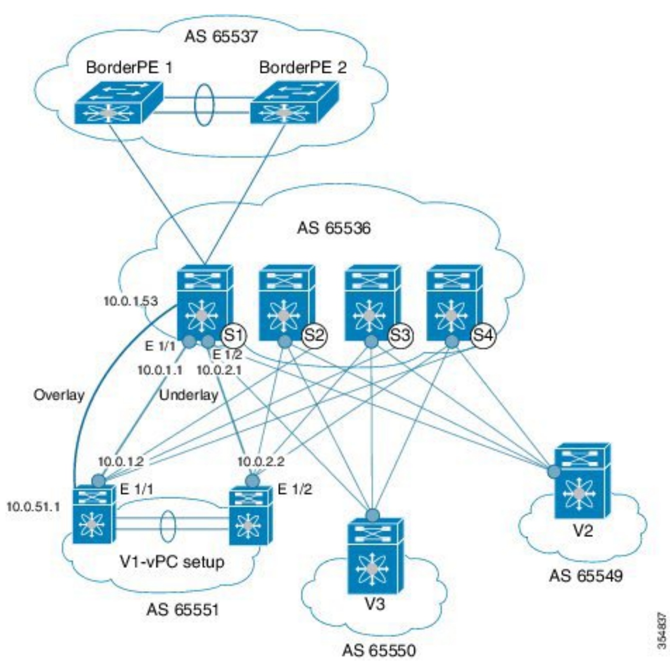
eBGP Underlay IP Network

Some customers would like to have the same protocol in the underlay and overlay in order to contain the number of protocols that need support in their network.

There are various ways to configure the eBGP based underlay. The configurations given in this section have been validated for function and convergence. The IP underlay based on eBGP can be built with these configurations detailed below. (For reference, see image below)

- The design below is following the multi AS model.
- eBGP underlay requires numbered interfaces between leaf and spine nodes. Numbered interfaces are used for the underlay BGP sessions as there is no other protocol to distribute peer reachability.
- The overlay sessions are configured on loopback addresses. This is to increase the resiliency in presence of link or node failures.
- BGP speakers on spine layer configure all leaf node eBGP neighbors individually. This is different from IBGP based peering which can be covered by dynamic BGP.
- Pointers for Multiple AS numbers in a fabric are given below:
    - All spine nodes configured as BGP speakers are in one AS.
    - All leaf nodes will have a unique AS number that is different than the BGP speakers in spine layer.
    - A pair of vPC leaf switch nodes, have the same AS number.
    - If a globally unique AS number is required to represent the fabric, then that can be configured on the border leaf or borderPE switches. All other nodes can use the private AS number range.
    - BGP Confederation has not been leveraged.

Figure 8: eBGP as underlay



eBGP configuration sample

Sample configurations for a spine switch and leaf switch are given below. The complete configuration is given for providing context, and the configurations added specifically for eBGP underlay are highlighted and further explained.

There is one BGP session per neighbor to set up the underlay. This is done within the global IPv4 address family. The session is used to distribute the loopback addresses for VTEP, Rendezvous Point (RP) and the eBGP peer address for the overlay eBGP session.

Spine switch S1 configuration—On the spine switch (S1 in this example), all leaf nodes are configured as eBGP neighbors.

(config) #

```
router bgp 65536
   router-id 10.1.1.53
   address-family ipv4 unicast
   redistribute direct route-map DIRECT-ROUTES-MAP
```

The **redistribute direct** command is used to advertise the loopback addresses for BGP and VTEP peering. It can be used to advertise any other direct routes in the global address space. The route map can filter the advertisement to include only eBGP peering and VTEP loopback addresses.

```
   maximum-paths 2
   address-family l2vpn evpn
     retain route-target all
```

Spine switch BGP speakers don’t have any VRF configuration. Hence, the **retain route-target all** command is needed to retain the routes and send them to leaf switch VTEPs. The **maximum-paths** command is used for ECMP path in the underlay.

Underlay session towards leaf switch V1 (vPC set up) —As mentioned above, the underlay sessions are configured on the numbered interfaces between spine and leaf switch nodes.

(config) #

```
neighbor 10.0.1.2 remote-as 65551
   address-family ipv4 unicast
      disable-peer-as-check
      send-community both
```

The vPC pair of switches has the same AS number. The **disable-peer-as-check** command is added to allow route propagation between the vPC switches as they are configured with the same AS, for example, for route type 5 routes. If the vPC switches have different AS numbers, this command is not required.

Underlay session towards the border leaf switch—The underlay configurations towards leaf and border leaf switches are the same, barring the changes in IP address and AS values.

Overlay session on the spine switch S1 towards the leaf switch V1

(config) #

```
route-map UNCHANGED permit 10
```

```
set ip next-hop unchanged
```

Note The route-map UNCHANGED is user defined whereas the keyword **unchanged** is an option within the **set ip next-hop** command. In eBGP, the next hop is changed to self when sending a route from one eBGP neighbor to another. The route map UNCHANGED is added to make sure that, for overlay routes, the originating leaf switch is set as next hop and not the spine switch. This ensures that VTEPs are next hops, and not spine switch nodes. The **unchanged** keyword ensures that the next-hop attribute in the BGP update to the eBGP peer is unmodified.

The overlay sessions are configured on loopback addresses.

(config) #

```
neighbor 10.0.51.1 remote-as 65551
   update-source loopback0
   ebgp-multihop 2
   address-family l2vpn evpn
     rewrite-evpn-rt-asn
       disable-peer-as-check
       send-community both
       route-map UNCHANGED out
```

The spine switch configuration concludes here. The *Route Target auto* feature configuration is given below for reference purposes:

(config) #

```
vrf context coke
    vni 50000
    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
```

The **rewrite-evpn-rt-asn** command is required if the *Route Target auto* feature is being used to configure EVPN RTs.

Route target auto is derived from the Local AS number configured on the switch and the Layer-3 VNID of the VRF i.e. Local AS:VNID. In Multi-AS topology, as illustrated in this guide, each leaf node is represented as a different local AS, and the route target generated for the same VRF will be different on every switch. The command **rewrite-evpn-rt-asn** replaces the ASN portion of the route target in the BGP update message with the local AS number. For example, if VTEP V1 has a Local AS 65551, VTEP V2 has a Local AS 65549, and spine switch S1 has a Local AS 65536, then the route targets for V1, V2 and S1 are as follows:

- V1—65551:50000
- V2—65549:50000
- S1—65536:50000

In this scenario, V2 advertises the route with RT 65549:50000, the spine switch S1 replaces it with RT 65536:50000, and finally when V1 gets the update, it replaces the route target in the update with 65551:50000. This matches the locally configured RT on V1. This command requires that it be configured on all BGP speakers in the fabric.

If the *Route Target auto* feature is not being used, i.e., matching RTs are required to be manually configured on all switches, then this command is not necessary.

Leaf switch VTEP V1 configuration—In the sample configuration below, VTEP V1's interfaces are designated as BGP neighbors. All leaf switch VTEPs including border leaf switch nodes have the following configurations towards spine switch neighbor nodes:

(config) #

```
router bgp 65551
  router-id 10.1.1.54
  address-family ipv4 unicast
     maximum-paths 2
     address-family l2vpn evpn
```

The **maximum-paths** command is used for ECMP path in the underlay.

Underlay session on leaf switch VTEP V1 towards spine switch S1

(config) #

```
neighbor 10.0.1.1 remote-as 65536
   address-family ipv4 unicast
     allowas-in
     send-community both
```

The **allowas-in** command is needed if leaf switch nodes have the same AS. In particular, the Cisco validated topology had a vPC pair of switches share an AS number.

Overlay session towards spine switch S1

(config) #

```
neighbor 10.1.1.53 remote-as 65536
   update-source loopback0
   ebgp-multihop 2
   address-family l2vpn evpn
   rewrite-evpn-rt-asn
     allowas-in
     send-community both
```

The **ebgp-multihop 2** command is needed as the peering for the overlay is on the loopback address. NX-OS considers that as multi hop even if the neighbor is one hop away.

vPC backup session

(config) #

```
route-map SET-PEER-AS-NEXTHOP permit 10
  set ip next-hop peer-address
```

```
neighbor 192.168.0.1 remote-as 65551
    update-source Vlan3801
    address-family ipv4 unicast
      send-community both
      route-map SET-PEER-AS-NEXTHOP out
```

Note This session is configured on the backup SVI between the vPC leaf switch nodes.

To complete configurations for the above image, configure the following:

- *V1 as a BGP neighbor to other spine switches.*
- *Repeat the procedure for other leaf switches.*

BGP Verification

Use the following commands for verifying BGP configuration:

```
show bgp all
show bgp ipv4 unicast neighbors
show ip route bgp
```

For a detailed list of commands, refer to the Configuration and Command Reference guides.

Multicast Routing in the VXLAN Underlay

The VXLAN EVPN Programmable Fabric supports multicast routing for transporting BUM (broadcast, unknown unicast and multicast) traffic.

Refer the table below to know the multicast protocol(s) your Cisco Nexus switches support:

Cisco Nexus Series Switch(es) Combination	Multicast Routing Option
Cisco Nexus 7000/7700 Series switches with Cisco Nexus 9000 Series switches	PIM ASM (Sparse Mode)
Cisco Nexus 9000 Series	PIM ASM (Sparse Mode) *or* PIM BiDir **Note** PIM BiDir is supported on Cisco Nexus 9300-EX and 9300-FX/FX2 platform switches.

You can transport BUM traffic without multicast, through *ingress replication* . Ingress replication is currently available on Cisco Nexus 9000 Series switches.

PIM ASM and PIM Bidir Underlay IP Network

Some multicast topology design pointers are given below:

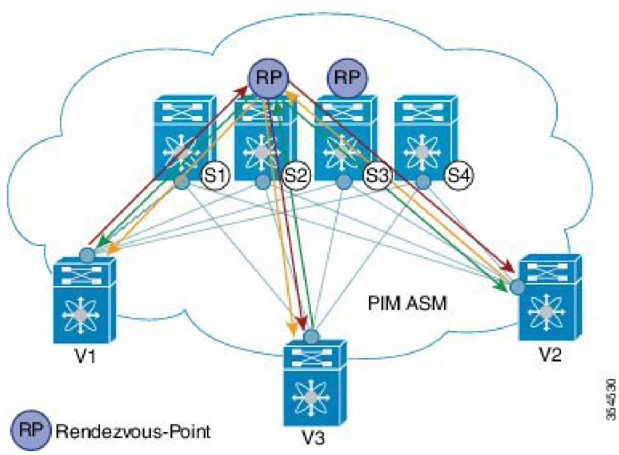
- Use spine/aggregation switches as Rendezvous-Point locations.

- Reserve a range of multicast groups (destination groups/DGroups) to service the overlay and optimize for diverse VNIs.
- In a spine-leaf topology with a lean spine,
  - Use multiple Rendezvous-Points across multiple spine switches.
  - Use redundant Rendezvous-Points.
  - Map different VNIs to different multicast groups, which are mapped to different Rendezvous-Points for load balancing.

Important The following configuration samples are from an IP underlay perspective and are not comprehensive. Functions such as PIM authentication, BFD for PIM, etc, are not shown here. Refer to the respective Cisco Nexus Series switch multicast configuration guide for complete information.

PIM Sparse-Mode (Any-Source Multicast [ASM])

Figure 9: PIM ASM as the IP multicast routing protocol



PIM ASM is supported on the Cisco Nexus 9000 series as the underlay multicast protocol.

In the above image, the leaf switches (V1, V2, and V3 having VTEP configuration) are at the bottom of the image. They are connected to the 4 spine switches (S1, S2, S3, and S4) that are depicted at the top of the image.

Two multicast Rendezvous-Points (S2 and S3) are configured. The second Rendezvous-Point is added for load sharing and redundancy purposes. *Anycast RP is represented in the PIM ASM topology image*. Anycast RP ensures redundancy and load sharing between the two Rendezvous-Points. To use Anycast RP, multiple spines serving as RPs will share the same IP address (the Anycast RP address). Meanwhile, each RP has its unique IP address added in the RP set for RPs to sync information with respect to sources between all spines which act as RPs.

The shared multicast tree is unidirectional, and uses the Rendezvous-Point for forwarding packets.

PIM ASM at a glance - 1 source tree per multicast group per leaf switch.

Programmable Fabric specific pointers are:

- All VTEPs that serve a VNI join a shared multicast tree. VTEPs V1, V2, and V3 have hosts attached from a single tenant (say x) and these VTEPs form a separate multicast (source, group) tree.
- A VTEP (say V1) might have hosts belonging to other tenants too. Each tenant may have different multicast groups associated with. A source tree is created for each tenant residing on the VTEP, if the tenants do not share a multicast group.

PIM ASM Configuration

Note For ease of use, the configuration mode from which you need to start configuring a task is mentioned at the beginning of each configuration.

Configuration tasks and corresponding show command output are displayed for a part of the topology in the image. For example, if the sample configuration is shown for a leaf switch and connected spine switch, the show command output for the configuration only displays corresponding configuration.

Leaf switch V1 Configuration — Configure RP reachability on the leaf switch.

PIM Anycast Rendezvous-Point association on leaf switch V1

(config) #

```
feature pim
ip pim rp-address 198.51.100.220 group-list 224.1.1.1
```

198.51.100.220 is the Anycast Rendezvous-Point IP address.

Loopback interface PIM configuration on leaf switch V1

(config) #

```
interface loopback 0
  ip address 209.165.201.20/32
  ip pim sparse-mode
```

Point-2-Point (P2P) interface PIM configuration for leaf switch V1 to spine switch S2 connectivity

(config) #

```
interface Ethernet 1/1
  no switchport
  ip address 209.165.201.14/31
  mtu 9216
  ip pim sparse-mode
.
.
```

Repeat the above configuration for a P2P link between V1 and the spine switch (S3) acting as the redundant Anycast Rendezvous-Point.

The VTEP also needs to be connected with spine switches (S1 and S4) that are not rendezvous points. A sample configuration is given below:

Point-2-Point (P2P) interface configuration for leaf switch V1 to non-rendezvous point spine switch (S1) connectivity

(config) #

```
interface Ethernet 2/2
  no switchport
  ip address 209.165.201.10/31
  mtu 9216
  ip pim sparse-mode
```

Repeat the above configuration for all P2P links between V1 and non- rendezvous point spine switches.

Repeat the complete procedure given above to configure all other leaf switches.

Rendezvous Point Configuration on the spine switches

PIM configuration on spine switch S2

(config) #

```
feature pim
```

Loopback Interface Configuration (RP)

(config) #

```
interface loopback 0
 ip address 10.10.100.100/32
 ip pim sparse-mode
```

Loopback interface configuration (Anycast RP)

(config) #

```
interface loopback 1
  ip address 198.51.100.220/32
  ip pim sparse-mode
```

Anycast-RP configuration on spine switch S2

Configure a spine switch as a Rendezvous Point and associate it with the loopback IP addresses of switches S2 and S3 for redundancy.

(config) #

```
feature pim
ip pim rp-address 198.51.100.220 group-list 224.1.1.1
ip pim anycast-rp 198.51.100.220 10.10.100.100
ip pim anycast-rp 198.51.100.220 10.10.20.100
```

```
.
.
```

Note The above configurations should also be implemented on the other spine switch (S3) performing the role of RP.

Non-RP Spine Switch Configuration

You also need to configure PIM ASM on spine switches that are not designated as rendezvous points, namely S1 and S4.

Earlier, leaf switch (VTEP) V1 has been configured for a P2P link to a non RP spine switch. A sample configuration on the non RP spine switch is given below.

PIM ASM global configuration on spine switch S1 (non RP)

(config) #

```
feature pim
ip pim rp-address 198.51.100.220 group-list 224.1.1.1
```

Loopback interface configuration (non RP)

(config) #

```
interface loopback 0
  ip address 10.10.100.103/32
  ip pim sparse-mode
```

Point-2-Point (P2P) interface configuration for spine switch S1 to leaf switch V1 connectivity

(config) #

```
interface Ethernet 2/2
  no switchport
  ip address 209.165.201.15/31
  mtu 9216
  ip pim sparse-mode
.
.
```

Repeat the above configuration for all P2P links between the non- rendezvous point spine switches and other leaf switches (VTEPs).

PIM ASM Verification

Use the following commands for verifying PIM ASM configuration:

```
Leaf-Switch-V1# show ip mroute 224.1.1.1

IP Multicast Routing Table for VRF "default"
```

```
(*, 224.1.1.1/32), uptime: 02:21:20, nve ip pim
  Incoming interface: Ethernet1/1, RPF nbr: 10.10.100.100
  Outgoing interface list: (count: 1)
    nve1, uptime: 02:21:20, nve

(10.1.1.54/32, 224.1.1.1/32), uptime: 00:08:33, ip mrib pim
  Incoming interface: Ethernet1/2, RPF nbr: 209.165.201.12
  Outgoing interface list: (count: 1)
    nve1, uptime: 00:08:33, mrib

(10.1.1.74/32, 224.1.1.1/32), uptime: 02:21:20, nve mrib ip pim
  Incoming interface: loopback0, RPF nbr: 10.1.1.74
  Outgoing interface list: (count: 1)
    Ethernet1/6, uptime: 00:29:19, pim

Leaf-Switch-V1# show ip pim rp

PIM RP Status Information for VRF "default"
BSR disabled
Auto-RP disabled
BSR RP Candidate policy: None
BSR RP policy: None
Auto-RP Announce policy: None
Auto-RP Discovery policy: None

RP: 198.51.100.220, (0), uptime: 03:17:43, expires: never,
  priority: 0, RP-source: (local), group ranges:
      224.0.0.0/9

Leaf-Switch-V1# show ip pim interface

PIM Interface Status for VRF "default"
Ethernet1/1, Interface status: protocol-up/link-up/admin-up
  IP address: 209.165.201.14, IP subnet: 209.165.201.14/31
  PIM DR: 209.165.201.12, DR's priority: 1
  PIM neighbor count: 1
  PIM hello interval: 30 secs, next hello sent in: 00:00:11
  PIM neighbor holdtime: 105 secs
  PIM configured DR priority: 1
  PIM configured DR delay: 3 secs
  PIM border interface: no
  PIM GenID sent in Hellos: 0x33d53dc1
  PIM Hello MD5-AH Authentication: disabled
  PIM Neighbor policy: none configured
  PIM Join-Prune inbound policy: none configured
  PIM Join-Prune outbound policy: none configured
  PIM Join-Prune interval: 1 minutes
  PIM Join-Prune next sending: 1 minutes
  PIM BFD enabled: no
  PIM passive interface: no
  PIM VPC SVI: no
  PIM Auto Enabled: no
  PIM Interface Statistics, last reset: never
    General (sent/received):
      Hellos: 423/425 (early: 0), JPs: 37/32, Asserts: 0/0
      Grafts: 0/0, Graft-Acks: 0/0
      DF-Offers: 4/6, DF-Winners: 0/197, DF-Backoffs: 0/0, DF-Passes: 0/0
    Errors:
      Checksum errors: 0, Invalid packet types/DF subtypes: 0/0
      Authentication failed: 0
      Packet length errors: 0, Bad version packets: 0, Packets from self: 0
      Packets from non-neighbors: 0
          Packets received on passiveinterface: 0
      JPs received on RPF-interface: 0
      (*,G) Joins received with no/wrong RP: 0/0
```

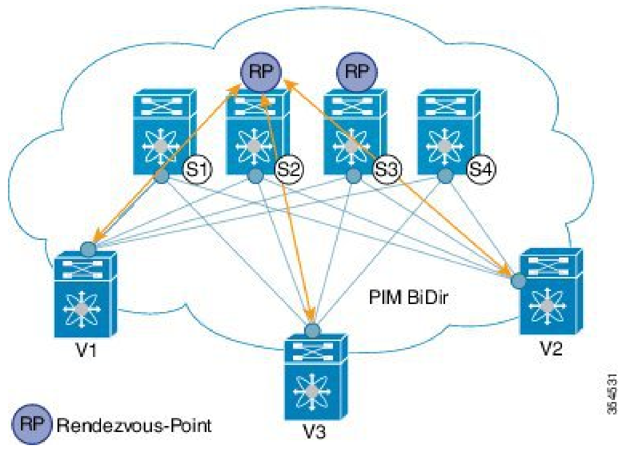
```
      (*,G)/(S,G) JPs received for SSM/Bidir groups: 0/0
      JPs filtered by inbound policy: 0
      JPs filtered by outbound policy: 0
loopback0, Interface status: protocol-up/link-up/admin-up
  IP address: 209.165.201.20, IP subnet: 209.165.201.20/32
  PIM DR: 209.165.201.20, DR's priority: 1
  PIM neighbor count: 0
  PIM hello interval: 30 secs, next hello sent in: 00:00:07
  PIM neighbor holdtime: 105 secs
  PIM configured DR priority: 1
  PIM configured DR delay: 3 secs
  PIM border interface: no
  PIM GenID sent in Hellos: 0x1be2bd41
  PIM Hello MD5-AH Authentication: disabled
  PIM Neighbor policy: none configured
  PIM Join-Prune inbound policy: none configured
  PIM Join-Prune outbound policy: none configured
  PIM Join-Prune interval: 1 minutes
  PIM Join-Prune next sending: 1 minutes
  PIM BFD enabled: no
  PIM passive interface: no
  PIM VPC SVI: no
  PIM Auto Enabled: no
  PIM Interface Statistics, last reset: never
    General (sent/received):
      Hellos: 419/0 (early: 0), JPs: 2/0, Asserts: 0/0
      Grafts: 0/0, Graft-Acks: 0/0
      DF-Offers: 3/0, DF-Winners: 0/0, DF-Backoffs: 0/0, DF-Passes: 0/0
    Errors:
      Checksum errors: 0, Invalid packet types/DF subtypes: 0/0
      Authentication failed: 0
      Packet length errors: 0, Bad version packets: 0, Packets from self: 0
     Packets from non-neighbors: 0
          Packets received on passiveinterface: 0
      JPs received on RPF-interface: 0
      (*,G) Joins received with no/wrong RP: 0/0
      (*,G)/(S,G) JPs received for SSM/Bidir groups: 0/0
      JPs filtered by inbound policy: 0
      JPs filtered by outbound policy: 0
```

```
Leaf-Switch-V1# show ip pim neighbor

PIM Neighbor Status for VRF "default"

Neighbor       Interface      Uptime    Expires    DR         Bidir-       BFD
                                                   Priority   Capable      State
10.10.100.100  Ethernet1/1    1w1d      00:01:33   1          yes          n/a
```

For a detailed list of commands, refer to the Configuration and Command Reference guides.

PIM Bidirectional (BiDir)

Figure 10: PIM BiDir as the IP multicast routing protocol



VXLAN BiDir underlay is supported on Cisco Nexus 9300-EX and 9300-FX/FX2/FX3 platform switches.

In the above image, the leaf switches (V1, V2, and V3) are at the bottom of the image. They are connected to the 4 spine switches (S1, S2, S3, and S4) that are depicted at the top of the image. The two PIM Rendezvous-Points using phantom RP mechanism are used for load sharing and redundancy purposes.

Note Load sharing happens only via different multicast groups, for the respective, different VNI.

With bidirectional PIM, one bidirectional, shared tree rooted at the RP is built for each multicast group. Source specific state are not maintained within the fabric which provides a more scalable solution.

Programmable Fabric specific pointers are:

- The 3 VTEPs share the same VNI and multicast group mapping to form a single multicast group tree.

PIM BiDir at a glance — *One shared tree per multicast group*.

PIM BiDir Configuration

The following is a configuration example of having two spine switches S2 and S3 serving as RPs using phantom RP for redundancy and loadsharing. Here S2 is the primary RP for group-list 227.2.2.0/26 and secondary for group-list 227.2.2.64/26. S3 is the primary RP for group-list 227.2.2.64/26 and secondary RP for group-list 227.2.2.0/26.

Note Phantom RP is used in a PIM BiDir environment where RP redundancy is designed using loopback networks with different mask lengths in the primary and secondary routers. These loopback interfaces are in the same subnet as the RP address, but with different IP addresses from the RP address. (Since the IP address advertised as RP address is not defined on any routers, the term phantom is used). The subnet of the loopback is advertised in the Interior Gateway Protocol (IGP). To maintain RP reachability, it is only necessary to ensure that a route to the RP exists.

Unicast routing longest match algorithms are used to pick the primary over the secondary router.

The primary router announces a longest match route (say, a /30 route for the RP address) and is preferred over the less specific route announced by the secondary router (a /29 route for the same RP address). The primary router advertises the /30 route of the RP, while the secondary router advertises the /29 route. The latter is only chosen when the primary router goes offline. We will be able to switch from the primary to the secondary RP at the speed of convergence of the routing protocol.

For ease of use, the configuration mode from which you need to start configuring a task is mentioned at the beginning of each configuration.

Configuration tasks and corresponding show command output are displayed for a part of the topology in the image. For example, if the sample configuration is shown for a leaf switch and connected spine switch, the show command output for the configuration only displays corresponding configuration.

Leaf switch V1 configuration

Phantom Rendezvous-Point association on leaf switch V1

(config) #

```
feature pim
ip pim rp-address 10.254.254.1 group-list 227.2.2.0/26 bidir
ip pim rp-address 10.254.254.65 group-list 227.2.2.64/26 bidir
```

Loopback interface PIM configuration on leaf switch V1

(config) #

```
interface loopback 0
  ip address 10.1.1.54/32
  ip pim sparse-mode
```

IP unnumbered P2P interface configuration on leaf switch V1

(config) #

```
interface Ethernet 1/1
  no switchport
  mtu 9192
  medium p2p
  ip unnumbered loopback 0
  ip pim sparse-mode
```

```
interface Ethernet 2/2
  no switchport
  mtu 9192
  medium p2p
  ip unnumbered loopback 0
  ip pim sparse-mode
```

Rendezvous Point configuration (on the two spine switches S2 and S3 acting as RPs)

Using phantom RP on spine switch S2

(config) #

```
feature pim
ip pim rp-address 10.254.254.1 group-list 227.2.2.0/26 bidir
ip pim rp-address 10.254.254.65 group-list 227.2.2.64/26 bidir
```

Loopback interface PIM configuration (RP) on spine switch S2/RP1

(config) #

```
interface loopback 0
  ip address 10.1.1.53/32
  ip pim sparse-mode
```

IP unnumbered P2P interface configuration on spine switch S2/RP1 to leaf switch V1

(config) #

```
interface Ethernet 1/1
  no switchport
  mtu 9192
  medium p2p
  ip unnumbered loopback 0
  ip pim sparse-mode
```

Loopback interface PIM configuration (for phantom RP) on spine switch S2/RP1

(config) #

```
interface loopback 1
  ip address 10.254.254.2/30
  ip pim sparse-mode
```

(config) #

```
interface loopback 2
  ip address 10.254.254.66/29
  ip pim sparse-mode
```

Using phantom RP on spine switch S3

(config) #

```
feature pim
ip pim rp-address 10.254.254.1 group-list 227.2.2.0/26 bidir
ip pim rp-address 10.254.254.65 group-list 227.2.2.64/26 bidir
```

Loopback interface PIM configuration (RP) on spine switch S3/RP2

(config) #

```
interface loopback 0
  ip address 10.10.50.100/32
  ip pim sparse-mode
```

IP unnumbered P2P interface configuration on spine switch S3/RP2 to leaf switch V1

(config) #

```
interface Ethernet 2/2
  no switchport
  mtu 9192
  medium p2p
  ip unnumbered loopback 0
  ip pim sparse-mode
```

Loopback interface PIM configuration (for phantom RP) on spine switch S3/RP2

(config) #

```
interface loopback 1
  ip address 10.254.254.66/30
  ip pim sparse-mode


interface loopback 2
  ip address 10.254.254.2/29
  ip pim sparse-mode
```

PIM BiDir Verification

Use the following commands for verifying PIM BiDir configuration:

```
Leaf-Switch-V1# show ip mroute

IP Multicast Routing Table for VRF "default"

(*, 227.2.2.0/26), bidir, uptime: 4d08h, pim ip
  Incoming interface: Ethernet1/1, RPF nbr: 10.1.1.53
  Outgoing interface list: (count: 1)
    Ethernet1/1, uptime: 4d08h, pim, (RPF)

(*, 227.2.2.0/32), bidir, uptime: 4d08h, nve ip pim
  Incoming interface: Ethernet1/1, RPF nbr: 10.1.1.53
  Outgoing interface list: (count: 2)
    Ethernet1/1, uptime: 4d08h, pim, (RPF)
    nve1, uptime: 4d08h, nve

(*, 227.2.2.64/26), bidir, uptime: 4d08h, pim ip
```

```
  Incoming interface: Ethernet1/5, RPF nbr: 10.10.50.100/32
  Outgoing interface list: (count: 1)
    Ethernet1/5, uptime: 4d08h, pim, (RPF)

(*, 232.0.0.0/8), uptime: 4d08h, pim ip
  Incoming interface: Null, RPF nbr: 0.0.0.0
  Outgoing interface list: (count: 0)


Leaf-Switch-V1# show ip pim rp

PIM RP Status Information for VRF "default"
BSR disabled
Auto-RP disabled
BSR RP Candidate policy: None
BSR RP policy: None
Auto-RP Announce policy: None
Auto-RP Discovery policy: None

RP: 10.254.254.1, (1),
 uptime: 4d08h   priority: 0,
 RP-source: (local),
 group ranges:
 227.2.2.0/26  (bidir)
RP: 10.254.254.65, (2),
 uptime: 4d08h   priority: 0,
 RP-source: (local),
 group ranges:
 227.2.2.64/26  (bidir)


Leaf-Switch-V1# show ip pim interface

PIM Interface Status for VRF "default"
loopback0, Interface status: protocol-up/link-up/admin-up
  IP address: 10.1.1.54, IP subnet: 10.1.1.54/32
  PIM DR: 10.1.1.54, DR's priority: 1
  PIM neighbor count: 0
  PIM hello interval: 30 secs, next hello sent in: 00:00:23
  PIM neighbor holdtime: 105 secs
  PIM configured DR priority: 1
  PIM configured DR delay: 3 secs
  PIM border interface: no
  PIM GenID sent in Hellos: 0x12650908
  PIM Hello MD5-AH Authentication: disabled
  PIM Neighbor policy: none configured
  PIM Join-Prune inbound policy: none configured
  PIM Join-Prune outbound policy: none configured
  PIM Join-Prune interval: 1 minutes
  PIM Join-Prune next sending: 1 minutes
  PIM BFD enabled: no
  PIM passive interface: no
  PIM VPC SVI: no
  PIM Auto Enabled: no
  PIM Interface Statistics, last reset: never
    General (sent/received):
      Hellos: 13158/0 (early: 0), JPs: 0/0, Asserts: 0/0
      Grafts: 0/0, Graft-Acks: 0/0
      DF-Offers: 0/0, DF-Winners: 0/0, DF-Backoffs: 0/0, DF-Passes: 0/0
    Errors:
      Checksum errors: 0, Invalid packet types/DF subtypes: 0/0
      Authentication failed: 0
      Packet length errors: 0, Bad version packets: 0, Packets from self: 0
      Packets from non-neighbors: 0
          Packets received on passiveinterface: 0
```

```
      JPs received on RPF-interface: 0
      (*,G) Joins received with no/wrong RP: 0/0
      (*,G)/(S,G) JPs received for SSM/Bidir groups: 0/0
      JPs filtered by inbound policy: 0
      JPs filtered by outbound policy: 0

Ethernet1/1, Interface status: protocol-up/link-up/admin-up
  IP unnumbered interface (loopback0)
  PIM DR: 10.1.1.54, DR's priority: 1
  PIM neighbor count: 1
  PIM hello interval: 30 secs, next hello sent in: 00:00:04
  PIM neighbor holdtime: 105 secs
  PIM configured DR priority: 1
  PIM configured DR delay: 3 secs
  PIM border interface: no
  PIM GenID sent in Hellos: 0x2534269b
  PIM Hello MD5-AH Authentication: disabled
  PIM Neighbor policy: none configured
  PIM Join-Prune inbound policy: none configured
  PIM Join-Prune outbound policy: none configured
  PIM Join-Prune interval: 1 minutes
  PIM Join-Prune next sending: 1 minutes
  PIM BFD enabled: no
  PIM passive interface: no
  PIM VPC SVI: no
  PIM Auto Enabled: no
  PIM Interface Statistics, last reset: never
    General (sent/received):
      Hellos: 13152/13162 (early: 0), JPs: 2/0, Asserts: 0/0
      Grafts: 0/0, Graft-Acks: 0/0
      DF-Offers: 9/5, DF-Winners: 6249/6254, DF-Backoffs: 0/1, DF-Passes: 0/1
    Errors:
      Checksum errors: 0, Invalid packet types/DF subtypes: 0/0
      Authentication failed: 0
      Packet length errors: 0, Bad version packets: 0, Packets from self: 0
      Packets from non-neighbors: 0
          Packets received on passiveinterface: 0
      JPs received on RPF-interface: 0
      (*,G) Joins received with no/wrong RP: 0/0
      (*,G)/(S,G) JPs received for SSM/Bidir groups: 0/0
      JPs filtered by inbound policy: 0
      JPs filtered by outbound policy: 0


Leaf-Switch-V1# show ip pim neighbor

PIM Neighbor Status for VRF "default"

Neighbor      Interface      Uptime    Expires      DR         Bidir-     BFD
                                                   Priority    Capable    State
10.1.1.53     Ethernet1/1    1w1d      00:01:33     1             yes      n/a
10.10.50.100 Ethernet2/2    1w1d      00:01:33     1             yes      n/a
```

For a detailed list of commands, refer to the Configuration and Command Reference guides.

Underlay deployment without multicast (Ingress replication)

Ingress replication is supported on Cisco Nexus 9000 Series switches.

Beginning in NX-OS release 9.3(3), Ingress replication is supported on Cisco Nexus 9300-GX switches.

CHAPTER 5

Configuring VXLAN BGP EVPN

This chapter contains the following sections:

About VXLAN BGP EVPN

About RD Auto

The auto-derived Route Distinguisher (rd auto) is based on the Type 1 encoding format as described in IETF RFC 4364 section 4.2 https://tools.ietf.org/html/rfc4364#section-4.2. The Type 1 encoding allows a 4-byte administrative field and a 2-byte numbering field. Within Cisco NX-OS, the auto derived RD is constructed with the IP address of the BGP Router ID as the 4-byte administrative field (RID) and the internal VRF identifier for the 2-byte numbering field (VRF ID).

The 2-byte numbering field is always derived from the VRF, but results in a different numbering scheme depending on its use for the IP-VRF or the MAC-VRF:

- The 2-byte numbering field for the IP-VRF uses the internal VRF ID starting at 1 and increments. VRF IDs 1 and 2 are reserved for the default VRF and the management VRF respectively. The first custom defined IP VRF uses VRF ID 3.
- The 2-byte numbering field for the MAC-VRF uses the VLAN ID + 32767, which results in 32768 for VLAN ID 1 and incrementing.

Example auto-derived Route Distinguisher (RD)

- IP-VRF with BGP Router ID 192.0.2.1 and VRF ID 6 - RD 192.0.2.1:6
- MAC-VRF with BGP Router ID 192.0.2.1 and VLAN 20 - RD 192.0.2.1:32787

About Route-Target Auto

The auto-derived Route-Target (route-target import/export/both auto) is based on the Type 0 encoding format as described in IETF RFC 4364 section 4.2 (https://tools.ietf.org/html/rfc4364#section-4.2). IETF RFC 4364 section 4.2 describes the Route Distinguisher format and IETF RFC 4364 section 4.3.1 refers that it is desirable to use a similar format for the Route-Targets. The Type 0 encoding allows a 2-byte administrative field and a 4-byte numbering field. Within Cisco NX-OS, the auto derived Route-Target is constructed with the Autonomous System Number (ASN) as the 2-byte administrative field and the Service Identifier (VNI) for the 4-byte numbering field.

2-byte ASN

The Type 0 encoding allows a 2-byte administrative field and a 4-byte numbering field. Within Cisco NX-OS, the auto-derived Route-Target is constructed with the Autonomous System Number (ASN) as the 2-byte administrative filed and the Service Identifier (VNI) for the 4-byte numbering field.

Examples of an auto derived Route-Target (RT):

- IP-VRF within ASN 65001 and L3VNI 50001 - Route-Target 65001:50001
- MAC-VRF within ASN 65001 and L2VNI 30001 - Route-Target 65001:30001

For Multi-AS environments, the Route-Targets must either be statically defined or rewritten to match the ASN portion of the Route-Targets.

https://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus9000/sw/7-x/command_references/configuration_commands/b_N9K_Config_Commands_703i7x/b_N9K_Config_Commands_703i7x_chapter_010010.html#wp4498893710

4-byte ASN

The Type 0 encoding allows a 2-byte administrative field and a 4-byte numbering field. Within Cisco NX-OS, the auto-derived Route-Target is constructed with the Autonomous System Number (ASN) as the 2-byte administrative filed and the Service Identifier (VNI) for the 4-byte numbering field. With the ASN demand of 4-byte length and the VNI requiring 24-bit (3-bytes), the Sub-Field length within the Extended Community is exhausted (2-byte Type and 6-byte Sub-Field). As a result of the length and format constraint and the importance of the Service Identifiers (VNI) uniqueness, the 4-byte ASN is represented in a 2-byte ASN named AS_TRANS, as described in IETF RFC 6793 section 9 (https://tools.ietf.org/html/rfc6793#section-9). The 2-byte ASN 23456 is registered by the IANA (https://www.iana.org/assignments/iana-as-numbers-special-registry/iana-as-numbers-special-registry.xhtml) as AS_TRANS, a special purpose AS number that aliases 4-byte ASNs.

Example auto derived Route-Target (RT) with 4-byte ASN (AS_TRANS):

- IP-VRF within ASN 65656 and L3VNI 50001 - Route-Target 23456:50001
- MAC-VRF within ASN 65656 and L2VNI 30001 - Route-Target 23456:30001

Note Beginning with Cisco NX-OS Release 9.2(1), auto derived Route-Target for 4-byte ASN is supported.

Guidelines and Limitations for VXLAN BGP EVPN

VXLAN BGP EVPN has the following guidelines and limitations:

- The following guidelines and limitations apply to VXLAN/VTEP using BGP EVPN:
  - SPAN source or destination is supported on any port.

  For more information, see the Cisco Nexus 9000 Series NX-OS System Management Configuration Guide, Release 9.3(x).
- When SVI is enabled on a VTEP (flood and learn, or EVPN) regardless of ARP suppression, make sure that ARP-ETHER TCAM is carved using the **hardware access-list tcam region arp-ether 256 double-wide** command. This requirement does not apply to Cisco Nexus 9200, 9300-EX, and 9300-FX/FX2/FX3 and 9300-GX platform switches and Cisco Nexus 9500 platform switches with 9700-EX/FX line cards.
- For the Cisco Nexus 9504 and 9508 with R-series line cards, VXLAN EVPN (Layer 2 and Layer 3) is only supported with the 9636C-RX and 96136YC-R line cards.
- VXLAN is not supported on N9K-C92348GC-X switches.
- You can configure EVPN over segment routing or MPLS. See the Cisco Nexus 9000 Series NX-OS Label Switching Configuration Guide, Release 9.3(x) 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 9000 Series NX-OS Label Switching Configuration Guide, Release 9.3(x) for more information.
- In a VXLAN EVPN setup that has 2K VNI scale configuration, the control plane down time may take more than 200 seconds. To avoid potential BGP flap, extend the graceful restart time to 300 seconds.
- Starting from Cisco NX-OS Release 9.3(5), new VXLAN uplink capabilities are introduced:
  - A physical interface in default VRF is supported as VXLAN uplink.
  - A parent interface in default VRF, carrying subinterfaces with VRF and dot1q tags, is supported as VXLAN uplink.
  - A subinterface in any VRF and/or with dot1q tag remains not supported as VXLAN uplink.
  - An SVI in any VRF remains not supported as VXLAN uplink.
  - In vPC with physical peer-link, a SVI can be leveraged as backup underlay, default VRF only between the vPC members (infra-VLAN, system nve infra-vlans).
  - 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.
  - FEX host interfaces remain not supported as VXLAN uplink and cannot have VTEPs connected (BUD node).
- During the vPC Border Gateway boot up process the NVE source loopback interface undergoes the hold down timer twice instead of just once. This is a day-1 and expected behavior.

- You need to configure the VXLAN uplink with **ip unreachables** in order to enable Path maximum transmission unit (MTU) discovery (PMTUD) in a VXLAN set up. PMTUD prevents fragmentation in the path between two endpoints by dynamically determining the lowest MTU along the path from the packet's source to its destination.
- In a VXLAN EVPN setup, border nodes must be configured with unique route distinguishers, preferably using the **auto rd** command. Not using unique route distinguishers across all border nodes is not supported. The use of unique route distinguishers is strongly recommended for all VTEPs of a fabric.
- 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.
- The ARP suppression setting must match across the entire fabric. For a specific VNID, all VTEPs must be either configured or not configured.
- Mobility Sequence number of a locally originated type-2 route (MAC/MAC-IP) can be mismatched between vPC peers, with one vTEP having a sequence number K while other vTEP in the same complex can have the same route with sequence number 0. This does not cause any functional impact and the traffic is not impacted even after the host moves.
- DHCP snooping (Dynamic Host Configuration Protocol snooping) is not supported on VXLAN VLANs.
- RACLs are not supported on VXLAN uplink interfaces. VACLs are not supported on VXLAN de-capsulated traffic in egress direction; this applies for the inner traffic coming from network (VXLAN) towards the access (Ethernet).

  As a best practice, always use PACLs/VACLs for the access (Ethernet) to the network (VXLAN) direction. See the Cisco Nexus 9000 Series NX-OS Security Configuration Guide, Release 9.3(x) for other guidelines and limitations for the VXLAN ACL feature.
- The Cisco Nexus 9000 QoS buffer-boost feature is not applicable for VXLAN traffic.
- For SVI-related triggers (such as shut/unshut or PIM enable/disable), a 30-second delay was added, allowing the Multicast FIB (MFIB) Distribution module (MFDM) to clear the hardware table before toggling between L2 and L3 modes or vice versa.
- For VXLAN BGP EVPN fabrics with EBGP, the following recommendations are applicable:
  - It is recommended to use loopbacks for the EBGP EVPN peering sessions (overlay control-plane).
  - It is a best practice to use the physical interfaces for EBGP IPv4/IPv6 peering sessions (underlay).
- Bind the NVE source-interface to a dedicated loopback interface and do not share this loopback with any function or peerings of Layer-3 protocols. A best practice is to use a dedicated loopback address for the VXLAN VTEP function.
- You must bind NVE to a loopback address that is separate from other loopback addresses that are required by Layer 3 protocols. NVE and other Layer 3 protocols using the same loopback is not supported.
- The NVE source-interface loopback is required to be present in the default VRF.
- Only EBGP peering between a VTEP and external nodes (Edge Router, Core Router or VNF) is supported.
  - EBGP peering from the VTEP to the external node using a physical interface or subinterfaces is recommended and it is a best practice (external connectivity).

- The EBGP peering from the VTEP to the external node can be in the default VRF or in a tenant VRF (external connectivity).
- The EBGP peering from the VTEP to a external node over VXLAN must be in a tenant VRF and must use the update-source of a loopback interface (peering over VXLAN).
- Using an SVI for EBGP peering on a from the VTEP to the External Node requires the VLAN to be local (not VXLAN extended).

- When configuring VXLAN BGP EVPN, only the "System Routing Mode: Default" is applicable for the following hardware platforms:
  - Cisco Nexus 9300 platform switches
  - Cisco Nexus 9300-EX platform switches
  - Cisco Nexus 9300-FX/FX2/FX3 platform switches
  - Cisco Nexus 9300-GX platform switches
  - Cisco Nexus 9500 platform switches with X9500 line cards
  - Cisco Nexus 9500 platform switches with X9700-EX and X9700-FX line cards
- Changing the “System Routing Mode” requires a reload of the switch.
- Cisco Nexus 9516 platform is not supported for VXLAN EVPN.
- VXLAN is supported on Cisco Nexus 9500 platform switches with the following line cards:
  - 9500-R
  - 9564PX
  - 9564TX
  - 9536PQ
  - 9700-EX
  - 9700-FX
- Cisco Nexus 9500 platform switches with 9700-EX or -FX line cards support 1G, 10G, 25G, 40G, 100G and 400G for VXLAN uplinks.
- Cisco Nexus 9200 and 9300-EX/FX/FX2/FX3 and -GX support 1G, 10G, 25G, 40G, 100G and 400G for VXLAN uplinks.
- The Cisco Nexus 9000 platform switches use standards conforming UDP port number 4789 for VXLAN encapsulation. This value is not configurable.
- The Cisco Nexus 9200 platform switches with Application Spine Engine (ASE2) have throughput constrains for packet sizes of 99-122 bytes; packet drops might be experienced.
- The VXLAN network identifier (VNID) 16777215 is reserved and should explicitly not be configured.
- Non-Disruptive In Service Software Upgrade (ND-ISSU) is supported on Nexus 9300 with VXLAN enabled. Exception is ND-ISSU support for Cisco Nexus 9300-FX3 and 9300-GX platform switch.

- Gateway functionality for VXLAN to MPLS (LDP), VXLAN to MPLS-SR (Segment Routing) and VXLAN to SRv6 can be operated on the same Cisco Nexus 9000 Series platform.
    - VXLAN to MPLS (LDP) Gateway is supported on the Cisco Nexus 3600-R and the Cisco Nexus 9500 with R-Series line cards.
    - VXLAN to MPLS-SR Gateway is supported on the Cisco Nexus 9300-FX2/FX3/GX and Cisco Nexus 9500 with R-Series line cards.
    - VXLAN to SRv6 is supported on the Cisco Nexus 9300-GX platform.
    - Multiple Tunnel Encapsulations (VXLAN, GRE and/or MPLS, static label or segment routing) can not co-exist on the same Cisco Nexus 9000 Series switch with Network Forwarding Engine (NFE).
- Resilient hashing is supported on the following switch platform with a VXLAN VTEP configured:
    - Cisco Nexus 9300-EX/FX/FX2/FX3/GX support ECMP resilient hashing.
    - Cisco Nexus 9300 with ALE uplink ports does not support resilient hashing.

Note Resilient hashing is disabled by default.

- It is recommended to use the **vpc orphan-ports suspend** command for single attached and/or routed devices on a Cisco Nexus 9000 platform switch acting as vPC VTEP.
- Routing protocol adjacencies using Anycast Gateway SVIs is not supported.

Note For information about VXLAN BGP EVPN scalability, see the Cisco Nexus 9000 Series NX-OS Verified Scalability Guide.

About VXLAN EVPN with Downstream VNI

Cisco NX-OS Release 9.3(5) introduces VXLAN EVPN with downstream VNI. In earlier releases, the VNI configuration must be consistent across all nodes in the VXLAN EVPN network in order to enable communication between them.

VXLAN EVPN with downstream VNI provides the following solutions:

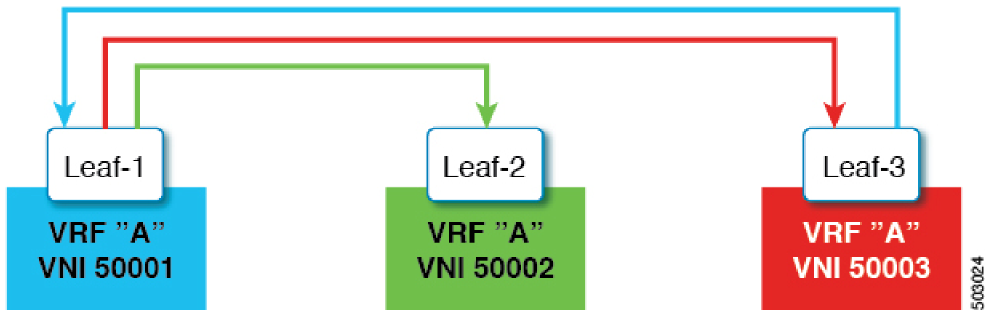
- Enables asymmetric VNI communication across nodes in a VXLAN EVPN network
- Provides customers access to a common shared service outside of their domain (tenant VRF)
- Supports communication between isolated VXLAN EVPN sites that have different sets of VNIs

Asymmetric VNIs

VXLAN EVPN with downstream VNI supports asymmetric VNI allocation.

The following figure shows an example of asymmetric VNIs. All three VTEPs have different VNIs configured for the same IP VRF or MAC VRF.

Figure 11: Asymmetric VNIs



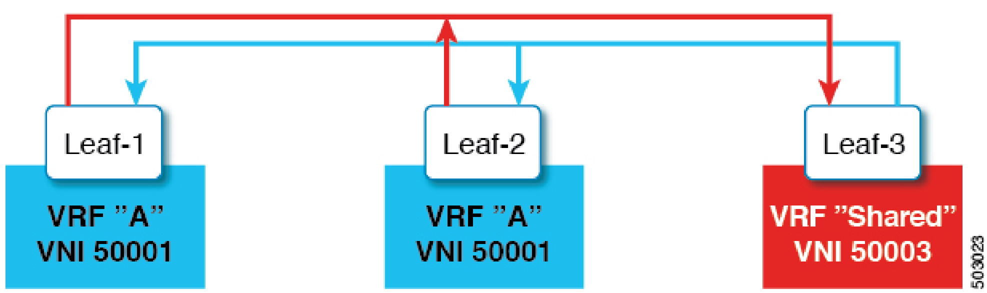
Shared Services VRFs

VXLAN EVPN with downstream VNI supports shared services VRFs. It does so by importing multiple L3VRFs into a single local L3VRF and supporting disparate values of downstream L3VNIs on a per-peer basis.

For example, a DNS server needs to serve multiple hosts in a data center regardless of the tenant VRFs on which the hosts sit. The DNS server is attached to a shared services VRF, which is attached to an L3VNI. To access this server from any of the tenant VRFs, the switches must import the routes from the shared services VRF to the tenant VRF, even though the L3VNI associated to the shared services VRF is different from the L3VNI associated to the tenant VRF.

In the following figure, Tenant VRF A in Leaf-1 can communicate with Tenant VRF A in Leaf-2. However, Tenant VRF A requires access to a shared service sitting behind Leaf-3.

Figure 12: Shared Services VRFs

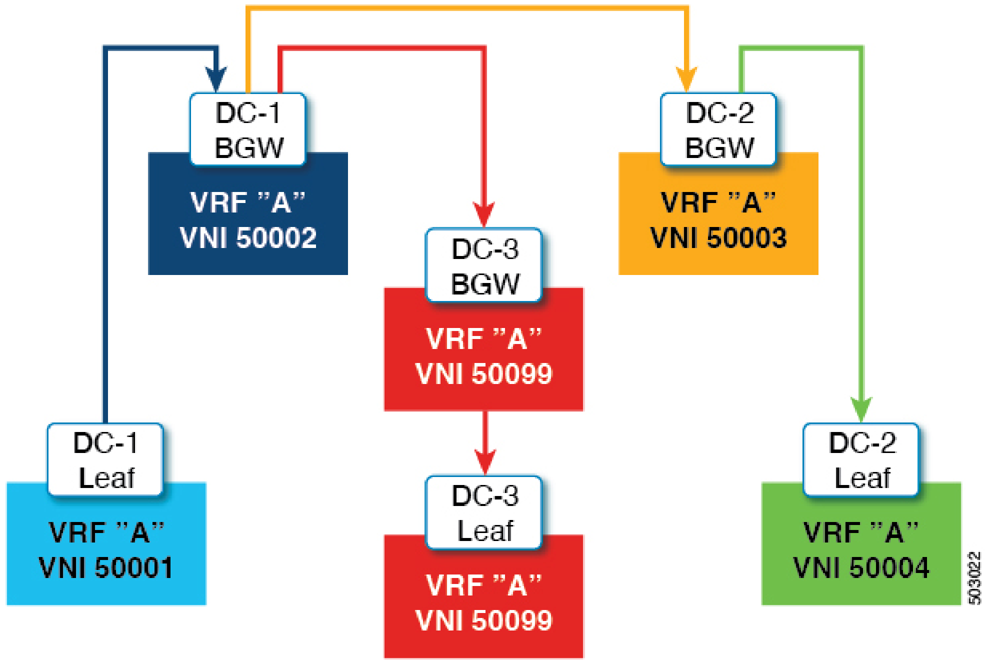


Multi-Site with Asymmetric VNIs

VXLAN EVPN with downstream VNI allows communication between sites that have different sets of VNIs. It does so by stitching the asymmetric VNIs at the border gateways.

In the following figure, DC-1 and DC-2 are asymmetric sites, and DC-3 is a symmetric site. Each site uses different VNIs within its site to communicate.

Figure 13: Multi-Site with Asymmetric VNIs



Guidelines and Limitations for VXLAN EVPN with Downstream VNI

VXLAN EVPN with downstream VNI has the following guidelines and limitations:

- Cisco Nexus 9332C, 9364C, 9300-EX, and 9300-FX/FX2/FXP platform switches and Cisco Nexus 9500 platform switches with -EX/FX line cards support VXLAN EVPN with downstream VNI.
- Beginning with Cisco NX-OS Release 9.3(7), Cisco Nexus 9300-GX platform switches support VXLAN EVPN with downstream VNI.
- VXLAN EVPN with downstream VNI is supported only on the IPv4 underlay.
- Downstream VNI is configured based on route-target export and import. The following conditions must be met to leverage Downstream VNI:
    - Downstream VNI requires the usage of different VRF (MAC-VRF or IP-VRF), each VRF must have a different VNI (Asymmetric VNI).
    - To import routes of a foreign VRF (MAC-VRF or IP-VRF) the appropriate route-target for the import into the local VRF must be configured.
    - The configuration of only auto-derived route-targets will not result in downstream VNI.
    - The export of VRF prefixes can be done by static or auto-derived route-target configuration.
    - The import of a foreign VRF's auto-derived route-target is supported.
    - The import of a foreign VRFs statically configured route-target is supported.

- Downstream VNI is supported for the following underlay constellations:
  - For downstream VNI with Layer-3 VNI, the underlay can be ingress replication or multicast based.
  - For downstream VNI with Layer-2 VNI, the underlay must be in ingress replication. Multicast based underlay is not supported with downstream VNI of Layer-2 VNIs.
- Downstream VNI requires to have consistent configuration:
  - All multi-site Border Gateway (BGW) in a site must have a consistent configuration.
  - All vPC members in a vPC domain must have consistent configuration.
- The usage of downstream VNI with multi-site requires all BGW across all sites to run at least Cisco NX-OS Release 9.3(5).
- For existing centralized VRF route leaking deployments, a brief traffic loss might occur during ISSU to Cisco NX-OS Release 9.3(5) or later.
- For successful downgrade from Cisco NX-OS Release 9.3(5) to a prior release, ensure that the asymmetric VNI configuration has been removed. Downstream VNI is not supported before Cisco NX-OS Release 9.3(5) and hence traffic forwarding would be impacted.
- Layer-3 VNIs (IP-VRF) can flexibly mapped between VNIs per peer.
  - VNI 50001 on VTEP1 can perform symmetric VNI with VNI 50001 and asymmetric VNI with VNI 50002 on VTEP2 at the same time.
  - VNI 50001 on VTEP1 can perform asymmetric VNI with VNI 50002 on VTEP2 and VNI 50003 on VTEP3.
  - VNI 50001 on VTEP1 can perform asymmetric VNI with VNI 50002 and VNI5003 on VTEP2 at the same time.
- Layer-2 VNIs (MAC-VRF) can only be mapped to one VNI per peer.
  - VNI 30001 on VTEP1 can perform asymmetric VNI with VNI 30002 on VTEP2 and VNI 30003 on VTEP3.
  - VNI 30001 on VTEP1 cannot perform asymmetric VNI with VNI 30002 and VNI 3003 on VTEP2 at the same time.
- iBGP sessions between vPC peer nodes in a VRF are not supported.
- BGP peering across VXLAN and Downstream VNI support the following constellations:
  - BGP peering between symmetric VNI is supported by using loopbacks.
  - BGP peering between asymmetric VNI is supported if the VNIs are in a direct message relationship. A loopback from VNI 50001 (on VTEP1) can peer with a loopback in VNI 50002 (on VTEP2).
  - BGP peering between asymmetric VNI is supported if the VNIs are in a direct message relationship but on different VTEPs. A loopback from VNI 50001 (on VTEP1) can peer with a loopback in VNI 50002 (on VTEP2 and VTEP3).
  - BGP peering between asymmetric VNI is not supported if the VNIs are in a 1:N relationship. A loopback in VNI 50001 (VTEP1) can’t peer with a loopback in VNI 50002 (VTEP2) and VNI 50003 (VTEP3) at the same time.

- VXLAN consistency checker is not supported for VXLAN EVPN with downstream VNI.
- VXLAN EVPN with downstream VNI is currently not supported with the following feature combinations:
  - VXLAN static tunnels
  - TRM and TRM with Multi-Site
  - CloudSec VXLAN EVPN Tunnel Encryption
  - ESI-based multihoming
  - Seamless integration of EVPN with L3VPN (MPLS SR)
  - VXLAN policy-based routing (PBR)
- Make sure that you configure L2VNI SVI on Anycast BGW to enable DSVNI MAC-IP Layer 3 label translation in a multisite environment. The functionality of DSVNI is limited for reoriginated routes, which requires as association between L2VNI and VRF. You can associate using the VRF member command in L2VNI SVI.

Configuring VXLAN BGP EVPN

Enabling VXLAN

Enable VXLAN and the EVPN.

SUMMARY STEPS

1. **feature vn-segment**
2. **feature nv overlay**
3. **feature vn-segment-vlan-based**
4. **feature interface-vlan**
5. **nv overlay evpn**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**feature vn-segment**	Enable VLAN-based VXLAN
Step 2	**feature nv overlay**	Enable VXLAN
Step 3	**feature vn-segment-vlan-based**	Enable VN-Segment for VLANs.
Step 4	**feature interface-vlan**	Enable Switch Virtual Interface (SVI).
Step 5	**nv overlay evpn**	Enable the EVPN control plane for VXLAN.

Configuring VLAN and VXLAN VNI

Note Step 3 to Step 6 are optional for configuring the VLAN for VXLAN VNI and are only necessary in case of a custom route distinguisher or route-target requirement (not using auto derivation).

SUMMARY STEPS

1. **vlan** *number*
2. **vn-segment** *number*
3. **evpn**
4. **vni** *number* **l2**
5. **rd auto**
6. **route-target both** {**auto** | *rt*}

DETAILED STEPS

	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.
Step 3	**evpn**	Enter EVI (EVPN Virtual Instance) configuration mode.
Step 4	**vni** *number* **l2**	Specify the Service Instance (VNI) for the EVI.
Step 5	**rd auto**	Specify the MAC-VRF's route distinguisher (RD).
Step 6	**route-target both** {**auto** | *rt*}	Configure the route target (RT) for import and export of MAC prefixes. The RT is used for a per-MAC-VRF prefix import/export policy. If you enter an RT, the following formats are supported: ASN2:NN, ASN4:NN, or IPV4:NN. **Note** Specifying the auto option is applicable only for IBGP. Manually configured route targets are required for EBGP and for asymmetric VNIs.

Configuring VRF for VXLAN Routing

Configure the tenant VRF.

Note Step 3 to step 6 are optional for configuring the VRF for VXLAN Routing and are only necessary in case of a custom route distinguisher or route-target requirement (not using auto derivation).

SUMMARY STEPS

1. **vrf context** *vrf-name*
2. **vni** *number*
3. **rd auto**
4. **address-family** {**ipv4** | **ipv6**} **unicast**
5. **route-target both** {**auto** | *rt*}
6. **route-target both** {**auto** | *rt*} **evpn**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**vrf context** *vrf-name*	Configure the VRF.
Step 2	**vni** *number*	Specify the VNI.
Step 3	**rd auto**	Specify the IP-VRF's route distinguisher (RD).
Step 4	**address-family** {**ipv4** | **ipv6**} **unicast**	Configure the IPv4 or IPv6 unicast address family.
Step 5	**route-target both** {**auto** | *rt*}	Configure the route target (RT) for import and export of IPv4 or IPv6 prefixes. The RT is used for a per-IP-VRF prefix import/export policy. If you enter an RT, the following formats are supported: ASN2:NN, ASN4:NN, or IPV4:NN. **Note** Specifying the **auto** option is applicable only for IBGP. Manually configured route targets are required for EBGP and for asymmetric VNIs.
Step 6	**route-target both** {**auto** | *rt*} **evpn**	Configure the route target (RT) for import and export of IPv4 or IPv6 prefixes. The RT is used for a per-VRF prefix import/export policy. If you enter an RT, the following formats are supported: ASN2:NN, ASN4:NN, or IPV4:NN. **Note** Specifying the **auto** option is applicable only for IBGP. Manually configured route targets are required for EBGP and for asymmetric VNIs.

Configuring SVI for Core-facing VXLAN Routing

Configure the core-facing SVI VRF.

SUMMARY STEPS

1. **vlan** *number*
2. **vn-segment** *number*

3. **interface** *vlan-number*
4. **mtu** *vlan-number*
5. **vrf member** *vrf-name*
6. **no** {**ip** |**ipv6**} **redirects**
7. **ip forward**
8. **ipv6 address use-link-local-only**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**vlan** *number*	Specify VLAN.
Step 2	**vn-segment** *number*	Map VLAN to VXLAN VNI to configure Layer 3 VNI under VXLAN VLAN.
Step 3	**interface** *vlan-number*	Specify VLAN interface.
Step 4	**mtu** *vlan-number*	MTU size in bytes <68-9216>.
Step 5	**vrf member** *vrf-name*	Assign to VRF.
Step 6	**no** {**ip** |**ipv6**} **redirects**	Disable sending IP redirect messages for IPv4 and IPv6.
Step 7	**ip forward**	Enable IPv4 based lookup even when the interface VLAN has no IP address defined.
Step 8	**ipv6 address use-link-local-only**	Enable IPv6 forwarding. **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.

Configuring SVI for Host-Facing VXLAN Routing

Configure the SVI for hosts, acting as Distributed Default Gateway.

SUMMARY STEPS

1. **fabric forwarding anycast-gateway-mac** *address*
2. **vlan** *number*
3. **vn-segment** *number*
4. **interface** *vlan-number*
5. **vrf member** *vrf-name*
6. **ip address** *address*
7. **fabric forwarding mode anycast-gateway**

DETAILED STEPS

	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	**vlan** *number*	Specify VLAN.
Step 3	**vn-segment** *number*	Specify vn-segment.
Step 4	**interface** *vlan-number*	Specify VLAN interface.
Step 5	**vrf member** *vrf-name*	Assign to VRF.
Step 6	**ip address** *address*	Specify IP address.
Step 7	**fabric forwarding mode anycast-gateway**	Associate SVI with anycast gateway under VLAN configuration mode.

Configuring the NVE Interface and VNIs Using Multicast

SUMMARY STEPS

1. **interface** *nve-interface*
2. **source-interface loopback1**
3. **host-reachability protocol bgp**
4. **global mcast-group** *ip-address* **{L2 | L3}**
5. **member vni** *vni*
6. **mcast-group** *ip address*
7. **member vni** *vni* **associate-vrf**
8. **mcast-group** *address*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**interface** *nve-interface*	Configure the NVE interface.
Step 2	**source-interface loopback1**	Binds the NVE source-interface to a dedicated loopback interface.
Step 3	**host-reachability protocol bgp**	This defines BGP as the mechanism for host reachability advertisement
Step 4	**global mcast-group** *ip-address* **{L2 | L3}**	Configures the mcast group globally (for all VNI) on a per-NVE interface basis. This applies and gets inherited s to all Layer 2 or Layer 3 VNIs.

	Command or Action	Purpose
		Note Layer3 macst group is only used for Tenant Routed Multicast (TRM).
Step 5	**member vni** *vni*	Add Layer 2 VNIs to the tunnel interface.
Step 6	**mcast-group** *ip address*	Configure the mcast group on a per-VNI basis. Add Layer 2 VNI specific mcast group and override the global set configuration. **Note** Instead of a mcast group, ingress replication can be configured.
Step 7	**member vni** *vni* **associate-vrf**	Add Layer-3 VNIs, one per tenant VRF, to the overlay. **Note** Required for VXLAN routing only.
Step 8	**mcast-group** *address*	Configure the mcast group on a per-VNI basis. Add Layer 3 VNI specific mcast group and override the global set configuration.

Configuring VXLAN EVPN Ingress Replication

For VXLAN EVPN ingress replication, the VXLAN VTEP uses a list of IP addresses of other VTEPs in the network to send BUM (broadcast, unknown unicast and multicast) traffic. These IP addresses are exchanged between VTEPs through the BGP EVPN control plane.

Note VXLAN EVPN ingress replication is supported on:

- Cisco Nexus Series 9300 Series switches (7.0(3)I1(2) and later).
- Cisco Nexus Series 9500 Series switches (7.0(3)I2(1) and later).

Before you begin: The following are required before configuring VXLAN EVPN ingress replication (7.0(3)I1(2) and later):

- Enable VXLAN.
- Configure VLAN and VXLAN VNI.
- Configure BGP on the VTEP.
- Configure RD and Route Targets for VXLAN Bridging.

SUMMARY STEPS

1. **interface** *nve-interface*
2. **host-reachability protocol bgp**
3. **global ingress-replication protocol bgp**

4. **member vni** *vni* **associate-vrf**
5. **member vni** *vni*
6. **ingress-replication protocol bgp**

DETAILED STEPS

	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	**global ingress-replication protocol bgp**	Enables globally (for all VNI) the VTEP to exchange local and remote VTEP IP addresses on the VNI in order to create the ingress replication list. This enables sending and receiving BUM traffic for the VNI. **Note** Using ingress-replication protocol bgp avoids the need for any multicast configurations that might have been required for configuring the underlay.
Step 4	**member vni** *vni* **associate-vrf**	Add Layer-3 VNIs, one per tenant VRF, to the overlay. **Note** Required for VXLAN routing only.
Step 5	**member vni** *vni*	Add Layer 2 VNIs to the tunnel interface.
Step 6	**ingress-replication protocol bgp**	Enables the VTEP to exchange local and remote VTEP IP addresses on a oer VNI basis in order to create the ingress replication list. This enables sending and receiving BUM traffic for the VNI and override the global configuration. **Note** Instead of a ingress replication, mcast group can be configured. **Note** Using **ingress-replication protocol bgp** avoids the need for any multicast configurations that might have been required for configuring the underlay.

Configuring BGP on the VTEP

SUMMARY STEPS

1. **router bgp** *number*
2. **router-id** *address*
3. **neighbor** *address* **remote-as** *number*
4. **address-family l2vpn evpn**
5. (Optional) **Allowas-in**
6. **send-community extended**

7. **vrf** *vrf-name*
8. **address-family ipv4 unicast**
9. **advertise** *l2vpn* **evpn**
10. **maximum-paths path** {**ibgp**}
11. **address-family ipv6 unicast**
12. **advertise** *l2vpn* **evpn**
13. **maximum-paths path** {**ibgp**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**router bgp** *number*	Configure BGP.
Step 2	**router-id** *address*	Specify router address.
Step 3	**neighbor** *address* **remote-as** *number*	Define MPBGP neighbors. Under each neighbor define L2VPN EVPN.
Step 4	**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 5	(Optional) **Allowas-in**	Only for EBGP deployment cases: Allows duplicate autonomous system (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 6	**send-community extended**	Configures community for BGP neighbors.
Step 7	**vrf** *vrf-name*	Specify VRF.
Step 8	**address-family ipv4 unicast**	Configure the address family for IPv4.
Step 9	**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 10	**maximum-paths path** {**ibgp**}	Enable ECMP for EVPN transported IP Prefixes within the IPv6 address-family of the respective VRF.
Step 11	**address-family ipv6 unicast**	Configure the address family for IPv6.
Step 12	**advertise** *l2vpn* **evpn**	Enable advertising EVPN routes.

	Command or Action	Purpose
		Note 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 13	**maximum-paths path** {**ibgp**}	Enable ECMP for EVPN transported IP Prefixes within the IPv6 address-family of the respective VRF.

Configuring iBGP for EVPN on the Spine

SUMMARY STEPS

1. **router bgp** *autonomous system number*
2. **neighbor** *address* **remote-as** *number*
3. **address-family l2vpn evpn**
4. **send-community extended**
5. **route-reflector-client**
6. **retain route-target all**
7. **address-family l2vpn evpn**
8. **disable-peer-as-check**
9. **route-map permitall out**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**router bgp** *autonomous system number*	Specify BGP.
Step 2	**neighbor** *address* **remote-as** *number*	Define neighbor.
Step 3	**address-family l2vpn evpn**	Configure address family Layer 2 VPN EVPN under the BGP neighbor.
Step 4	**send-community extended**	Configures community for BGP neighbors.
Step 5	**route-reflector-client**	Enable Spine as Route Reflector.
Step 6	**retain route-target all**	Configure retain route-target all under address-family Layer 2 VPN EVPN [global]. **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 7	**address-family l2vpn evpn**	Configure address family Layer 2 VPN EVPN under the BGP neighbor.

	Command or Action	Purpose
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	**route-map permitall out**	Applies route-map to keep the next-hop unchanged. **Note** Required for eBGP.

Configuring eBGP for EVPN on the Spine

SUMMARY STEPS

1. **route-map NEXT-HOP-UNCH permit 10**
2. **set ip next-hop unchanged**
3. **router bgp** *autonomous system number*
4. **address-family l2vpn evpn**
5. **retain route-target all**
6. **neighbor** *address* **remote-as** *number*
7. **address-family l2vpn evpn**
8. **disable-peer-as-check**
9. **send-community extended**
10. **route-map NEXT-HOP-UNCH out**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**route-map NEXT-HOP-UNCH permit 10**	Configure route-map to keepthe next-hop unchanged for EVPN routes.
Step 2	**set ip next-hop unchanged**	Set next-hop address. **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.

	Command or Action	Purpose
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]. **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.
Step 9	**send-community extended**	Configures community for BGP neighbors.
Step 10	**route-map NEXT-HOP-UNCH out**	Applies route-map to keep the next-hop unchanged.

Suppressing ARP

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

Note For information on configuring ACL TCAM regions, see the *Configuring IP ACLs* chapter of the Cisco Nexus 9000 Series NX-OS Security Configuration Guide.

SUMMARY STEPS

1. **hardware access-list tcam region arp-ether** *size* **double-wide**
2. **interface nve 1**
3. **global suppress-arp**
4. **member vni** *vni-id*
5. **suppress-arp**
6. **suppress-arp disable**

DETAILED STEPS

	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. **Note** Configuring the **hardware access-list tcam region arp-ether** *size* **double-wide** command is not required for Cisco Nexus 9200, 9300-EX, and 9300-FX/FX2/FX3 and 9300-GX platform switches.
Step 2	**interface nve 1**	Create the network virtualization endpoint (NVE) interface.
Step 3	**global suppress-arp**	Configure to suppress ARP globally for all Layer 2 VNI.within the NVE interface.
Step 4	**member vni** *vni-id*	Specify VNI ID.
Step 5	**suppress-arp**	Configure to suppress ARP under Layer 2 VNI and overrides the global set default.
Step 6	**suppress-arp disable**	Disables the global setting of the ARP suppression on a specific VNI.

Disabling VXLANs

SUMMARY STEPS

1. **configure terminal**
2. **no nv overlay evpn**
3. **no feature vn-segment-vlan-based**
4. **no feature nv overlay**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	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

	Command or Action	Purpose
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

For IP addresses:

Cisco NX-OS supports duplicate detection for IP addresses. This enables the detection of duplicate IP addresses based on the number of moves in a given time-interval (seconds), if host appears simultaneously under two VTEP's.

Simultaneous availability of host under two VTEP's is detected by host mobility logic with 600 msec refresh timeout for IPv4 hosts and default refresh time out logic for IPv6 addresses (default is 3 seconds).

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

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**)

Wherever a host IP address is permanently frozen, a syslog message is written by HMM.

```
2021 Aug 26 01:08:26 leaf hmm: (vrf-name) [IPv4] Freezing potential duplicate host
20.2.0.30/32, reached recover count (5) threshold
```

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.

Command	Description
`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.

For MAC addresses:

Cisco NX-OS supports duplicate detection for MAC addresses. This enables the detection of duplicate 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.)

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
```

MAC address remains in permanently frozen list until both local and remote entry exists.

Unconfiguring below commands will not disable permanently frozen functionality rather will change the parameters to default values.

- **l2rib dup-host-mac-detection**
- **l2rib dup-host-recovery**

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).

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

Configuring Event History Size for L2RIB

To set the event history size for the L2RIB component follow these steps:

SUMMARY STEPS

1. **configure terminal**
2. **l2rib event-history** { **mac** | **mac-ip** | **loop-detection** } **size** { **default** | **medium** | **high** | **very-high** }
3. **clear l2rib event-history** { **mac** | **mac-ip** | **loop-detection** } **size** { **default** | **medium** | **high** | **very-high** }

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enter global configuration mode.
Step 2	**l2rib event-history** { **mac** | **mac-ip** | **loop-detection** } **size** { **default** | **medium** | **high** | **very-high** } **Example:** `switch(config)# l2rib event-history mac size low`	Sets the event history size for the L2RIB component.
Step 3	**clear l2rib event-history** { **mac** | **mac-ip** | **loop-detection** } **size** { **default** | **medium** | **high** | **very-high** } **Example:** `switch(config)# clear l2rib event-history mac size low`	Clears the set event history size for the L2RIB component.

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.

Command	Purpose
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.

Verifying the VXLAN EVPN with Downstream VNI Configuration

To display the VXLAN EVPN with downstream VNI configuration information, enter one of the following commands:

Command	Purpose
show bgp evi *l2-evi*	Displays the VRF associated with an L2VNI.
show forwarding adjacency nve platform	Displays both symmetric and asymmetric NVE adjacencies with the corresponding DestInfoIndex.
show forwarding route vrf *vrf*	Displays the egress VNI or downstream VNI for each next-hop.
show ip route detail vrf *vrf*	Displays the egress VNI or downstream VNI for each next-hop.

Command	Purpose
show l2route evpn mac-ip all detail	Displays labeled next-hops that are present in the remote MAC routes.
show l2route evpn imet all detail	Displays the egress VNI associated with the remote peer.
show nve peers control-plane-vni peer-ip *ip-address*	Displays the egress VNI or downstream VNI for each NVE adjacency.

The following example shows sample output for the **show bgp evi** *l2-evi* command:

```
switch# show bgp evi 100
-----------------------------------------------
  L2VNI ID                      : 100 (L2-100)
  RD                            : 3.3.3.3:32867
  Secondary RD                  : 1:100
  Prefixes (local/total)        : 1/6
  Created                       : Jun 23 22:35:13.368170
  Last Oper Up/Down             : Jun 23 22:35:13.369005 / never
  Enabled                       : Yes
  Associated IP-VRF             : vni100
  Active Export RT list         :
        100:100
  Active Import RT list         :
        100:100
```

The following example shows sample output for the **show forwarding adjacency nve platform** command:

```
switch# show forwarding adjacency nve platform
slot  1
=======
IPv4 NVE adjacency information

next_hop:12.12.12.12   interface:nve1 (0x49000001)  table_id:1
  Peer_id:0x49080002  dst_addr:12.12.12.12  src_addr:13.13.13.13  RefCt:1  PBRCt:0
Flags:0x440800
cp : TRUE, DCI peer: FALSE is_anycast_ip FALSE dsvni peer: FALSE
  HH:0x7a13f  DstInfoIndex:0x3002
    tunnel init: unit-0:0x3 unit-1:0x0

next_hop:12.12.12.12   interface:nve1 (0x49000001)  table_id:1
  Peer_id:0x49080002  dst_addr:12.12.12.12  src_addr:13.13.13.13  RefCt:1  PBRCt:0
Flags:0x10440800
cp : TRUE, DCI peer: FALSE is_anycast_ip FALSE dsvni peer: TRUE
  HH:0x7a142  DstInfoIndex:0x3ffd
    tunnel init: unit-0:0x6 unit-1:0x0
...
```

The following example shows sample output for the **show forwarding route vrf** *vrf* command:

```
switch# show forwarding route vrf vrf1000

slot  1
=======

IPv4 routes for table vrf1000/base

--------------+-------------------+--------------+-----------------+-----------------
Prefix        | Next-hop          | Interface    | Labels          | Partial Install
```

```
--------------+-------------------+--------------+-----------------+-----------------
…..
10.1.1.11/32    12.12.12.12         nve1        dsvni: 301000
10.1.1.20/32    123.123.123.123     nve1        dsvni: 301000
10.1.1.21/32    30.30.30.30         nve1        dsvni: 301000
10.1.1.30/32    10.1.1.30           Vlan10
```

The following example shows sample output for the **show ip route detail vrf** *vrf* command:

```
switch# show ip route detail vrf default
IP Route Table for VRF "default"
  '*' denotes best ucast next-hop
  '**' denotes best mcast next-hop
  '[x/y]' denotes [preference/metric]
  '%<string>' in via output denotes VRF <string>

193.0.1.0/24, ubest/mbest: 4/0
      *via 30.1.0.2, Eth1/1, [100/0], 00:00:05, urib_dt6-client1 segid: 6544, tunnelid:
0x7b9 encap: VXLAN

      *via 30.1.1.2, Eth1/1, [100/0], 00:00:05, urib_dt6-client1 segid: 6545, (Asymmetric)
 tunnelid: 0x7ba encap: VXLAN

      *via 30.1.2.2, Eth1/1, [100/0], 00:00:05, urib_dt6-client1 segid: 6546, (Asymmetric)
 tunnelid: 0x7bb encap: VXLAN
```

The following example shows sample output for the **show l2route evpn mac-ip all detail** command:

```
switch# 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 (Orp):Orphan
Topology Mac Address    Host IP   Prod   Flags Seq No  Next-Hops
-------- -------------- --------  ------ ----- ------- ----------------------------
5        0000.0005.1301 1.3.13.1  BGP    --     0      102.1.13.1 (Label: 2000005)
5        0000.0005.1401 1.3.14.1  BGP    --     0      102.1.145.1 (Label: 2000005)
```

The following example shows sample output for the **show l2route evpn imet all detail** command:

```
switch# show l2route evpn imet all

Flags- (F): Originated From Fabric, (W): Originated from WAN

Topology ID VNI         Prod  IP Addr     Flags
----------- ----------- ----- ----------- ---------
3           2000003     BGP   102.1.13.1    -
3           2000003     BGP   102.1.31.1    -
3           2000003     BGP   102.1.32.1    -
3           2000003     BGP   102.1.145.1   -
```

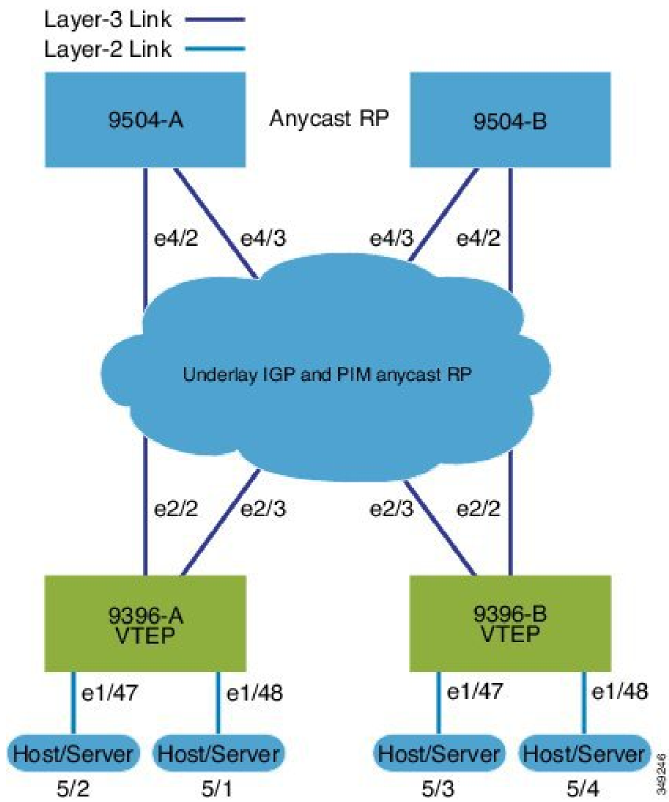
The following example shows sample output for the **show nve peers control-plane-vni** command. In this example, 3000003 is the downstream VNI.

```
switch# show nve peers control-plane-vni peer-ip 203.1.1.1
Peer       VNI     Learn-Source Gateway-MAC     Peer-type  Egress-VNI SW-BD  State

---------  -----   ------------ --------------- ---------- ---------- -----
----------------------
203.1.1.1  2000003 BGP          f40f.1b6f.f8db   FAB        3000003      3005
peer-vni-add-complete
```

Example of VXLAN BGP EVPN (IBGP)

An example of a VXLAN BGP EVPN (IBGP):

Figure 14: 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 Router ID, PIM, 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 224.0.0.0/4
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

```
nv overlay evpn
```

  - Enable the relevnt Protocols

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

  - Configure Loopback for local Router ID, PIM, 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 AnycastRP

```
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 224.0.0.0/4
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 underlayrouting

```
router ospf 1
```

- Configure interfaces for Spine-leaf interconnect

```
interface Ethernet4/2
  ip address 192.168.3.42/24
  ip router ospf 1 area 0.0.0.0
  ip pim sparse-mode
  no shutdown

interface Ethernet4/3
  ip address 192.168.4.43/24
  ip router ospf 1 area 0.0.0.0
  ip pim sparse-mode
  no shutdown
```

- 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 Router ID, PIM, 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 Loopback for local VTEP IP

```
interface loopback1
  ip address 33.1.1.1/32
  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
  shutdown
```

- Configure route-map to Redistribute Host-SVI (Silent Host)

```
route-map HOST-SVI permit 10
  match tag 54321
```

- Configure PIM RP

```
ip pim rp-address 100.1.1.1 group-list 224.0.0.0/4
```

- Create VLANs

```
vlan 1001-1002
```

- Create overlay VRF VLAN and configure vn-segment

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

- Create overlay VRF VLAN and configure vn-segment

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

- Configure Core-facing SVI for VXLAN routing

```
interface vlan101
 no shutdown
  vrf member vxlan-900001
  ip forward
  no ip redirects
  ipv6 address use-link-local-only
  no ipv6 redirects
```

- 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
  rd auto
```

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

```
\
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 tag 54321
  ipv6 address 4:1:0:1::1/64 tag 54321
  fabric forwarding mode anycast-gateway

interface vlan1002
  no shutdown
  vrf member vxlan-900001
```

```
  ip address 4.2.2.1/24 tag 54321
  ipv6 address 4:2:0:1::1/64 tag 54321
  fabric forwarding mode anycast-gateway
```

- Configure ACL TCAM region for ARP suppression

Note The **hardware access-list tcam region arp-ether 256 double-wide** command is not needed for Cisco Nexus 9300-EX and 9300-FX/FX2/FX3 and 9300-GX platform switches.

```
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 Option 1 for a small number of VNIs. Use Option 2 to leverage the simplified configuration mode.

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
    mcast-group 239.0.0.1
  member vni 2001002
    mcast-group 239.0.0.1
```

Option 2

```
interface nve1
  source-interface loopback1
  host-reachability protocol bgp
  global mcast-group 239.0.0.1 L2
  member vni 2001001
  member vni 2001002
  member vni 2001007-2001010
```

- Configure interfaces for hosts/servers

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

interface Ethernet1/48
  switchport
```

```
  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
      redistribute direct route-map HOST-SVI
    address-family ipv6 unicast
      redistribute direct route-map HOST-SVI
```

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 commands in EVPN mode do not need to be entered.

```
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

    ```
    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 underlayrouting

    ```
    router ospf 1
    ```

  - Configure Loopback for local Router ID, PIM, and BGP

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

  - Configure Loopback for local VTEP IP

    ```
    interface loopback1
      ip address 44.1.1.1/32
      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 192.168.3.22/24
      ip router ospf 1 area 0.0.0.0
      ip pim sparse-mode
      no shutdown

    interface Ethernet2/3
      no switchport
      ip address 192.168.4.23/24
      ip router ospf 1 area 0.0.0.0
      ip pim sparse-mode
      shutdown
    ```

  - Configure route-map to Redistribute Host-SVI (Silent Host)

    ```
    route-map HOST-SVI permit 10
      match tag 54321
    ```

  - Configure PIM RP

```
ip pim rp-address 100.1.1.1 group-list 224.0.0.0/4
```

- Create VLANs

```
vlan 1001-1002
```

- Create overlay VRF VLAN and configure vn-segment

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

- Configure Core-facing SVI for VXLAN routing

```
interface vlan101
 no shutdown
  vrf member vxlan-900001
  ip forward
  no ip redirects
  ipv6 address use-link-local-only
  no ipv6 redirects
```

- 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
  rd auto
```

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

```
  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

Note The **hardware access-list tcam region arp-ether 256 double-wide** command is not needed for Cisco Nexus 9300-EX and 9300-FX/FX2/FX3 and 9300-GX platform switches.

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

-

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 leverage the simplified configuration mode.

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
    mcast-group 239.0.0.1
  member vni 2001002
    mcast-group 239.0.0.1
```

Option 2

```
interface nve1
  interface nve1
  source-interface loopback1
  host-reachability protocol bgp
  global mcast-group 239.0.0.1 L2
  member vni 2001001
  member vni 2001002
  member vni 2001007-2001010
```

- Configure interfaces for hosts/servers

```
interface Ethernet1/47
  switchport
```

```
  switchport access vlan 1002

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

- 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
  vrf vxlan-900001
    address-family ipv4 unicast
      redistribute direct route-map HOST-SVI
    address-family ipv6 unicast
      redistribute direct route-map HOST-SVI
```

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** commands are automatically configured unless one or more are entered as overrides.

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

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

```
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
```

- Configure interface vlan on Border Gateway (BGW)

```
interface vlan101
  no shutdown
  vrf member evpn-tenant-3103101
  no ip redirects
  ip address 101.1.0.1/16
  ipv6 address cafe:101:1::1/48
  no ipv6 redirects
  fabric forwarding mode anycast-gateway
```

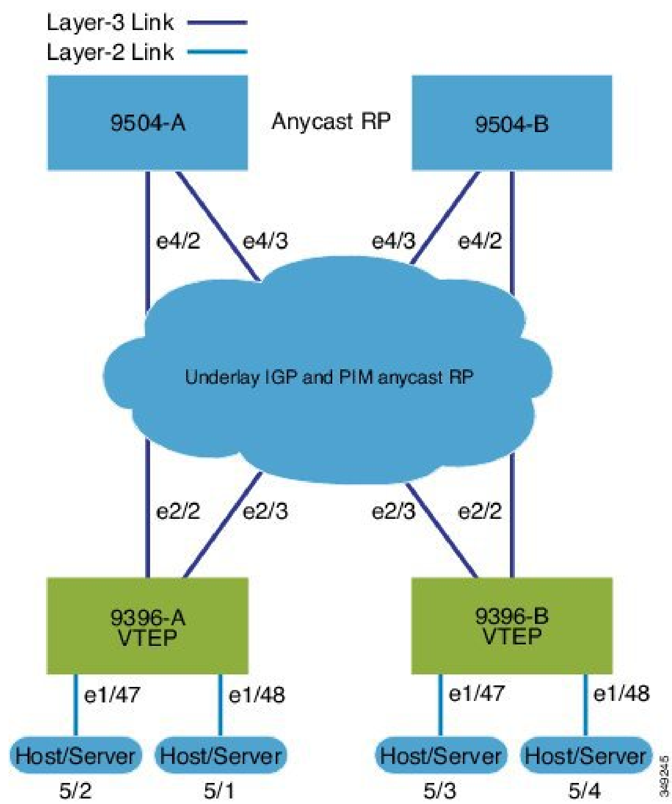
Note When you have IBGP session between BGWs and EBGP fabric is used, you need to configure the route-map to make VIP or VIP_R route advertisement with higher AS-PATH when local VIP or VIP_R is down (due to reload or fabric link flap). A sample route-map configuration is provided below. In this example 192.0.2.1 is VIP address and 198.51.100.1 is BGP VIP route's nexthop learned from same BGW site.

```
ip prefix-list vip_ip seq 5 permit 192.0.2.1/32
ip prefix-list vip_route_nh seq 5 permit 198.51.100.1/32

route-map vip_ip permit 5
  match ip address prefix-list vip_ip
  match ip next-hop prefix-list vip_route_nh
  set as-path prepend 5001 5001 5001
route-map vip_ip permit 10
```

Example of VXLAN BGP EVPN (EBGP)

An example of a VXLAN BGP EVPN (EBGP):

Figure 15: 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 Router ID, PIM, and BGP

```
interface loopback0
  ip address 10.1.1.1/32 tag 12345
  ip pim sparse-mode
```

  - Configure Loopback for Anycast RP

```
interface loopback1
  ip address 100.1.1.1/32 tag 12345
  ip pim sparse-mode
```

  - Configure Anycast RP

```
ip pim rp-address 100.1.1.1 group-list 224.0.0.0/4
```

```
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 NEXT-HOP-UNCH permit 10
  set ip next-hop unchanged
```

- Configure route-map to Redistribute Loopback

```
route-map LOOPBACK permit 10
  match tag 12345
```

- 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 NEXT-HOP-UNCH
    retain route-target all
  neighbor 30.1.1.1 remote-as 200
    update-source loopback0
    ebgp-multihop 3
    address-family l2vpn evpn
      send-community both
      disable-peer-as-check
      route-map NEXT-HOP-UNCH out
  neighbor 40.1.1.1 remote-as 200
    update-source loopback0
    ebgp-multihop 3
    address-family l2vpn evpn
      send-community both
      disable-peer-as-check
      route-map NEXT-HOP-UNCH out
```

- Configure BGP underlay for the IPv4 unicast address family.

```
 address-family ipv4 unicast
    redistribute direct route-map LOOPBACK
  neighbor 192.168.1.22 remote-as 200
    update-source ethernet4/2
    address-family ipv4 unicast
      allowas-in
      disable-peer-as-check
  neighbor 192.168.2.23 remote-as 200
    update-source ethernet4/3
    address-family ipv4 unicast
```

```
      allowas-in
      disable-peer-as-check
```

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

    ```
    nv overlay evpn
    ```

  - Enable the relevant protocols

    ```
    feature bgp
    feature pim
    ```

  - Configure Loopback for local Router ID, PIM, and BGP

    ```
    interface loopback0
      ip address 20.1.1.1/32 tag 12345
      ip pim sparse-mode
    ```

  - Configure Loopback for AnycastRP

    ```
    interface loopback1
      ip address 100.1.1.1/32 tag 12345
      ip pim sparse-mode
    ```

  - Configure Anycast RP

    ```
    ip pim rp-address 100.1.1.1 group-list 224.0.0.0/4
    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 NEXT-HOP-UNCH permit 10
      set ip next-hop unchanged
    ```

  - Configure route-map to Redistribute Loopback

    ```
    route-map LOOPBACK permit 10
      match tag 12345
    ```

  - Configure interfaces for Spine-leaf interconnect

    ```
    interface Ethernet4/2
      no switchport
      ip address 192.168.3.42/24
      ip router ospf 1 area 0.0.0.0
      ip pim sparse-mode
      no shutdown

    interface Ethernet4/3
      no switchport
      ip address 192.168.4.43/24
      ip router ospf 1 area 0.0.0.0
    ```

```
  ip pim sparse-mode
  shutdown
```

- Configure BGP overlay for the EVPN address family

```
router bgp 100
  router-id 20.1.1.1
  address-family l2vpn evpn
    nexthop route-map NEXT-HOP-UNCH
    retain route-target all
  neighbor 30.1.1.1 remote-as 200
    update-source loopback0
    ebgp-multihop 3
    address-family l2vpn evpn
      send-community both
      disable-peer-as-check
      route-map NEXT-HOP-UNCH out
  neighbor 40.1.1.1 remote-as 200
    update-source loopback0
    ebgp-multihop 3
    address-family l2vpn evpn
      send-community both
      disable-peer-as-check
      route-map NEXT-HOP-UNCH out
```

- Configure the BGP underlay for the IPv4 unicast address family.

```
  address-family ipv4 unicast
    redistribute direct route-map LOOPBACK
  neighbor 192.168.3.22 remote-as 200
    update-source ethernet4/2
    address-family ipv4 unicast
      allowas-in
      disable-peer-as-check
  neighbor 192.168.4.43 remote-as 200
    update-source ethernet4/3
    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
```

  - 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 Router ID, PIM, and BGP.

```
interface loopback0
  ip address 30.1.1.1/32
  ip pim sparse-mode
```

- Configure Loopback for VTEP.

```
interface loopback1
  ip address 33.1.1.1/32
  ip pim sparse-mode
```

- Configure interfaces for Spine-leaf interconnect.

```
interface Ethernet2/2
  no switchport
  ip address 192.168.1.22/24
  ip pim sparse-mode
  no shutdown

interface Ethernet2/3
  no switchport
  ip address 192.168.4.23/24
  ip pim sparse-mode
  shutdown
```

- Configure route-map to Redistribute Host-SVI (Silent Host).

```
route-map HOST-SVI permit 10
  match tag 54321
```

- Enable PIM RP.

```
ip pim rp-address 100.1.1.1 group-list 224.0.0.0/4
```

- Create VLANs.

```
vlan 1001-1002
```

- Create overlay VRF VLAN and configure vn-segment.

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

- Configure core-facing SVI for VXLAN routing.

```
interface vlan101
 no shutdown
  vrf member vxlan-900001
  ip forward
  no ip redirects
  ipv6 address use-link-local-only
  no ipv6 redirects
```

- Create VLAN and provide mapping toVXLAN.

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

- Create VRF and configure VNI

```
vrf context vxlan-900001
  vni 900001
  rd auto
```

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

```
    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 tag 54321
  ipv6 address 4:1:0:1::1/64 tag 54321
  fabric forwarding mode anycast-gateway

interface vlan1002
  no shutdown
  vrf member vxlan-900001
  ip address 4.2.2.1/24 tag 54321
  ipv6 address 4:2:0:1::1/64 tag 54321
  fabric forwarding mode anycast-gateway
```

- Configure ACL TCAM region for ARP suppression

Note The **hardware access-list tcam region arp-ether 256 double-wide** command is not needed for Cisco Nexus 9300-EX and 9300-FX/FX2/FX3 and 9300-GX platform switches.

```
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 Option 1 for a small number of VNIs. Use Option 2 to leverage the simplified configuration mode.

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
    mcast-group 239.0.0.1
  member vni 2001002
    mcast-group 239.0.0.1
```

Option 2

```
interface nve1
  source-interface loopback1
  host-reachability protocol bgp
  global mcast-group 239.0.0.1 L2
  member vni 2001001
  member vni 2001002
  member vni 2001007-2001010
```

- Configure interfaces for hosts/servers.

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

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

- Configure BGP underlay for the IPv4 unicast address family.

```
router bgp 200
  router-id 30.1.1.1
  address-family ipv4 unicast
    redistribute direct route-map LOOPBACK
  neighbor 192.168.1.42 remote-as 100
    update-source ethernet2/2
    address-family ipv4 unicast
      allowas-in
      disable-peer-as-check
```

```
neighbor 192.168.4.43 remote-as 100
  update-source ethernet2/3
  address-family ipv4 unicast
    allowas-in
    disable-peer-as-check
```

- Configure BGP overlay for the EVPN address family.

```
address-family l2vpn evpn
  nexthop route-map NEXT-HOP-UNCH
  retain route-target all
neighbor 10.1.1.1 remote-as 100
  update-source loopback0
  ebgp-multihop 3
  address-family l2vpn evpn
    send-community both
    disable-peer-as-check
    route-map NEXT-HOP-UNCH out
neighbor 20.1.1.1 remote-as 100
  update-source loopback0
  ebgp-multihop 3
  address-family l2vpn evpn
    send-community both
    disable-peer-as-check
    route-map NEXT-HOP-UNCH out
vrf vxlan-900001
```

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

```
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.

```
nv overlay evpn
```

  - Enable the relevant protocols.

```
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 Router ID, PIM, and BGP.

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

  - Configure Loopback for VTEP.

```
interface loopback1
  ip address 44.1.1.1/32
  ip pim sparse-mode
```

  - Configure interfaces for Spine-leaf interconnect.

```
interface Ethernet2/2
  no switchport
  ip address 192.168.3.22/24
  ip pim sparse-mode
  no shutdown

interface Ethernet2/3
  no switchport
  ip address 192.168.2.23/24
  ip pim sparse-mode
  shutdown
```

  - Configure route-map to Redistribute Host-SVI (Silent Host).

```
route-map HOST-SVI permit 10
  match tag 54321
```

  - Enable PIM RP

```
ip pim rp-address 100.1.1.1 group-list 224.0.0.0/4
```

  - Create VLANs

```
vlan 1001-1002
```

- Create overlay VRF VLAN and configure vn-segment.

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

- Configure core-facing SVI for VXLAN routing.

```
interface vlan101
 no shutdown
  vrf member vxlan-900001
  ip forward
  no ip redirects
  ipv6 address use-link-local-only
  no ipv6 redirects
```

- 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
  rd auto
```

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

```
    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 tag 54321
  ipv6 address 4:1:0:1::1/64 tag 54321
  fabric forwarding mode anycast-gateway

interface vlan1002
  no shutdown
  vrf member vxlan-900001
  ip address 4.2.2.1/24 tag 54321
  ipv6 address 4:2:0:1::1/64 tag 54321
  fabric forwarding mode anycast-gateway
```

- Configure ACL TCAM region for ARP suppression

Note The **hardware access-list tcam region arp-ether 256 double-wide** command is not needed for Cisco Nexus 9300-EX and 9300-FX/FX2/FX3 and 9300-GX platform switches.

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

•

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 leverage the simplified configuration mode.

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
    mcast-group 239.0.0.1
  member vni 2001002
    mcast-group 239.0.0.1
```

Option 2

```
interface nve1
  source-interface loopback1
  host-reachability protocol bgp
  global mcast-group 239.0.0.1 L2
  member vni 2001001
  member vni 2001002
  member vni 2001007-2001010
```

- Configure interfaces for hosts/servers

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

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

- Configure BGP underlay for the IPv4 unicast address family.

```
router bgp 200
  router-id 40.1.1.1
  address-family ipv4 unicast
    redistribute direct route-map LOOPBACK
  neighbor 192.168.3.42 remote-as 100
    update-source ethernet2/2
    address-family ipv4 unicast
      allowas-in
      disable-peer-as-check
  neighbor 192.168.2.43 remote-as 100
    update-source ethernet2/3
    address-family ipv4 unicast
      allowas-in
      disable-peer-as-check
```

- Configure BGP overlay for the EVPN address family.

```
  address-family l2vpn evpn
    nexthop route-map NEXT-HOP-UNCH
    retain route-target all
  neighbor 10.1.1.1 remote-as 100
    update-source loopback0
    ebgp-multihop 3
    address-family l2vpn evpn
      send-community both
      disable-peer-as-check
      route-map NEXT-HOP-UNCH out
  neighbor 20.1.1.1 remote-as 100
    update-source loopback0
    ebgp-multihop 3
    address-family l2vpn evpn
      send-community both
      disable-peer-as-check
      route-map NEXT-HOP-UNCH out
  vrf vxlan-900001
```

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

```
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         State LearnType Uptime   Router-Mac
--------- --------------- ----- --------- -------- -----------------
nve1      30.1.1.1        Up    CP        00:00:38 6412.2574.9f27
```

- **show nve vni**

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

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]
nve1      2001002  225.4.0.1         Up    CP   L2 [1002]
```

- **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
```

Note The **show vxlan interface** command is not supported for the Cisco Nexus 9300-EX, 9300-FX/FX2/FX3, and 9300-GX platform switches.

- **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**

```
leaf3# show bgp l2vpn evpn summary
BGP summary information for VRF default, address family L2VPN EVPN
BGP router identifier 40.0.0.4, local AS number 10
BGP table version is 60, L2VPN EVPN config peers 1, capable peers 1
21 network entries and 21 paths using 2088 bytes of memory
BGP attribute entries [8/1152], BGP AS path entries [0/0]
BGP community entries [0/0], BGP clusterlist entries [1/4]

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down
State/PfxRcd
40.0.0.1        4    10    8570    8565       60    0    0    5d22h 6
leaf3#
```

- **show bgp l2vpn evpn**

```
leaf3# show bgp l2vpn evpn
BGP routing table information for VRF default, address family L2VPN EVPN
BGP table version is 60, local router ID is 40.0.0.4
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-injected
Origin codes: i - IGP, e - EGP, ? - incomplete, | - multipath, & - backup

   Network            Next Hop            Metric     LocPrf     Weight Path
Route Distinguisher: 40.0.0.2:32868
*>i[2]:[0]:[10001]:[48]:[0000.8816.b645]:[0]:[0.0.0.0]/216
                      40.0.0.2                          100          0 i
*>i[2]:[0]:[10001]:[48]:[0011.0000.0034]:[0]:[0.0.0.0]/216
                      40.0.0.2                          100          0 i
```

- **show l2route evpn mac all**

```
leaf3# show l2route evpn mac all
Topology    Mac Address    Prod   Next Hop (s)
----------- -------------- ------ ---------------
101         0000.8816.b645 BGP    40.0.0.2
101         0001.0000.0033 Local  Ifindex 4362086
101         0001.0000.0035 Local  Ifindex 4362086
101         0011.0000.0034 BGP    40.0.0.2
```

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

```
leaf3# show l2route evpn mac-ip all
Topology ID Mac Address    Prod Host IP                 Next Hop (s)
----------- -------------- ---- ------------------------------------------------------
101         0011.0000.0034 BGP  5.1.3.2                      40.0.0.2
102         0011.0000.0034 BGP  5.1.3.2                      40.0.0.2
```

CHAPTER 6

Configuring VXLAN with IPv6 in the Underlay (VXLANv6)

This chapter contains the following sections:

Information About Configuring VXLANv6

VXLAN BGP EVPN is deployed with IPv4 underlay and IPv4 VTEP. Hosts in the overlay can be IPv4 or IPv6. Support is added for VXLAN with IPv6 in the Underlay (VXLANv6) with an IPv6 VTEP. This requires IPv6 versions of the unicast routing protocols and utilizing ingress replication for multi-destination traffic (BUM) in the underlay.

This solution is targeted for deployments where the VTEP is IPv6 only and the underlay is IPv6. The BGP sessions between the leaf and spine are also IPv6. The overlay hosts can be either IPv4 or IPv6.

VXLANv6 feature supports BGP unnumbered peering in the underlay.

The following protocols are supported in the underlay:

- IS-IS
- OSPFv3
- eBGP

Information About vPC and VXLAN with IPv6 in the Underlay (VXLANv6)

vPC VTEPs use vMAC (virtual MAC) with the VIP/PIP feature. vMAC is used with VIP and the system MAC is used with PIP.

In the IPv4 underlay, vMAC is derived from the IPv4 VIP address:

VMAC = 0x02 + 4 bytes IPv4 VIP address.

In the IPv6 underlay, VIP is IPv6 (128 bits) which cannot be used to generate a conflict free unique vMAC (48 bits). The default method is to autogenerate the vMAC by picking the last 48 bits from the IPv6 VIP:

Autogenerated vMAC = 0x06 + the last 4 bytes of the IPv6 VIP address.

If there are two vPC complexes which have different VIPs but the same last 4 bytes of IPv6 address in the VIP, both autogenerate the same vMAC. For a remote VTEP, it sees vMAC flopping between two different VIPs. This is not an issue for Cisco Nexus 9000 Series switches which support VXLAN IPv6.

For other vendor boxes, if this is an issue for interoperability reasons, the vMAC can be manually configured on Cisco Nexus 9000 Series switches to override the autogenerated vMAC. The default behavior for VXLAN with IPv6 in the Underlay (VXLANv6) is to autogenerate the VMAC. If a VMAC is configured manually, the manually configured VMAC takes precedence.

```
interface nve1
    virtual-rmac <48-bit mac address>
```

The VMAC must be managed by the administrator just like the VIP/PIP and must be unique in the fabric. All the preceding behavior is for VXLAN with IPv6 in the Underlay (VXLANv6) only and nothing changes about VMAC creation and advertisement for VXLAN IPv4 in the underlay.

The default behavior is that vMAC is autogenerated from the configured VIP and advertised. There is no need to use the **virtual-rmac** command as previously described except for interoperability cases. There is no need to use the existing **advertise virtual-rmac** command for VXLAN with IPv6 in the Underlay (VXLANv6).

Information About vPC Peer Keepalive and VXLAN with IPv6 in the Underlay (VXLANv6)

The modification for vPC is to allow IPv6 addresses to be used for the peer-keepalive link. The link can be on the management interface or any other interface. The keepalive link becomes operational only when both peers are configured correctly either with the IPv4 or IPv6 address and those addresses are reachable from each peer. Peer-keepalive can be configured on in-band and out-of-band interfaces.

Note peer-keepalive must be a global unicast address.

The configuration command for **peer-keepalive** accepts an IPv6 address

```
vpc domain 1
peer-keepalive destination 001:002::003:004 source 001:002::003:005 vrf management
```

Guidelines and Limitations for VXLAN with IPv6 in the Underlay (VXLANv6)

VXLAN with IPv6 in the Underlay (VXLANv6) has the following guidelines and limitations:

- Dual Stack (IPv4 and IPv6) is not supported for VXLAN underlay. It should either be IPv4 or IPv6, not both.
- NVE Source interface loopback for VTEP can either be IPv4 (VXLANv4) or IPv6 (VXLANv6), and not both.
- Next hop address in overlay (in bgp l2vpn evpn address family updates) should be resolved in underlay URIB to the same address family. For example, the use of VTEP (NVE source loopback) IPv4 addresses in fabric should only have BGP l2vpn evpn peering over IPv4 addresses.
- Usage of IPv6 LLA requires the TCAM Region for **ing-sup** to be re-carved from the default value of 512 to 768. This step requires a copy run start and reload

The following Cisco Nexus platforms are supported to provide the VTEP function (leaf and border). The BGP route reflector can be provided by any Cisco Nexus platform that supports the EVPN **address-family** command over an IPv6 MP-BGP peering.

- Cisco Nexus 9332C
- Cisco Nexus 9364C
- Cisco Nexus 9300-EX
- Cisco Nexus 9300-FX
- Cisco Nexus 9300-FX2
- Cisco Nexus 9300-FX3
- Cisco Nexus 9300-FXP
- Cisco Nexus 9300-GX

VXLAN with IPv6 in the Underlay (VXLANv6) supports the following features:

- Address Resolution Protocol (ARP) suppression in the overlay
- Access Control List (ACL) Quality of Service (QoS)
- Border Node with VRF-Lite
- Dynamic Host Configuration Protocol (DHCP)
- Guestshell support
- Internet Group Management Protocol (IGMP) Snooping in the overlay
- Virtual Extensible Local Area Network (VXLAN) Operation, Administration, and Maintenance (OAM)

- Storm Control for host ports (Access Side)
- Virtual Port Channel (vPC) with VIP and PIP support

VXLAN with IPv6 in the Underlay (VXLANv6) does not support the following features:

- Downstream VNI
- Bidirectional Forwarding Detection (BFD)
- Centralized Route Leak
- Cisco Data Center Network Manager (DCNM) integration
- Cross Connect
- EVPN Multi-homing with Ethernet Segment (ES)
- Fabric Extender (FEX) attached to a VXLAN-enabled switch.
- VXLAN Flood and Learn
- MACsec
- Multiprotocol Label Switching (MPLS) and Locator/ID Separation Protocol (LISP) handoff
- Multicast underlay (PIM-BiDir, Protocol Independent Multicast (PIM) Any Source Multicast (ASM), Snooping)
- NetFlow
- Overlay IGMP Snooping
- **peer vtep** command
- Sampled Flow (sFLOW)
- Static ingress replication (IR)
- Tenant Routed Multicast (TRM)
- Virtual Network Functions (VNF) Multipath
- vPC Fabric Peering
- VXLAN Access Features
    - 802.1x
    - Port security
    - Private VLAN (PVLAN)
    - PV translation (Switching/Routing)
    - Q-in-Q with QinVNI
    - QinVNI and SelQinVNI
- VXLAN Policy-Based Routing (PBR)
- VXLAN Multi-Site

Beginning with Cisco NX-OS Release 10.1(1), IPv6 Underlay is supported for N9K-C9316D-GX, N9K-C93600CD-GX, and N9K-C9364C-GX TOR switches.

Other guidelines and limitations:

- VXLAN/Fibre Channel co-existence

Configuring the VTEP IP Address

SUMMARY STEPS

1. **configure terminal**
2. **interface nve1**
3. **source-interface loopback** *src-if*
4. **exit**
5. **interface loopback** *loopback_number*
6. **ipv6 address** *ipv6_format*
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enter global configuration mode.
Step 2	**interface nve1** **Example:** `switch(config)# interface nve1`	Configure the NVE interface.
Step 3	**source-interface loopback** *src-if* **Example:** `switch(config-if-nve)# source interface loopback 1`	The source interface must be a loopback interface that is configured on the switch with a valid /128 IP address. This /128 IP address must be known by the intermediate devices in the transport network and the remote VTEPs. This is accomplished by advertising it through a dynamic routing protocol in the transport network. **Note** The IPv6 address on **`loopback1`** must be a /128 address. The VTEP IP address cannot be a link local IPv6 address.
Step 4	**exit** **Example:** `switch(config-if-nve)# exit`	Exit configuration mode.

	Command or Action	Purpose
Step 5	**interface loopback** *loopback_number* **Example:** `switch(config)# interface loopback 1`	Configure the loopback interface.
Step 6	**ipv6 address** *ipv6_format* **Example:** `switch(config-if)# ipv6 address 2001:db8:0:0:1:0:0:1/128`	Configure IPv6 address on the interface.
Step 7	**exit** **Example:** `switch(config-if)# exit`	Exit configuration mode.

Configuring vPC for VXLAN with IPv6 in the Underlay (VXLANv6)

VXLAN with IPv4 in the underlay leveraged the concept of a secondary IP address (VIP) used in vPC. IPv6 does not have the concept of secondary addresses as does IPv4. However, multiple IPv6 global addresses can be configured on an interface, which are treated equally in priority.

The CLI for the VIP configuration has been extended to specify the loopback interface that carries the VIP if there is a VXLAN with IPv6 in the Underlay (VXLANv6) vPC. The IPv6 primary IP address (PIP) and VIP are in two separate loopback interfaces.

Similar to IPv4, if there are multiple IPv6 addresses specified on either loopback, the lowest IP is selected for each.

The following steps outline the configuration of a VTEP IP (VIP/PIP) required on a vPC setup.

Note The **anycast loopback** command is used only for VXLAN with IPv6 in the Underlay (VXLANv6).

SUMMARY STEPS

1. **configure terminal**
2. **interface nve1**
3. **source-interface loopback** *src-if* **anycast loopback** *any-if*
4. **exit**
5. **interface loopback** *loopback_number*
6. **ipv6 address** *ipv6_format*
7. **exit**
8. **interface loopback** *loopback_number*
9. **ipv6 address** *ipv6_format*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enter global configuration mode.
Step 2	**interface nve1** **Example:** `switch(config)# interface nve1`	Configure the NVE interface.
Step 3	**source-interface loopback** *src-if* **anycast loopback** *any-if* **Example:** `switch(config-if-nve)# source interface loopback 1 anycast loopback 2`	The source interface must be a loopback interface that is configured on the switch with a valid /128 IP address. This /128 IP address must be known by the transient devices in the transport network and the remote VTEPs. This is accomplished by advertising it through a dynamic routing protocol in the transport network. **Note** The IPv6 address on loopback1, the primary IP address (PIP), and loopback2, the secondary IP address ()VIP), must be a /128 address. The VTEP IP address cannot be a link local IPv6 address.
Step 4	**exit** **Example:** `switch(config-if-nve)# exit`	Exit configuration mode.
Step 5	**interface loopback** *loopback_number* **Example:** `switch(config)# interface loopback 1`	Configure the loopback interface.
Step 6	**ipv6 address** *ipv6_format* **Example:** `switch(config-if)# ipv6 address 2001:db8:0:0:1:0:0:1/128`	Configure IPv6 address on the interface.
Step 7	**exit** **Example:** `switch(config-if)# exit`	Exit configuration mode.
Step 8	**interface loopback** *loopback_number* **Example:** `switch(config)# interface loopback 2`	Configure the loopback interface.
Step 9	**ipv6 address** *ipv6_format* **Example:**	Configure IPv6 address on the interface.

	Command or Action	Purpose
	switch(config-inf)# **ipv6 address 2001:db8:0:0:1:0:0:2/128**	

Example Configurations for VXLAN with IPv6 in the Underlay (VXLANv6)

The following are configuration examples for VXLAN with IPv6 in the Underlay (VXLANv6):

With IPv6 address set/match in next-hop, BGP must set/match the IPv6 next-hop address in route type-2 (MAC-IP) and route type-5 (IP Prefix).

Under route-map:

```
set ipv6 next-hop <vtep address>
match ipv6 next-hop  <vtep address>
```

BGP Underlay

Note BGP IPv6 neighbor must support L2VPN EVPN address-family session.

Note The router ID in VXLAN with IPv6 in the Underlay (VXLANv6) must be an IPv4 address.

The BGP router ID is a 32-bit value that is often represented by an IPv4 address. By default, Cisco NX-OS sets the router ID to the IPv4 address of a loopback interface on the router. For VXLAN with IPv6 in the Underlay (VXLANv6), none of the loopbacks need to have an IPv4 address in which case the default selection of router ID does not happen correctly. You can configure the router ID manually to an IPv4 address.

BGP RD (Route distinguisher) which is 64 bits in length can be configured using the autonomous system number of the 4-byte IP address. For VXLAN with IPv6 in the Underlay (VXLANv6), when using an IP address for configuring RD, you must use IPv4 as in the case of VXLAN IPv4.

```
feature bgp
nv overlay evpn

router bgp 64496
   ! IPv4 router id
   router-id  35.35.35.35
   ! Redistribute the igp/bgp routes
   address-family ipv6 unicast
     redistribute direct route-map allow

   ! For IPv6 session, directly connected peer interface
   neighbor 2001:DB8:0:1::55
     remote-as 64496
     address-family ipv6 unicast
```

OSPFv3 Underlay

```
feature ospfv3

router ospfv3 201
router-id 290.0.2.1

interface ethernet 1/2
ipv6 address 2001:0DB8::1/48
ipv6 ospfv3 201 area 0.0.0.10
```

IS-IS Underlay

```
router isis Enterprise
is-type level-1
net 49.0001.0000.0000.0003.00

interface ethernet 2/1
ipv6 address 2001:0DB8::1/48
isis circuit-type level-1
ipv6 router isis Enterprise
```

Verifying VXLAN with IPv6 in the Underlay (VXLANv6)

To display the status for the VXLAN with IPv6 in the Underlay (VXLANv6) configuration, enter one of the following commands:

Table 2: VXLAN with IPv6 in the Underlay (VXLANv6) Verification Commands

Command	Purpose
show running-config interface nve 1	Displays interface NVE 1 running configuration information.
show nve interface 1 detail	Displays NVE interface detail.
show nve peers	Displays the peering time and VNI information for VTEP peers.
show nve vni ingress-replication	Displays NVE VNI ingress replication information.
show nve peers 2018:1015::abcd:1234:3 int nv1 counters	Displays NVE peers counter information.
show bgp l2vpn evpn 1012.0383.9600	Displays BGP L2VPN information for route type 2.
show bgp l2vpn evpn 303:304::1	Displays BGP L2VPN EVPN for route type 3.
show bgp l2vpn evpn 5.116.204.0	Displays BGP L2VPN EVPN for route type 5.
show l2route peerid	Displays L2route peerid.
show l2route topology detail	Displays L2route topology detail.
show l2route evpn imet all detail	Displays L2route EVPN imet detail.

Command	Purpose
show l2route fl all	Display L2route flood list detail.
show l2route mac all detail	Displays L2route MAC detail.
show l2route mac-ip all detail	Displays MAC address and host IP address.
show ip route 1.191.1.0 vrf vxlan-10101	Displays route table for VRF.
show forwarding ipv4 route 1.191.1.0 detail vrf vxlan-10101	Displays forwarding information.
show ipv6 route vrf vxlan-10101	Displays IPv6 routing table.
show bgp l2vpn evpn	Displays BGP's updated routes.
show bgp evi *evi-id*	Displays BGP EVI information.
show forwarding distribution peer-id	Displays forwarding information.
show forwarding nve l2 ingress-replication-peers	Displays forwarding information for ingress replication.
show forwarding nve l3 peers	Displays nv3 Layer 3 peers information.
show forwarding ecmp platform	Displays forwarding ECMP platform information.
show forwarding ecmp platform	Displays forwarding ECMP platform information.
show forwarding nve l3 ecmp	Displays forwarding NVE Layer 3 ECMP information.

Example of the **show running-config interface nve 1**

Command

```
switch# show running-config interface nve 1
interface nve1
  no shutdown
  source-interface loopback1 anycast loopback2
  host-reachability protocol bgp
  member vni 10011
    ingress-replication protocol bgp
  member vni 20011 associate-vrf
```

Example of the **show nve interface 1 detail**

Command

```
switch# show nve interface nve 1 detail
Interface: nve1, State: Up, encapsulation: VXLAN
 VPC Capability: VPC-VIP-Only [notified]
 Local Router MAC: a093.51cf.78f7
 Host Learning Mode: Control-Plane
 Source-Interface: loopback1 (primary: 30:3:1::2)
 Anycast-Interface: loopback2 (secondary: 303:304::1)
 Source Interface State: Up
 Anycast Interface State: Up
 Virtual RMAC Advertisement: Yes
```

```
NVE Flags:
Interface Handle: 0x49000001
Source Interface hold-down-time: 745
Source Interface hold-up-time: 30
Remaining hold-down time: 0 seconds
Virtual Router MAC: 0600.0000.0001
Interface state: nve-intf-add-complete
```

Example of the **show nve peers** Command

```
switch# show nve peers
Interface Peer-IP          State LearnType Uptime   Router-Mac
--------- ---------------  ----- --------- -------- -----------------
nve1      1:1::1:1          Up    CP       00:44:09   5087.89d4.6bb7
```

Up

Example of the **show nve vni ingress-replication**

Command

```
switch# show nve vni ingress-replication
Interface VNI      Replication List  Source  Up Time
--------- -------- ----------------- ------- -------

nve1      10011    1:1::1:1           BGP-IMET   00:46:55
```

Example of the **show nve peers** *ipv6-address* **int nv1 counters** Command .

```
switch# show nve peers 2018:2015::abcd:1234:3 int nve 1 counters
Peer IP: 2018:1015::abcd:1234:3
TX
        0 unicast packets 0 unicast bytes
        0 multicast packets 0 multicast bytes
RX
        0 unicast packets 0 unicast bytes
        0 multicast packets 0 multicast bytes
```

Example of the **show bgp l2vpn evpn** Command for Route-Type 2.

```
switch# show bgp l2vpn evpn 1012.0383.9600
BGP routing table information for VRF default, address family L2VPN EVPN
Route Distinguisher: 30.3.1.1:34067    (L2VNI 2001300)
BGP routing table entry for [2]:[0]:[0]:[48]:[1012.0383.9600]:[0]:[0.0.0.0]/216, version
1051240
Paths: (1 available, best #1)
Flags: (0x000102) (high32 00000000) on xmit-list, is not in l2rib/evpn
Multipath: iBGP

  Advertised path-id 1
  Path type: local, path is valid, is best path, no labeled nexthop
  AS-Path: NONE, path locally originated
    303:304::1 (metric 0) from 0:: (30.3.1.1)
      Origin IGP, MED not set, localpref 100, weight 32768
      Received label 2001300
      Extcommunity: RT:2:2001300 ENCAP:8

  Path-id 1 advertised to peers:
    2::21              2::66
BGP routing table entry for [2]:[0]:[0]:[48]:[1012.0383.9600]:[32]:[4.231.115.2]/272, version
 1053100
Paths: (1 available, best #1)
```

```
Flags: (0x000102) (high32 00000000) on xmit-list, is not in l2rib/evpn
Multipath: iBGP

  Advertised path-id 1
  Path type: local, path is valid, is best path, no labeled nexthop
  AS-Path: NONE, path locally originated
    303:304::1 (metric 0) from 0:: (30.3.1.1)
      Origin IGP, MED not set, localpref 100, weight 32768
      Received label 2001300 3003901
      Extcommunity: RT:2:2001300 RT:2:3003901 ENCAP:8 Router MAC:0600.0000.0001

  Path-id 1 advertised to peers:
    2::21              2::66
```

Example of the **show bgp l2vpn evpn** Command for Route-Type 3

```
switch# show bgp l2vpn evpn 303:304::1
BGP routing table information for VRF default, address family L2VPN EVPN
Route Distinguisher: 30.3.1.1:32769    (L2VNI 2000002)
BGP routing table entry for [3]:[0]:[128]:[303:304::1]/184, version 1045060
Paths: (1 available, best #1)
Flags: (0x000002) (high32 00000000) on xmit-list, is not in l2rib/evpn
Multipath: iBGP

  Advertised path-id 1
  Path type: local, path is valid, is best path, no labeled nexthop
  AS-Path: NONE, path locally originated
    303:304::1 (metric 0) from 0:: (30.3.1.1)
      Origin IGP, MED not set, localpref 100, weight 32768
      Extcommunity: RT:2:2000002 ENCAP:8
      PMSI Tunnel Attribute:
        flags: 0x00, Tunnel type: Ingress Replication
        Label: 2000002, Tunnel Id: 303:304::1

  Path-id 1 advertised to peers:
    2::21              2::66
```

Example of the **show bgp l2vpn evpn** Command for Route-Type 5

```
switch# show bgp l2vpn evpn 5.116.204.0
BGP routing table information for VRF default, address family L2VPN EVPN
Route Distinguisher: 2.0.0.52:302
BGP routing table entry for [5]:[0]:[0]:[24]:[5.116.204.0]/224, version 119983
Paths: (2 available, best #2)
Flags: (0x000002) (high32 00000000) on xmit-list, is not in l2rib/evpn, is not in HW
Multipath: iBGP

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

  Gateway IP: 0.0.0.0
  AS-Path: 65001 5300 , path sourced external to AS
    3::52 (metric 200) from 2::66 (2.0.0.66)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 3003301
      Extcommunity: RT:2:3003301 ENCAP:8 Router MAC:f80b.cb53.4897
      Originator: 2.0.0.52 Cluster list: 2.0.0.66

  Advertised path-id 1
  Path type: internal, path is valid, is best path, no labeled nexthop
             Imported to 2 destination(s)
             Imported paths list: evpn-tenant-0301 default
  Gateway IP: 0.0.0.0
  AS-Path: 65001 5300 , path sourced external to AS
```

```
    3::52 (metric 200) from 2::21 (2.0.0.21)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 3003301
      Extcommunity: RT:2:3003301 ENCAP:8 Router MAC:f80b.cb53.4897
      Originator: 2.0.0.52 Cluster list: 2.0.0.21

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

Example of the **show l2route peerid** Command

```
switch# show l2route peerid
NVE Ifhdl    IP Address                              PeerID       Ifindex      Num of MAC's
 Num of NH's
------------ --------------------------------------- ------------ ------------ ------------
 ------------
1224736769   4999:1::1:1:1                            4            1191182340   23377
  0
```

Example of the **show l2route topology detail** Command

```
switch# show l2route topology detail
Flags:(L2cp)=L2 Ctrl Plane; (Dp)=Data Plane; (Imet)=Data Plane BGP IMET; (L3cp)=L3 Ctrl
Plane; (Bfd)=BFD over Vxlan; (Bgp)=BGP EVPN; (Of)=Open Flow mode; (Mix)=Open Flow IR mixed
 mode; (Acst)=Anycast GW on spine;
Topology ID   Topology Name   Attributes
-----------   -------------   ----------
101           Vxlan-10101     VNI: 10101
                              Encap:1 IOD:0 IfHdl:1224736769
                              VTEP IP: 5001:1::1:1:7
                              Emulated IP: ::
                              Emulated RO IP: 0.0.0.0
                              TX-ID: 2004 (Rcvd Ack: 0)
                              RMAC: 00fe.c83e.84a7, VRFID: 3
                              VMAC: 00fe.c83e.84a7
                              VMAC RO: 0000.0000.0000
                              Flags: L3cp, Sub_Flags: --, Prev_Flags: -
```

Example of the **show l2route evpn imet all detail** Command

```
switch# show l2route evpn imet all detail
Flags- (F): Originated From Fabric, (W): Originated from WAN

Topology ID  VNI   Prod  IP Addr        Eth Tag PMSI-Flags Flags   Type Label(VNI)  Tunnel
 ID     NFN Bitmap
-----------  ----- ----- -------------- ------- ---------- ------- ---- -----------
------------- ----------
901          10901 BGP   4999:1::1:1:1  0       0          -       6    10901
4999:1::1:1:1
```

Example of the **show l2route fl all** Command

```
switch# show l2route fl all
Topology ID Peer-id     Flood List                              Service Node
----------- ----------- --------------------------------------- ------------
901         4           4999:1::1:1:1                           no
```

Example of the **show l2route mac all detail** Command

```
switch# show l2route mac all detail

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, (Orp): Orphan

Topology    Mac Address    Prod   Flags         Seq No     Next-Hops

----------- -------------- ------ ------------- ----------
---------------------------------------
901         0016.0901.0001 BGP    SplRcv        0          6002:1::1:1:1

            Route Resolution Type: Regular
            Forwarding State: Resolved (PeerID: 2)
            Sent To: L2FM
          Encap: 1
```

Example of the **show l2route mac-ip all detail** Command

```
switch# show l2route mac-ip all detail
 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 (Orp):Orphan
Topology    Mac Address    Host IP                                 Prod   Flags         Seq
 No     Next-Hops
----------- -------------- --------------------------------------- ------ ----------
---------- ---------------
901         0016.0901.0001 46.1.1.101                               BGP    --            0
        6002:1::1:1:1
            Sent To: ARP
            encap-type:1
```

Example of the **show ip route 1.191.1.0 vrf vxlan-10101** Command

```
switch# show ip route 1.191.1.0 vrf vxlan-10101
IP Route Table for VRF "vxlan-10101"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

1.191.1.0/29, ubest/mbest: 6/0
    *via fe80::2fe:c8ff:fe09:8fff%default, Po1001, [200/0], 00:56:21, bgp-4002, internal,
tag 4007 (evpn)
segid: 10101 VTEP:(5001:1::1:1:1, underlay_vrf: 1) encap: VXLAN

*via fe80::2fe:c8ff:fe09:8fff%default, Po1002, [200/0], 00:56:21, bgp-4002, internal, tag
4007 (evpn)
segid: 10101 VTEP:(5001:1::1:1:1, underlay_vrf: 1) encap: VXLAN

    *via fe80::2fe:c8ff:fe09:8fff%default, Po1001, [200/0], 00:56:32, bgp-4002, internal,
tag 4007 (evpn)
segid: 10101 VTEP:(5001:1::1:1:2, underlay_vrf: 1) encap: VXLAN

    *via fe80::2fe:c8ff:fe09:8fff%default, Po1002, [200/0], 00:56:32, bgp-4002, internal,
tag 4007 (evpn)
segid: 10101 VTEP:(5001:1::1:1:2, underlay_vrf: 1) encap: VXLAN
```

Example of the **show forwarding ipv4 route 1.191.1.0 detail vrf vxlan-10101** Command

```
switch# show forwarding ipv4 route 1.191.1.0 detail vrf vxlan-10101

slot  1
=======
Prefix 1.191.1.0/29, No of paths: 2, Update time: Mon Apr 15 15:38:17 2019
```

```
   5001:1::1:1:1      nve1
   5001:1::1:1:2      nve1
```

Example of the **show ipv6 route vrf vxlan-10101** Command

```
switch# show ipv6 route vrf vxlan-10101
IPv6 Routing Table for VRF "vxlan-10101"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]

2:2:2::101/128, ubest/mbest: 1/0
    *via 5001:1::1:1:1/128%default, [200/0], 00:55:31, bgp-4002, internal, tag 4002 (evpn)
 segid 10101
VTEP:(5001:1::1:1:1, underlay_vrf: 1) encap: VXLAN
```

Example of the **show forwarding distribution peer-id**

Command

```
switch# show forwarding distribution peer-id
UFDM Peer-id allocations: App id 0
App: VXLAN   Vlan: 1     Id: 4999:1::1:1:1 0x49030001 Peer-id: 0x6
App: VXLAN   Vlan: 1     Id: 5001:1::1:1:1 0x49030001 Peer-id: 0x2
App: VXLAN   Vlan: 1     Id: 5001:1::1:1:2 0x49030001 Peer-id: 0x1
App: VXLAN   Vlan: 1     Id: 5001:1::1:1:7 0x49030001 Peer-id: 0x7
App: VXLAN   Vlan: 1     Id: 5001:1::1:2:101 0x49030001 Peer-id: 0x8
App: VXLAN   Vlan: 1     Id: 5001:1::1:2:102 0x49030001 Peer-id: 0x5
App: VXLAN   Vlan: 1     Id: 5001:1::1:2:103 0x49030001 Peer-id: 0x9
App: VXLAN   Vlan: 1     Id: 5001:1::1:2:104 0x49030001 Peer-id: 0xa
App: VXLAN   Vlan: 1     Id: 5001:1::1:2:105 0x49030001 Peer-id: 0xb
App: VXLAN   Vlan: 1     Id: 5001:1::1:2:106 0x49030001 Peer-id: 0xc
App: VXLAN   Vlan: 1     Id: 5001:1::1:2:107 0x49030001 Peer-id: 0xd
```

Example of the **show forwarding nve l2 ingress-replication-peers**

Command

```
switch# show forwarding nve l2 ingress-replication-peers
slot  1
=======

Total count of VLANS with ingr-repl peers: 1950
VLAN 1024 VNI 0 Vtep Ifindex 0x0 plt_space : 0x1ca75e14
         peer :  6002:1::1:1:1
         peer :  5001:1::1:1:7
         peer :  4999:1::1:1:1

PSS VLAN:1024, VNI:0, vtep:0x0x0, peer_cnt:3
         peer :  6002:1::1:1:1  marked : 0
         peer :  5001:1::1:1:7  marked : 0
         peer :  4999:1::1:1:1  marked : 0
 VLAN 1280 VNI 0 Vtep Ifindex 0x0 plt_space : 0x1ca75e14
         peer :  6002:1::1:1:1
         peer :  5001:1::1:1:7
         peer :  4999:1::1:1:1

PSS VLAN:1280, VNI:0, vtep:0x0x0, peer_cnt:3
         peer :  6002:1::1:1:1  marked : 0
         peer :  5001:1::1:1:7  marked : 0
         peer :  4999:1::1:1:1  marked : 0
```

Example of the **show forwarding nve l3 peers**

Command

```
switch# show forwarding nve l3 peers
slot  1
=======

EVPN configuration state: disabled, PeerVni Adj enabled
NVE cleanup transaction-id 0
tunnel_id    Peer_id  Peer_address   Interface       rmac         origin state del count
------------------------------------------------------------------------------------------
0x0        1225261062 4999:1::1:1:1        nve1      0600.0001.0001 URIB       merge-done
no    100
0x0        1225261058 5001:1::1:1:1        nve1      2cd0.2d51.9f1b NVE        merge-done
no    100
0x0        1225261057 5001:1::1:1:2        nve1      00a6.cab6.bbbb NVE        merge-done
no    100
0x0        1225261063 5001:1::1:1:7        nve1      00fe.c83e.84a7 URIB       merge-done
no    100
0x0        1225261064 5001:1::1:2:101      nve1      0000.5500.0001 URIB       merge-done
no    100
0x0        1225261061 5001:1::1:2:102      nve1      0000.5500.0002 URIB       merge-done
no    100
0x0        1225261065 5001:1::1:2:103      nve1      0000.5500.0003 URIB       merge-done
no    100
0x0        1225261066 5001:1::1:2:104      nve1      0000.5500.0004 URIB       merge-done
no    100
0x0        1225261067 5001:1::1:2:105      nve1      0000.5500.0005 URIB       merge-done
no    100
```

Example of the **show forwarding ecmp platform**

Command

```
switch# show forwarding ecmp platform
slot  1
=======



ECMP Hash: 0x198b8aae, Num Paths: 2, Hw index: 0x17532
Partial Install: No
Hw ecmp-index: unit-0:1073741827 unit-1:0 unit-2:0, cmn-index: 95538
Hw NVE ecmp-index: unit-0:0 unit-1:0 unit-2:0, cmn-index: 95538
Refcount: 134, Holder: 0x0,   Intf: Ethernet1/101, Nex-Hop: fe80:7::1:2
    Hw adj: unit-0:851977 unit-1:0 unit-2:0, cmn-index: 500010 LIF:4211
  Intf: Ethernet1/108, Nex-Hop: fe80:8::1:2
    Hw adj: unit-0:851978 unit-1:0 unit-2:0, cmn-index: 500012 LIF:4218
   VOBJ count: 0, VxLAN VOBJ count: 0, VxLAN: 0


ECMP Hash: 0x2bb2905e, Num Paths: 3, Hw index: 0x17533
Partial Install: No
Hw ecmp-index: unit-0:1073741828 unit-1:0 unit-2:0, cmn-index: 95539
Hw NVE ecmp-index: unit-0:0 unit-1:0 unit-2:0, cmn-index: 95539
Refcount: 16, Holder: 0x0,   Intf: Ethernet1/101, Nex-Hop: fe80:7::1:2
    Hw adj: unit-0:851977 unit-1:0 unit-2:0, cmn-index: 500010 LIF:4211
  Intf: Ethernet1/108, Nex-Hop: fe80:8::1:2
    Hw adj: unit-0:851978 unit-1:0 unit-2:0, cmn-index: 500012 LIF:4218
  Intf: port-channel1003, Nex-Hop: fe80:9::1:2
    Hw adj: unit-0:851976 unit-1:0 unit-2:0, cmn-index: 500011 LIF:4106
   VOBJ count: 0, VxLAN VOBJ count: 0, VxLAN: 0
```

Example of the **show forwarding ecmp recursive** Command

```
switch# show forwarding ecmp recursive
slot  1
=======

Virtual Object 17 (vxlan):
    Hw vobj-index (0): unit-0:851976 unit-1:0 unit-2:0, cmn-index: 99016
    Hw NVE vobj-index (0): unit-0:0 unit-1:0 unit-2:0, cmn-index: 99016
    Hw vobj-index (1): unit-0:0 unit-1:0 unit-2:0, cmn-index: 0
    Hw NVE vobj-index (1): unit-0:0 unit-1:0 unit-2:0 cmn-index: 0
    Num prefixes : 1
Partial Install: No
    Active paths:
        Recursive NH 5001:1::1:2:10a/128 , table 0x80000001
    CNHs:
        fe80:9::1:2, port-channel1003
        Hw adj: unit-0:851976 unit-1:0 unit-2:0, cmn-index: 500011, LIF:4106
        Hw NVE adj: unit-0:0 unit-1:0 unit-2:0, cmn-index: 500011, LIF:4106
    Hw instance new : (0x182c8, 99016) ls count new 1
    FEC: fec_type 0
        VOBJ Refcount : 1
Virtual Object 167 (vxlan): ECMP-idx1:0x17536(95542), ECMP-idx2:0x0(0),
    Hw vobj-index (0): unit-0:1073741832 unit-1:0 unit-2:0, cmn-index: 99166
    Hw NVE vobj-index (0): unit-0:3 unit-1:0 unit-2:0, cmn-index: 99166
    Hw vobj-index (1): unit-0:0 unit-1:0 unit-2:0, cmn-index: 0
    Hw NVE vobj-index (1): unit-0:0 unit-1:0 unit-2:0 cmn-index: 0
    Num prefixes : 1
Partial Install: No
    Active paths:
        Recursive NH 5001:1::1:3:125/128 , table 0x80000001
    CNHs:
        fe80:7::1:2, Ethernet1/101
        Hw adj: unit-0:851977 unit-1:0 unit-2:0, cmn-index: 500010, LIF:4211
        Hw NVE adj: unit-0:0 unit-1:0 unit-2:0, cmn-index: 500010, LIF:4211
        fe80:8::1:2, Ethernet1/108
        Hw adj: unit-0:851978 unit-1:0 unit-2:0, cmn-index: 500012, LIF:4218
        Hw NVE adj: unit-0:0 unit-1:0 unit-2:0, cmn-index: 500012, LIF:4218
    Hw instance new : (0x1835e, 99166) ls count new 2
    FEC: fec_type 0
        VOBJ Refcount : 1
```

Example of the **show forwarding nve l3 ecmp** Command

```
switch# show forwarding nve l3 ecmp
slot  1
=======


ECMP Hash: 0x70a50e4, Num Paths: 2, Hw Index: 0x17534
table_id: 403, flags: 0x0, adj_flags: 0x0, Ref-ct: 101
  tunnel_id: 5001:1::1:1:1, segment_id: 10101
  tunnel_id: 5001:1::1:1:2, segment_id: 10101
Hw ecmp-index: unit0: 1073741830 unit1: 0 unit2: 0

ECMP Hash: 0x1189f35e, Num Paths: 2, Hw Index: 0x17535
table_id: -2147483245, flags: 0x0, adj_flags: 0x0, Ref-ct: 50
  tunnel_id: 5001:1::1:1:1, segment_id: 10101
  tunnel_id: 5001:1::1:1:2, segment_id: 10101
```

```
Hw ecmp-index: unit0: 1073741831 unit1: 0 unit2: 0
```

CHAPTER 7

Configuring External VRF Connectivity and Route Leaking

This chapter contains the following sections:

Configuring External VRF Connectivity

About External Layer-3 Connectivity for VXLAN BGP EVPN Fabrics

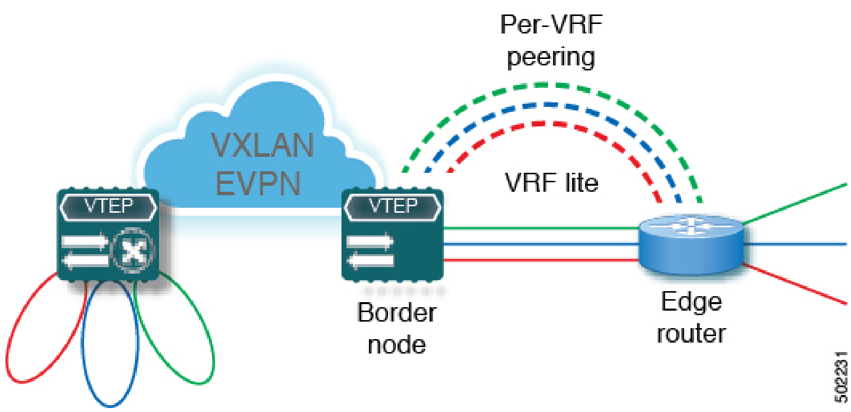
A VXLAN BGP EVPN fabric can be extended by using per-VRF IP routing to achieve external connectivity. The approach that is used for the Layer-3 extensions is commonly referred to as VRF Lite, while the functionality itself is more accurately defined as Inter-AS Option A or back-to-back VRF connectivity.

VXLAN BGP EVPN - VRF-lite brief

Some pointers are given below:

- The VXLAN BGP EVPN fabrics is depicted on the left in the following figure.
- Routes within the fabric are exchanged between all Edge-Devices (VTEPs) as well as Route-Reflectors; the control-plane used is MP-BGP with EVPN address-family.
- The Edge-Devices (VTEPs) acting as border nodes are configured to pass on prefixes to the external router (ER). This is achieved by exporting prefixes from MP-BGP EVPN to IPv4/IPv6 per-VRF peerings.
- Various routing protocols can be used for the per-VRF peering. While eBGP is the protocol of choice, IGPs like OSPF, IS-IS or EIGRP can be leveraged but require redistribution

Figure 16: External Layer-3 Connectivity - VRF-lite



Guidelines and Limitations for External VRF Connectivity and Route Leaking

The following guidelines and limitations apply to external Layer 3 connectivity for VXLAN BGP EVPN fabrics:

- Support is added for Cisco Nexus 9504 and 9508 platform switches with Cisco Nexus 96136YC-R and 9636C-RX line cards.
- A physical Layer 3 interface (parent interface) can be used for external Layer 3 connectivity (that is, VRF default).
- The parent interface to multiple subinterfaces cannot be used for external Layer 3 connectivity (that is, Ethernet1/1 for a VRF default). You can use a subinterface instead.
- Beginning with Cisco NX-OS Release 9.3(5), VTEPs support VXLAN-encapsulated traffic over parent interfaces if subinterfaces are configured.
- VTEPs do not support VXLAN-encapsulated traffic over subinterfaces, regardless of VRF participation or IEEE 802.1Q encapsulation.
- Mixing subinterfaces for VXLAN and non-VXLAN VLANs is not supported.
- The **import map** command applied under address-family ipv4 unicast does not control what gets imported into the EVPN table L3VNI counterpart.
- If TRM is configured, SVIs must not be used to interconnect to the external router.

Configuring VXLAN BGP EVPN with eBGP for VRF-lite

Configuring VRF for VXLAN Routing and External Connectivity using BGP

Configure the VRF on the border node.

SUMMARY STEPS

1. **configure terminal**
2. **vrf context** *vrf-name*
3. **vni** *number*
4. **rd** {**auto** | *rd*}

5. **address-family** {**ipv4** | **ipv6**} **unicast**
6. **route-target both** {**auto** | *rt*}
7. **route-target both** {**auto** | *rt*} **evpn**
8. Repeat Step 1 through Step 7 for every L3VNI.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**vrf context** *vrf-name*	Configure the VRF.
Step 3	**vni** *number*	Specify the VNI. The VNI associated with the VRF is often referred to as a Layer 3 VNI, L3VNI, or L3VPN. The L3VNI is configured as the common identifier across the participating VTEPs.
Step 4	**rd** {**auto**	*rd*}	Specify the VRF's route distinguisher (RD). The RD uniquely identifies a VTEP within an L3VNI. If you enter an RD, the following formats are supported: ASN2:NN, ASN4:NN, or IPV4:NN.
Step 5	**address-family** {**ipv4**	**ipv6**} **unicast**	Configure the IPv4 or IPv6 unicast address family.
Step 6	**route-target both** {**auto**	*rt*}	Configure the route target (RT) for import and export of IPv4 prefixes. The RT is used for a per-VRF prefix import/export policy. If you enter an RT, the following formats are supported: ASN2:NN, ASN4:NN, or IPV4:NN. Manually configured RTs are required to support asymmetric VNIs.
Step 7	**route-target both** {**auto**	*rt*} **evpn**	Configure the route target (RT) for import and export of IPv4 prefixes. The RT is used for a per-VRF prefix import/export policy. If you enter an RT, the following formats are supported: ASN2:NN, ASN4:NN, or IPV4:NN. Manually configured RTs are required to support asymmetric VNIs.
Step 8	Repeat Step 1 through Step 7 for every L3VNI.	

Configuring the L3VNI's Fabric Facing VLAN and SVI on the Border Node

SUMMARY STEPS

1. **configure terminal**
2. **vlan** *number*
3. **vn-segment** *number*
4. **interface** *vlan-number*
5. **mtu** *value*
6. **vrf member** *vrf-name*
7. **ip forward**

8. **no ip redirects**
9. **ipv6** *ip-address*
10. **no ipv6 redirects**
11. Repeat Step 2 through Step 10 for every L3VNI.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enter configuration mode.
Step 2	**vlan** *number*	Specify the VLAN id that is used for the L3VNI.
Step 3	**vn-segment** *number*	Map the L3VNI to the VLAN for VXLAN EVPN routing.
Step 4	**interface** *vlan-number*	Specify the SVI (Switch Virtual Interface) for VXLAN EVPN routing.
Step 5	**mtu** *value*	Specify the MTU for the L3VNI.
Step 6	**vrf member** *vrf-name*	Map the SVI to the matching VRF context.
Step 7	**ip forward**	Enable IPv4 forwarding for the L3VNI.
Step 8	**no ip redirects**	Disable ICMP redirects
Step 9	**ipv6** *ip-address*	Enable IPv6 forwarding for the L3VNI.
Step 10	**no ipv6 redirects**	Disable ICMPv6 redirects.
Step 11	Repeat Step 2 through Step 10 for every L3VNI.	

Configuring the VTEP on the Border Node

SUMMARY STEPS

1. **configure terminal**
2. **interface nve1**
3. **member vni** *vni* **associate-vrf**
4.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**interface nve1**	Configure the NVE interface.
Step 3	**member vni** *vni* **associate-vrf**	Add Layer-3 VNIs, one per tenant VRF, to the overlay.
Step 4		Repeat Step 3 for every L3VNI.

Configuring the BGP VRF Instance on the Border Node for IPv4 per-VRF Peering

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *autonomous-system-number*
3. **vrf** *vrf-name*
4. **address-family ipv4 unicast**
5. **advertise l2vpn evpn**
6. **maximum-paths ibgp** *number*
7. **maximum-paths** *number*
8. **neighbor** *address* **remote-as** *number*
9. **update-source** *type/id*
10. **address-family ipv4 unicast**
11. Repeat Step 3 through Step 10 for every L3VNI that requires external connectivity for IPv4.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**router bgp** *autonomous-system-number*	Configure BGP. The range of the *autonomous-system-number* is from 1 to 4294967295.
Step 3	**vrf** *vrf-name*	Specify the VRF.
Step 4	**address-family ipv4 unicast**	Configure address family for IPv4.
Step 5	**advertise l2vpn evpn**	Enable the advertisement of EVPN routes within IPv4 address-family.
Step 6	**maximum-paths ibgp** *number*	Enabling equal cost multipathing (ECMP) for iBGP prefixes. The range for *number* if 1 to 64. The default is 1.
Step 7	**maximum-paths** *number*	Enabling equal cost multipathing (ECMP) for eBGP prefixes.
Step 8	**neighbor** *address* **remote-as** *number*	Define eBGP neighbor IPv4 address and remote Autonomous-System (AS) number.
Step 9	**update-source** *type/id*	Define interface for eBGP peering.
Step 10	**address-family ipv4 unicast**	Activate the IPv4 address family for IPv4 prefix exchange.
Step 11	Repeat Step 3 through Step 10 for every L3VNI that requires external connectivity for IPv4.	

Configuring the BGP VRF Instance on the Border Node for IPv6 per-VRF Peering

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *autonomous-system-number*
3. **vrf** *vrf-name*
4. **address-family ipv6 unicast**
5. **advertise l2vpn evpn**
6. **maximum-paths ibgp** *number*
7. **maximum-paths** *number*
8. **neighbor** *address* **remote-as** *number*
9. **update-source** *type/id*
10. **address-family ipv6 unicast**
11. Repeat Step 3 Through Step 10 for every L3VNI that requires external connectivity for IPv6.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**router bgp** *autonomous-system-number*	Configure BGP.
Step 3	**vrf** *vrf-name*	Specify the VRF.
Step 4	**address-family ipv6 unicast**	Configure address family for IPv4.
Step 5	**advertise l2vpn evpn**	Enable the advertisement of EVPN routes within IPv6 address-family.
Step 6	**maximum-paths ibgp** *number*	Enabling equal cost multipathing (ECMP) for iBGP prefixes.
Step 7	**maximum-paths** *number*	Enabling equal cost multipathing (ECMP) for eBGP prefixes.
Step 8	**neighbor** *address* **remote-as** *number*	Define eBGP neighbor IPv6 address and remote Autonomous-System (AS) number.
Step 9	**update-source** *type/id*	Define interface for eBGP peering.
Step 10	**address-family ipv6 unicast**	Configure address family for IPv6.
Step 11	Repeat Step 3 Through Step 10 for every L3VNI that requires external connectivity for IPv6.	

Configuring the Sub-Interface Instance on the Border Node for Per-VRF Peering - Version 1

SUMMARY STEPS

1. **configure terminal**

2. interface *type/id*
3. **no switchport**
4. **no shutdown**
5. **exit**
6. interface *type/id*
7. **encapsulation dot1q** *number*
8. **vrf member** *vrf-name*
9. **ip address** *address*
10. **no shutdown**
11. Repeat Step 5 through Step 9 for every per-VRF peering.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	interface *type/id*	Configure parent interface.
Step 3	**no switchport**	Disable Layer-2 switching mode on interface.
Step 4	**no shutdown**	Bring up parent interface.
Step 5	**exit**	Exit interface configuration mode.
Step 6	interface *type/id*	Define the Sub-Interface instance.
Step 7	**encapsulation dot1q** *number*	Configure the VLAN ID for the sub-interface. The *number* argument can have a value from 1 to 3967.
Step 8	**vrf member** *vrf-name*	Map the Sub-Interface to the matching VRF context.
Step 9	**ip address** *address*	Configure the Sub-Interfaces IP address.
Step 10	**no shutdown**	Bring up Sub-Interface.
Step 11	Repeat Step 5 through Step 9 for every per-VRF peering.	

VXLAN BGP EVPN - Default-Route, Route Filtering on External Connectivity

About Configuring Default Routing for External Connectivity

For default-route advertisement into a VXLAN BGP EVPN fabric, we have to ensure that the default-route advertised into the fabric is at the same time not advertised outside of the fabric. For this case, it is necessary to have route filtering in place that prevents this eventuality.

Configuring the Default Route in the Border Nodes VRF

SUMMARY STEPS

1. **configure terminal**
2. **vrf context** *vrf-name*

3. **ip route 0.0.0.0/0** *next-hop*
4. **ipv6 route 0::/0** *next-hop*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**vrf context** *vrf-name*	Configure the VRF.
Step 3	**ip route 0.0.0.0/0** *next-hop*	Configure the IPv4 default-route.
Step 4	**ipv6 route 0::/0** *next-hop*	Configure the IPv6 default-route.

Configuring the BGP VRF Instance on the Border Node for IPv4/IPv6 Default-Route Advertisement

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *autonomous-system-number*
3. **vrf** *vrf-name*
4. **address-family ipv4 unicast**
5. **network 0.0.0.0/0**
6. **address-family ipv6 unicast**
7. **network 0::/0**
8. **neighbor** *address***remote-as** *number*
9. **update-source** *type/id*
10. **address-family {ipv4 | ipv6} unicast**
11. **route-map** *name* **out**
12. Repeat Step 3 through Step 11 for every L3VNI that requires external connectivity with default-route filtering.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**router bgp** *autonomous-system-number*	Configure BGP.
Step 3	**vrf** *vrf-name*	Specify the VRF.
Step 4	**address-family ipv4 unicast**	Configure the IPv4 Unicast address-family. Required for IPv6 over VXLAN with IPv4 underlay.
Step 5	**network 0.0.0.0/0**	Creating IPv4 default-route network statement.
Step 6	**address-family ipv6 unicast**	Configure the IPv6 unicast address-family.
Step 7	**network 0::/0**	Creating IPv6 default-route network statement.

	Command or Action	Purpose
Step 8	**neighbor** *address***remote-as** *number*	Define eBGP neighbor IPv4 address and remote Autonomous-System (AS) number.
Step 9	**update-source** *type/id*	Define interface for eBGP peering
Step 10	**address-family {ipv4 | ipv6} unicast**	Activate the IPv4 or IPv6 address family for IPv4/IPv6 prefix exchange.
Step 11	**route-map** *name* **out**	Attach route-map for egress route filtering.
Step 12	Repeat Step 3 through Step 11 for every L3VNI that requires external connectivity with default-route filtering.	

Configuring Route Filtering for IPv4 Default-Route Advertisement

You can configure route filtering for IPv4 default-route advertisement.

SUMMARY STEPS

1. **configure terminal**
2. **ip prefix-list** *name* **seq 5 permit 0.0.0.0/0**
3. **route-map** *name* **deny 10**
4. **match ip address prefix-list** *name*
5. **route-map** *name* **permit 1000**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**ip prefix-list** *name* **seq 5 permit 0.0.0.0/0**	Configure IPv4 prefix-list for default-route filtering.
Step 3	**route-map** *name* **deny 10**	Create route-map with leading deny statement to prevent the default-route of being advertised via External Connectivity.
Step 4	**match ip address prefix-list** *name*	Match against the IPv4 prefix-list that contains the default-route.
Step 5	**route-map** *name* **permit 1000**	Create route-map with trailing allow statement to advertise non-matching routes via External Connectivity.

Configuring Route Filtering for IPv6 Default-Route Advertisement

SUMMARY STEPS

1. **configure terminal**
2. **ipv6 prefix-list** *name* **seq 5 permit 0::/0**
3. **route-map** *name* **deny 10**
4. **match ipv6 address prefix-list** *name*

5. **route-map** *name* **permit 1000**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**ipv6 prefix-list** *name* **seq 5 permit 0::/0**	Configure IPv6 prefix-list for default-route filtering.
Step 3	**route-map** *name* **deny 10**	Create route-map with leading deny statement to prevent the default-route of being advertised via External Connectivity.
Step 4	**match ipv6 address prefix-list** *name*	Match against the IPv6 prefix-list that contains the default-route.
Step 5	**route-map** *name* **permit 1000**	Create route-map with trailing allow statement to advertise non-matching routes via External Connectivity.

About Configuring Default-Route Distribution and Host-Rote Filter

Per-default, a VXLAN BGP EVPN fabric always advertises all known routes via the External Connectivity. As not in all circumstances it is beneficial to advertise IPv4 /32 or IPv6 /128 Host-Routes, a respective route filtering approach can become necessary.

Configuring the BGP VRF Instance on the Border Node for IPv4/IPv6 Host-Route Filtering

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *autonomous-system-number*
3. **vrf** *vrf-name*
4. **neighbor** *address* **remote-as** *number*
5. **update-source** *type/id*
6. **address-family {ipv4 | ipv6} unicast**
7. **route-map** *name* **out**
8. Repeat Step 3 through Step 7 for every L3VNI that requires external connectivity with host-route filtering.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**router bgp** *autonomous-system-number*	Configure BGP.
Step 3	**vrf** *vrf-name*	Specify the VRF.
Step 4	**neighbor** *address* **remote-as** *number*	Define eBGP neighbor IPv4/IPv6 address and remote Autonomous-System (AS) number.
Step 5	**update-source** *type/id*	Define interface for eBGP peering.

	Command or Action	Purpose
Step 6	**address-family {ipv4 | ipv6} unicast**	Activate the IPv4 or IPv6 address family for IPv4/IPv6 prefix exchange.
Step 7	**route-map** *name* **out**	Attach route-map for egress route filtering.
Step 8	Repeat Step 3 through Step 7 for every L3VNI that requires external connectivity with host-route filtering.	

Configuring Route Filtering for IPv4 Host-Route Advertisement

SUMMARY STEPS

1. **configure terminal**
2. **ip prefix-list** *name* **seq 5 permit 0.0.0.0/0 eq 32**
3. **route-map** *name* **deny 10**
4. **match ip address prefix-list** *name*
5. **route-map** *name* **permit 1000**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**ip prefix-list** *name* **seq 5 permit 0.0.0.0/0 eq 32**	Configure IPv4 prefix-list for host-route filtering.
Step 3	**route-map** *name* **deny 10**	Create route-map with leading deny statement to prevent the default-route of being advertised via External Connectivity.
Step 4	**match ip address prefix-list** *name*	Match against the IPv4 prefix-list that contains the host-route.
Step 5	**route-map** *name* **permit 1000**	Create route-map with trailing allow statement to advertise non-matching routes via external connectivity.

Configuring Route Filtering for IPv6 Host-Route Advertisement

SUMMARY STEPS

1. **configure terminal**
2. **ipv6 prefix-list** *name* **seq 5 permit 0::/0 eq 128**
3. **route-map** *name* **deny 10**
4. **match ipv6 address prefix-list** *name*
5. **route-map** *name* **permit 1000**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**ipv6 prefix-list** *name* **seq 5 permit 0::/0 eq 128**	Configure IPv4 prefix-list for host-route filtering.
Step 3	**route-map** *name* **deny 10**	Create route-map with leading deny statement to prevent the default-route of being advertised via External Connectivity.
Step 4	**match ipv6 address prefix-list** *name*	Match against the IPv4 prefix-list that contains the host-route.
Step 5	**route-map** *name* **permit 1000**	Create route-map with trailing allow statement to advertise non-matching routes via External Connectivity.

Example - Configuring VXLAN BGP EVPN with eBGP for VRF-lite

An example of external connectivity from VXLAN BGP EVPN to an external router using VRF-lite.

Configuring VXLAN BGP EVPN Border Node

The VXLAN BGP EVPN Border Node acts as neighbor device to the External Router. The VRF Name is purely localized and can be different to the VRF Name on the External Router, only significance is the L3VNI must be consistent across the VXLAN BGP EVPN fabric. For the ease of reading, the VRF and interface enumeration will be consistently used.

The configuration examples represents a IPv4 and IPv6 dual-stack approach; IPv4 or IPv6 can be substituted of each other.

```
vrf context myvrf_50001
  vni 50001
  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
!
vlan 2000
  vn-segment 50001
!
interface Vlan2000
  no shutdown
  mtu 9216
  vrf member myvrf_50001
  no ip redirects
  ip forward
  ipv6 address use-link-local-only
  no ipv6 redirects
!
interface nve1
  no shutdown
  host-reachability protocol bgp
  source-interface loopback1
  member vni 50001 associate-vrf
!
```

```
router bgp 65002
  vrf myvrf_50001
    router-id 10.2.0.6
    address-family ipv4 unicast
      advertise l2vpn evpn
      maximum-paths ibgp 2
      maximum-paths 2
    address-family ipv6 unicast
      advertise l2vpn evpn
      maximum-paths ibgp 2
      maximum-paths 2
    neighbor 10.31.95.95
      remote-as 65099
      address-family ipv4 unicast
    neighbor 2001::95/64
      remote-as 65099
      address-family ipv4 unicast
!
interface Ethernet1/3
  no switchport
  no shutdown
interface Ethernet1/3.2
  encapsulation dot1q 2
  vrf member myvrf_50001
  ip address 10.31.95.31/24
  ipv6 address 2001::31/64
  no shutdown
```

Configuring Default-Route, Route Filtering on External Connectivity

The VXLAN BGP EVPN Border Node has the ability to advertise IPv4 and IPv6 default-route within the fabric. In cases where it is not beneficial to advertise the Host Routes from the VXLAN BGP EVPN fabric to the External Router, these IPv4 /32 and IPv6 /128 can be filtered at the External Connectivity peering configuration.

```
ip prefix-list default-route seq 5 permit 0.0.0.0/0 le 1
ipv6 prefix-list default-route-v6 seq 5 permit 0::/0
!
ip prefix-list host-route seq 5 permit 0.0.0.0/0 eq 32
ipv6 prefix-list host-route-v6 seq 5 permit 0::/0 eq 128
!
route-map extcon-rmap-filter deny 10
  match ip address prefix-list default-route
route-map extcon-rmap-filter deny 20
  match ip address prefix-list host-route
route-map extcon-rmap-filter permit 1000
!
route-map extcon-rmap-filter-v6 deny 10
  match ipv6 address prefix-list default-route-v6
route-map extcon-rmap-filter-v6 deny 20
  match ip address prefix-list host-route-v6
route-map extcon-rmap-filter-v6 permit 1000
!
vrf context myvrf_50001
  ip route 0.0.0.0/0 10.31.95.95
  ipv6 route 0::/0 2001::95/64
!
router bgp 65002
  vrf myvrf_50001
    address-family ipv4 unicast
      network 0.0.0.0/0
    address-family ipv6 unicast
      network 0::/0
```

```
    neighbor 10.31.95.95
      remote-as 65099
      address-family ipv4 unicast
        route-map extcon-rmap-filter out
    neighbor 2001::95/64
      remote-as 65099
      address-family ipv4 unicast
        route-map extcon-rmap-filter-v6 out
```

Configuring External Router

The External Router performs as a neighbor device to the VXLAN BGP EVPN border node. The VRF Name is purely localized and can be different to the VRF Name on the VXLAN BGP EVPN Fabric. For the ease of reading, the VRF and interface enumeration will be consistently used.

The configuration examples represents a IPv4 and IPv6 dual-stack approach; IPv4 or IPv6 can be substituted of each other.

```
vrf context myvrf_50001
!
router bgp 65099
  vrf myvrf_50001
    address-family ipv4 unicast
      maximum-paths 2
    address-family ipv6 unicast
      maximum-paths 2
    neighbor 10.31.95.31
      remote-as 65002
      address-family ipv4 unicast
    neighbor 2001::31/64
      remote-as 65002
      address-family ipv4 unicast
!
interface Ethernet1/3
  no switchport
  no shutdown
interface Ethernet1/3.2
  encapsulation dot1q 2
  vrf member myvrf_50001
  ip address 10.31.95.95/24
  Ipv6 address 2001::95/64
  no shutdown
```

Configuring VXLAN BGP EVPN with OSPF for VRF-lite

Configuring VRF for VXLAN Routing and External Connectivity using OSPF

Configure the BGP VRF instance on the border node for OSPF per-VRF peering.

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *autonomous-system-number*
3. **vrf** *vrf-name*
4. **address-family ipv4 unicast**
5. **advertise l2vpn evpn**
6. **maximum-paths ibgp** *number*
7. **redistribute ospf** *name* **route-map** *name*

8. Repeat Step 3 through Step 7 for every per-VRF peering.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**router bgp** *autonomous-system-number*	Configure BGP.
Step 3	**vrf** *vrf-name*	Specify the VRF.
Step 4	**address-family ipv4 unicast**	Configure the IPv4 address family.
Step 5	**advertise l2vpn evpn**	Enable the advertisement of EVPN routes within the address family.
Step 6	**maximum-paths ibgp** *number*	Enabling equal-cost multipathing (ECMP) for iBGP prefixes.
Step 7	**redistribute ospf** *name* **route-map** *name*	Define redistribution from OSPF into BGP.
Step 8	Repeat Step 3 through Step 7 for every per-VRF peering.	

Configuring the Route-Map for BGP to OSPF Redistribution

SUMMARY STEPS

1. **configure terminal**
2. **route-map** *name* **permit 10**
3. **match route-type internal**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**route-map** *name* **permit 10**	Create route-map for BGP to OSPF redistribution
Step 3	**match route-type internal**	Redistribution route-map must allow the matching of BGP internal route-types if iBGP is used in the VXLAN BGP EVPN fabric.

Configuring the OSPF on the Border Node for Per-VRF Peering

SUMMARY STEPS

1. **configure terminal**
2. **router ospf** *instance*
3. **vrf** *vrf-name*
4. **redistribute bgp** *autonomous-system-number* **route-map** *name*
5. Repeat Step 3 through Step 4 for every per-VRF peering.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**router ospf** *instance*	Configure OSPF.
Step 3	**vrf** *vrf-name*	Specify the VRF.
Step 4	**redistribute bgp** *autonomous-system-number* **route-map** *name*	Define redistribution from BGP to OSPF.
Step 5	Repeat Step 3 through Step 4 for every per-VRF peering.	

Configuring the Sub-Interface Instance on the Border Node for Per-VRF Peering - Version 2

SUMMARY STEPS

1. **configure terminal**
2. **interface** *type/id*
3. **no switchport**
4. **no shutdown**
5. **exit**
6. **interface** *type/id*
7. **encapsulation dot1q** *number*
8. **vrf member** *vrf-name*
9. **ip address** *address*
10. **ip ospf network point-to-point**
11. **ip router ospf** *name* **area** *area-id*
12. **no shutdown**
13. Repeat Step 5 through Step 12 for every per-VRF peering.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**interface** *type/id*	Configure parent interface.
Step 3	**no switchport**	Disable Layer-2 switching mode on interface.
Step 4	**no shutdown**	Bring up parent interface.
Step 5	**exit**	Exit interface configuration mode.
Step 6	**interface** *type/id*	Define the Sub-Interface instance.
Step 7	**encapsulation dot1q** *number*	Configure the VLAN ID for the sub-interface. The range is from 2 to 4093.
Step 8	**vrf member** *vrf-name*	Map the Sub-Interface to the matching VRF context.

	Command or Action	Purpose
Step 9	**ip address** *address*	Configure the Sub-Interfaces IP address.
Step 10	**ip ospf network point-to-point**	Define OSPF network-type for sub-interface.
Step 11	**ip router ospf** *name* **area** *area-id*	Configure the OSPF instance.
Step 12	**no shutdown**	Bring up Sub-Interface.
Step 13	Repeat Step 5 through Step 12 for every per-VRF peering.	

Example - Configuration VXLAN BGP EVPN with OSPF for VRF-lite

An example of external connectivity from VXLAN BGP EVPN to an External Router using VRF-lite.

Configuring VXLAN BGP EVPN Border Node with OSPF

The VXLAN BGP EVPN Border Node acts as neighbor device to the External Router. The VRF Name is purely localized and can be different to the VRF Name on the External Router, only significance is the L3VNI must be consistent across the VXLAN BGP EVPN fabric. For the ease of reading, the VRF and interface enumeration will be consistently used.

The configuration examples represents a IPv4 approach with OSPFv2.

```
route-map extcon-rmap-BGP-to-OSPF permit 10
  match route-type internal
route-map extcon-rmap-OSPF-to-BGP permit 10
!
vrf context myvrf_50001
  vni 50001
  rd auto
  address-family ipv4 unicast
    route-target both auto
    route-target both auto evpn
!
vlan 2000
  vn-segment 50001
!
interface Vlan2000
  no shutdown
  mtu 9216
  vrf member myvrf_50001
  no ip redirects
  ip forward
!
interface nve1
  no shutdown
  host-reachability protocol bgp
  source-interface loopback1
  member vni 50001 associate-vrf
!
router bgp 65002
  vrf myvrf_50001
    router-id 10.2.0.6
    address-family ipv4 unicast
      advertise l2vpn evpn
      maximum-paths ibgp 2
      maximum-paths 2
      redistribute ospf EXT route-map extcon-rmap-OSPF-to-BGP
!
```

```
router ospf EXT
  vrf myvrf_50001
    redistribute bgp 65002 route-map extcon-rmap-BGP-to-OSPF
!
interface Ethernet1/3
  no switchport
  no shutdown
interface Ethernet1/3.2
  encapsulation dot1q 2
  vrf member myvrf_50001
  ip address 10.31.95.31/24
  ip ospf network point-to-point
  ip router ospf EXT area 0.0.0.0
  no shutdown
```

Configuring Route Leaking

About Centralized VRF Route-Leaking for VXLAN BGP EVPN Fabrics

VXLAN BGP EVPN uses MP-BGP and its route-policy concept to import and export prefixes. The ability of this very extensive route-policy model allows to leak routes from one VRF to another VRF and vice-versa; any combination of custom VRF or VRF default can be used. VRF route-leaking is a switch-local function at specific to a location in the network, the location where the cross-VRF route-target import/export configuration takes place (leaking point). The forwarding between the different VRFs follows the control-plane, the location of where the configuration for the route-leaking is performed - hence Centralized VRF route-leaking. With the addition of VXLAN BGP EVPN, the leaking point requires to advertise the cross-VRF imported/exported route and advertise them towards the remote VTEPs or External Routers.

The advantage of Centralized VRF route-leaking is that only the VTEP acting as leaking point requires the special capabilities needed, while all other VTEPs in the network are neutral to this function.

Guidelines and Limitations for Centralized VRF Route-Leaking

The following are the guidelines and limitations for Centralized VRF Route-Leaking:

- Each prefix must be imported into each VRF for full cross-VRF reachability.
- The **feature bgp** command is required for the **export vrf default** command.
- If a VTEP has a less specific local prefix in its VRF, the VTEP might not be able to reach a more specific prefix in a different VRF.
- VXLAN routing in hardware and packet reencapsulation at VTEP is required for Centralized VRF Route-Leaking with BGP EVPN.
- Beginning with Cisco NX-OS Release 9.3(5), asymmetric VNIs are used to support Centralized VRF Route-Leaking. For more information, see About VXLAN EVPN with Downstream VNI, on page 80.

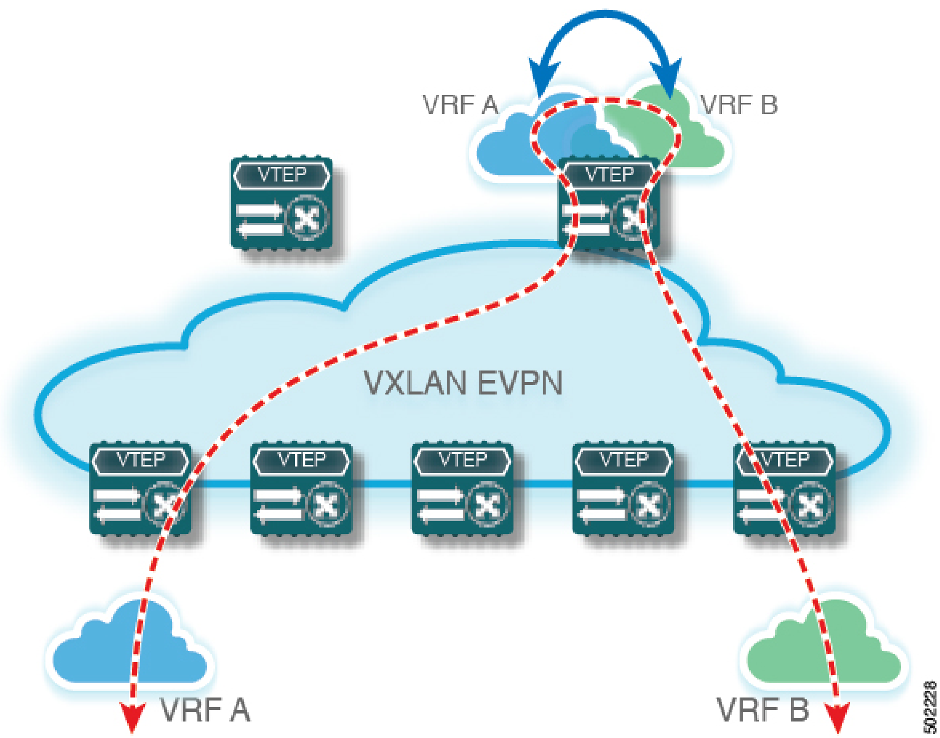
Centralized VRF Route-Leaking Brief - Specific Prefixes Between Custom VRF

Some pointers are given below:

- The Centralized VRF route-leaking for VXLAN BGP EVPN fabrics is depicted within Figure 2.

- BGP EVPN prefixes are cross-VRF leaked by exporting them from VRF Blue with an import into VRF Red and vice-versa. The Centralized VRF route-leaking is performed on the centralized Routing-Block (RBL) and could be any or multiple VTEPs.
- Configured less specific prefixes (aggregates) are advertised from the Routing-Block to the remaining VTEPs in the respective destination VRF.
- BGP EVPN does not export prefixes that were previously imported to prevent the occurrence of routing loops.

Figure 17: Centralized VRF Route-Leaking - Specific Prefixes with Custom VRF



Configuring Centralized VRF Route-Leaking - Specific Prefixes between Custom VRF

Configuring VRF Context on the Routing-Block VTEP

This procedure applies equally to IPv6.

SUMMARY STEPS

1. **configure terminal**
2. **vrf context** *vrf-name*
3. **vni** *number*
4. **rd auto**
5. **address-family ipv4 unicast**

6. **route-target both** {**auto** | *rt*}
7. **route-target both** {**auto** | *rt*} **evpn**
8. **route-target import** *rt-from-different-vrf*
9. **route-target import** *rt-from-different-vrf* **evpn**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**vrf context** *vrf-name*	Configure the VRF.
Step 3	**vni** *number*	Specify the VNI. The VNI associated with the VRF is often referred to as Layer 3 VNI, L3VNI, or L3VPN. The L3VNI is configured as a common identifier across the participating VTEPs.
Step 4	**rd auto**	Specify the VRF's route distinguisher (RD). The RD uniquely identifies a VTEP within an L3VNI.
Step 5	**address-family ipv4 unicast**	Configure the IPv4 unicast address family.
Step 6	**route-target both** {**auto** | *rt*}	Configure the route target (RT) for import and export of IPv4 prefixes. The RT is used for a per-VRF prefix import/export policy. If you enter an RT, the following formats are supported: ASN2:NN, ASN4:NN, or IPV4:NN. Manually configured RTs are required to support asymmetric VNIs.
Step 7	**route-target both** {**auto** | *rt*} **evpn**	Configure the route target (RT) for import and export of IPv4 prefixes. The RT is used for a per-VRF prefix import/export policy. If you enter an RT, the following formats are supported: ASN2:NN, ASN4:NN, or IPV4:NN. Manually configured RTs are required to support asymmetric VNIs.
Step 8	**route-target import** *rt-from-different-vrf*	Configure the RT for importing IPv4 prefixes from the leaked-from VRF. The following formats are supported: ASN2:NN, ASN4:NN, or IPV4:NN.
Step 9	**route-target import** *rt-from-different-vrf* **evpn**	Configure the RT for importing IPv4 prefixes from the leaked-from VRF. The following formats are supported: ASN2:NN, ASN4:NN, or IPV4:NN.

Configuring the BGP VRF instance on the Routing-Block

This procedure applies equally to IPv6.

SUMMARY STEPS

1. **configure terminal**

2. **router bgp** *autonomous-system number*
3. **vrf** *vrf-name*
4. **address-family ipv4 unicast**
5. **advertise l2vpn evpn**
6. **aggregate-address** *prefix/mask*
7. **maximum-paths ibgp** *number*
8. **maximum-paths** *number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**router bgp** *autonomous-system number*	Configure BGP.
Step 3	**vrf** *vrf-name*	Specify the VRF.
Step 4	**address-family ipv4 unicast**	Configure address family for IPv4
Step 5	**advertise l2vpn evpn**	Enable the advertisement of EVPN routes within IPv4 address-family.
Step 6	**aggregate-address** *prefix/mask*	Create less specific prefix aggregate into the destination VRF.
Step 7	**maximum-paths ibgp** *number*	Enabling equal cost multipathing (ECMP) for iBGP prefixes.
Step 8	**maximum-paths** *number*	Enabling equal cost multipathing (ECMP) for eBGP prefixes

Example - Configuration Centralized VRF Route-Leaking - Specific Prefixes Between Custom VRF

Configuring VXLAN BGP EVPN Routing-Block

The VXLAN BGP EVPN Routing-Block acts as centralized route-leaking point. The leaking configuration is localized such that control-plane leaking and data-path forwarding follow the same path. Most significantly is the VRF configuration of the Routing-Block and the advertisement of the less specific prefixes (aggregates) into the respective destination VRFs.

```
vrf context Blue
  vni 51010
  rd auto
  address-family ipv4 unicast
    route-target both auto
    route-target both auto evpn
    route-target import 65002:51020
    route-target import 65002:51020 evpn
!
vlan 2110
  vn-segment 51010
!
interface Vlan2110
  no shutdown
  mtu 9216
```

```
  vrf member Blue
  no ip redirects
  ip forward
!
vrf context Red
  vni 51020
  rd auto
  address-family ipv4 unicast
    route-target both auto
    route-target both auto evpn
    route-target import 65002:51010
    route-target import 65002:51010 evpn
!
vlan 2120
  vn-segment 51020
!
interface Vlan2120
  no shutdown
  mtu 9216
  vrf member Blue
  no ip redirects
  ip forward
!
interface nve1
  no shutdown
  host-reachability protocol bgp
  source-interface loopback1
  member vni 51010 associate-vrf
  member vni 51020 associate-vrf
!
router bgp 65002
  vrf Blue
    address-family ipv4 unicast
      advertise l2vpn evpn
      aggregate-address 10.20.0.0/16
      maximum-paths ibgp 2
      Maximum-paths 2
  vrf Red
    address-family ipv4 unicast
      advertise l2vpn evpn
      aggregate-address 10.10.0.0/16
      maximum-paths ibgp 2
      Maximum-paths 2
```

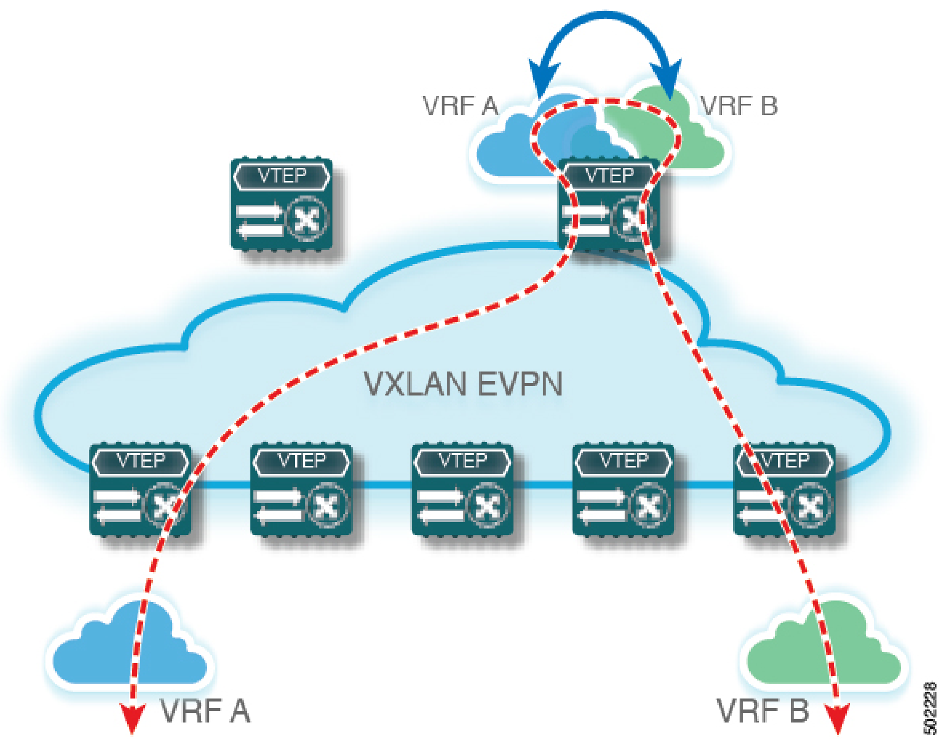
Centralized VRF Route-Leaking Brief - Shared Internet with Custom VRF

Some pointers follow:

- The Shared Internet with VRF route-leaking for VXLAN BGP EVPN fabrics is depicted in the following figure.
- The default-route is made exported from the Shared Internet VRF and re-advertisement within VRF Blue and VRF Red on the Border Node.
- Ensure the default-route in VRF Blue and VRF Red is not leaked to the Shared Internet VRF.
- The less specific prefixes for VRF Blue and VRF Red are exported for the Shared Internet VRF and re-advertised as necessary.
- Configured less specific prefixes (aggregates) that are advertised from the Border Node to the remaining VTEPs to the destination VRF (Blue or Red).

- BGP EVPN does not export prefixes that were previously imported to prevent the occurrence of routing loops.

Figure 18: Centralized VRF Route-Leaking - Shared Internet with Custom VRF



Configuring Centralized VRF Route-Leaking - Shared Internet with Custom VRF

Configuring Internet VRF on Border Node

This procedure applies equally to IPv6.

SUMMARY STEPS

1. **configure terminal**
2. **vrf context** *vrf-name*
3. **vni** *number*
4. **ip route 0.0.0.0/0** *next-hop*
5. **rd auto**
6. **address-family ipv4 unicast**
7. **route-target both** {**auto** | *rt*}
8. **route-target both** *shared-vrf-rt* **evpn**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**vrf context** *vrf-name*	Configure the VRF.
Step 3	**vni** *number*	Specify the VNI. The VNI associated with the VRF is often referred to as Layer 3 VNI, L3VNI, or L3VPN. The L3VNI is configured as a common identifier across the participating VTEPs.
Step 4	**ip route 0.0.0.0/0** *next-hop*	Configure the default route in the shared internet VRF to the external router.
Step 5	**rd auto**	Specify the VRF's route distinguisher (RD). The RD uniquely identifies a VTEP within an L3VNI.
Step 6	**address-family ipv4 unicast**	Configure the IPv4 unicast address family. This configuration is required for IPv4 over VXLAN with IPv4 underlay.
Step 7	**route-target both** {**auto** | *rt*}	Configure the route target (RT) for the import and export of EVPN and IPv4 prefixes. If you enter an RT, the following formats are supported: ASN2:NN, ASN4:NN, or IPV4:NN. Manually configured RTs are required to support asymmetric VNIs.
Step 8	**route-target both** *shared-vrf-rt* **evpn**	Configure a special route target (RT) for the import and export of the shared IPv4 prefixes. An additional import/export map for further qualification is supported.

Configuring Shared Internet BGP Instance on the Border Node

This procedure applies equally to IPv6.

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *autonomous-system number*
3. **vrf** *vrf-name*
4. **address-family ipv4 unicast**
5. **advertise l2vpn evpn**
6. **aggregate-address** *prefix/mask*
7. **maximum-paths ibgp** *number*
8. **maximum-paths** *number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**router bgp** *autonomous-system number*	Configure BGP.
Step 3	**vrf** *vrf-name*	Specify the VRF.
Step 4	**address-family ipv4 unicast**	Configure address family for IPv4
Step 5	**advertise l2vpn evpn**	Enable the advertisement of EVPN routes within IPv4 address-family.
Step 6	**aggregate-address** *prefix/mask*	Create less specific prefix aggregate into the destination VRF.
Step 7	**maximum-paths ibgp** *number*	Enabling equal cost multipathing (ECMP) for iBGP prefixes.
Step 8	**maximum-paths** *number*	Enabling equal cost multipathing (ECMP) for eBGP prefixes.

Configuring Custom VRF on Border Node

This procedure applies equally to IPv6

SUMMARY STEPS

1. **configure terminal**
2. **ip prefix-list** *name* **seq 5 permit 0.0.0.0/0**
3. **route-map** *name* **deny 10**
4. **match ip address prefix-list** *name*
5. **route-map** *name* **permit 20**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**ip prefix-list** *name* **seq 5 permit 0.0.0.0/0**	Configure IPv4 prefix-list for default-route filtering.
Step 3	**route-map** *name* **deny 10**	Create route-map with leading deny statement to prevent the default-route of being leaked.
Step 4	**match ip address prefix-list** *name*	Match against the IPv4 prefix-list that contains the default-route.
Step 5	**route-map** *name* **permit 20**	Create route-map with trailing allow statement to advertise non-matching routes via route-leaking.

Configuring Custom VRF Context on the Border Node - 1

This procedure applies equally to IPv6.

SUMMARY STEPS

1. **configure terminal**
2. **vrf context** *vrf-name*
3. **vni** *number*
4. **rd auto**
5. **ip route 0.0.0.0/0 Null0**
6. **address-family ipv4 unicast**
7. **route-target both** {**auto** | *rt*}
8. **route-target both** {**auto** | *rt*} **evpn**
9. **import map** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**vrf context** *vrf-name*	Configure the VRF.
Step 3	**vni** *number*	Specify the VNI. The VNI associated with the VRF is often referred to as Layer 3 VNI, L3VNI, or L3VPN. The L3VNI is configured as the common identifier across the participating VTEPs.
Step 4	**rd auto**	Specify the VRF's route distinguisher (RD). The RD uniquely identifies a VTEP within an L3VNI.
Step 5	**ip route 0.0.0.0/0 Null0**	Configure default-route in common VRF to attract traffic towards Border Node with Shared Internet VRF.
Step 6	**address-family ipv4 unicast**	Configure the IPv4 address family. This configuration is required for IPv4 over VXLAN with IPv4 underlay.
Step 7	**route-target both** {**auto** | *rt*}	Configure the route target (RT) for the import and export of IPv4 prefixes within the IPv4 address family The RT is used for a per-VRF prefix import/export policy. If you enter an RT, the following formats are supported: ASN2:NN, ASN4:NN, or IPV4:NN. Manually configured RTs are required to support asymmetric VNIs.
Step 8	**route-target both** {**auto** | *rt*} **evpn**	Configure the route target (RT) for the import and export of IPv4 prefixes within the IPv4 address family The RT is used for a per-VRF prefix import/export policy. If you enter an RT, the following formats are supported: ASN2:NN, ASN4:NN, or IPV4:NN. Manually configured RTs are required to support asymmetric VNIs.

	Command or Action	Purpose
Step 9	**import map** *name*	Apply a route-map on routes being imported into this routing table.

Configuring Custom VRF Instance in BGP on the Border Node

This procedure applies equally to IPv6.

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *autonomous-system-number*
3. **vrf** *vrf-name*
4. **address-family ipv4 unicast**
5. **advertise l2vpn evpn**
6. **network 0.0.0.0/0**
7. **maximum-paths ibgp** *number*
8. **maximum-paths** *number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**router bgp** *autonomous-system-number*	Configure BGP.
Step 3	**vrf** *vrf-name*	Specify the VRF.
Step 4	**address-family ipv4 unicast**	Configure address family for IPv4.
Step 5	**advertise l2vpn evpn**	Enable the advertisement of EVPN routes within IPv4 address-family.
Step 6	**network 0.0.0.0/0**	Creating IPv4 default-route network statement.
Step 7	**maximum-paths ibgp** *number*	Enabling equal cost multipathing (ECMP) for iBGP prefixes.
Step 8	**maximum-paths** *number*	Enabling equal cost multipathing (ECMP) for eBGP prefixes.

Example - Configuration Centralized VRF Route-Leaking - Shared Internet with Custom VRF

An example of Centralized VRF route-leaking with Shared Internet VRF

Configuring VXLAN BGP EVPN Border Node for Shared Internet VRF

The VXLAN BGP EVPN Border Node provides a centralized Shared Internet VRF. The leaking configuration is localized such that control-plane leaking and data-path forwarding following the same path. Most significantly is the VRF configuration of the Border Node and the advertisement of the default-route and less specific prefixes (aggregates) into the respective destination VRFs.

```
vrf context Shared
  vni 51099
  ip route 0.0.0.0/0 10.9.9.1
  rd auto
  address-family ipv4 unicast
    route-target both auto
    route-target both auto evpn
    route-target both 99:99
    route-target both 99:99 evpn
!
vlan 2199
  vn-segment 51099
!
interface Vlan2199
  no shutdown
  mtu 9216
  vrf member Shared
  no ip redirects
  ip forward
!
ip prefix-list PL_DENY_EXPORT seq 5 permit 0.0.0.0/0
!
route-map RM_DENY_IMPORT deny 10
 match ip address prefix-list PL_DENY_EXPORT
route-map RM_DENY_IMPORT permit 20
!
vrf context Blue
  vni 51010
  ip route 0.0.0.0/0 Null0
  rd auto
  address-family ipv4 unicast
    route-target both auto
    route-target both auto evpn
    route-target both 99:99
    route-target both 99:99 evpn
    import map RM_DENY_IMPORT
!
vlan 2110
  vn-segment 51010
!
interface Vlan2110
  no shutdown
  mtu 9216
  vrf member Blue
  no ip redirects
  ip forward
!
vrf context Red
  vni 51020
  ip route 0.0.0.0/0 Null0
  rd auto
  address-family ipv4 unicast
    route-target both auto
    route-target both auto evpn
    route-target both 99:99
    route-target both 99:99 evpn
    import map RM_DENY_IMPORT
!
vlan 2120
  vn-segment 51020
!
```

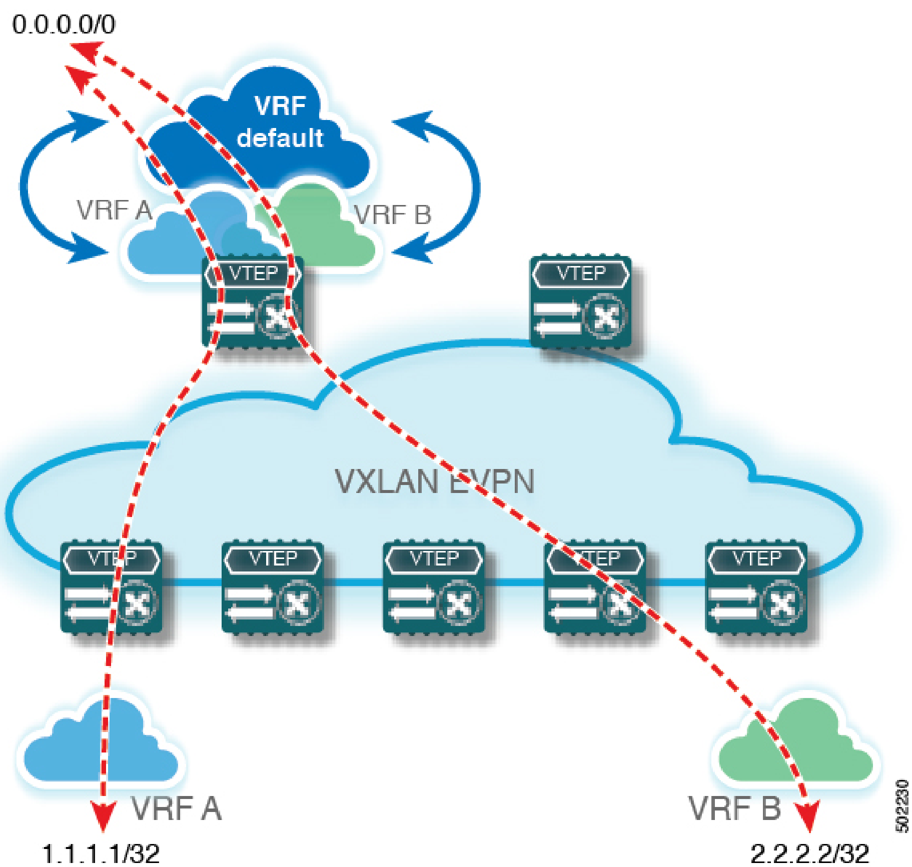
```
interface Vlan2120
  no shutdown
  mtu 9216
  vrf member Blue
  no ip redirects
  ip forward
!
interface nve1
  no shutdown
  host-reachability protocol bgp
  source-interface loopback1
  member vni 51099 associate-vrf
  member vni 51010 associate-vrf
  member vni 51020 associate-vrf
!
router bgp 65002
  vrf Shared
    address-family ipv4 unicast
      advertise l2vpn evpn
      aggregate-address 10.10.0.0/16
      aggregate-address 10.20.0.0/16
      maximum-paths ibgp 2
      maximum-paths 2
  vrf Blue
    address-family ipv4 unicast
      advertise l2vpn evpn
      network 0.0.0.0/0
      maximum-paths ibgp 2
      maximum-paths 2
  vrf Red
    address-family ipv4 unicast
      advertise l2vpn evpn
      network 0.0.0.0/0
      maximum-paths ibgp 2
      maximum-paths 2
```

Centralized VRF Route-Leaking Brief - Shared Internet with VRF Default

Some pointers are given below:

- The Shared Internet with VRF route-leaking for VXLAN BGP EVPN fabrics is depicted within Figure 4.
- The default-route is made exported from VRF default and re-advertisement within VRF Blue and VRF Red on the Border Node.
- Ensure the default-route in VRF Blue and VRF Red is not leaked to the Shared Internet VRF
- The less specific prefixes for VRF Blue and VRF Red are exported to VRF default and re-advertised as necessary.
- Configured less specific prefixes (aggregates) that are advertised from the Border Node to the remaining VTEPs to the destination VRF (Blue or Red).
- BGP EVPN does not export prefixes that were previously imported to prevent the occurrence of routing loops.

Figure 19: Centralized VRF Route-Leaking - Shared Internet with VRF Default



Configuring Centralized VRF Route-Leaking - Shared Internet with VRF Default

Configuring VRF Default on Border Node

This procedure applies equally to IPv6.

SUMMARY STEPS

1. **configure terminal**
2. **ip route 0.0.0.0/0** *next-hop*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**ip route 0.0.0.0/0** *next-hop*	Configure default-route in VRF default to external router (example)

Configuring BGP Instance for VRF Default on the Border Node

This procedure applies equally to IPv6.

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *autonomous-system number*
3. **address-family ipv4 unicast**
4. **aggregate-address** *prefix/mask*
5. **maximum-paths** *number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**router bgp** *autonomous-system number*	Configure BGP.
Step 3	**address-family ipv4 unicast**	Configure address family for IPv4.
Step 4	**aggregate-address** *prefix/mask*	Create less specific prefix aggregate in VRF default.
Step 5	**maximum-paths** *number*	Enabling equal cost multipathing (ECMP) for eBGP prefixes.

Configuring Custom VRF on Border Node

This procedure applies equally to IPv6

SUMMARY STEPS

1. **configure terminal**
2. **ip prefix-list** *name* **seq 5 permit 0.0.0.0/0**
3. **route-map** *name* **deny 10**
4. **match ip address prefix-list** *name*
5. **route-map** *name* **permit 20**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**ip prefix-list** *name* **seq 5 permit 0.0.0.0/0**	Configure IPv4 prefix-list for default-route filtering.
Step 3	**route-map** *name* **deny 10**	Create route-map with leading deny statement to prevent the default-route of being leaked.
Step 4	**match ip address prefix-list** *name*	Match against the IPv4 prefix-list that contains the default-route.

	Command or Action	Purpose
Step 5	**route-map** *name* **permit 20**	Create route-map with trailing allow statement to advertise non-matching routes via route-leaking.

Configuring Filter for Permitted Prefixes from VRF Default on the Border Node

This procedure applies equally to IPv6.

SUMMARY STEPS

1. **configure terminal**
2. **route-map** *name* **permit 10**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**route-map** *name* **permit 10**	Create route-map with allow statement to advertise routes via route-leaking to the customer VRF and subsequently remote VTEPs.

Configuring Custom VRF Context on the Border Node - 2

This procedure applies equally to IPv6.

SUMMARY STEPS

1. **configure terminal**
2. **vrf context** *vrf-name*
3. **vni** *number*
4. **rd auto**
5. **ip route 0.0.0.0/0 Null0**
6. **address-family ipv4 unicast**
7. **route-target both** {**auto** | *rt*}
8. **route-target both** {**auto** | *rt*} **evpn**
9. **route-target both** *shared-vrf-rt*
10. **route-target both** *shared-vrf-rt* **evpn**
11. **import vrf default map** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**vrf context** *vrf-name*	Configure the VRF.

	Command or Action	Purpose
Step 3	**vni** *number*	Specify the VNI. The VNI associated with the VRF is often referred to as Layer 3 VNI, L3VNI, or L3VPN. The L3VNI is configured as the common identifier across the participating VTEPs.
Step 4	**rd auto**	Specify the VRF's route distinguisher (RD). The RD uniquely identifies a VTEP within an L3VNI.
Step 5	**ip route 0.0.0.0/0 Null0**	Configure default-route in common VRF to attract traffic towards Border Node with Shared Internet VRF.
Step 6	**address-family ipv4 unicast**	Configure the IPv4 address family. This configuration is required for IPv4 over VXLAN with IPv4 underlay.
Step 7	**route-target both** {**auto** | *rt*}	Configure the route target (RT) for the import and export of EVPN and IPv4 prefixes within the IPv4 address family. If you enter an RT, the following formats are supported: ASN2:NN, ASN4:NN, or IPV4:NN. Manually configured RTs are required to support asymmetric VNIs.
Step 8	**route-target both** {**auto** | *rt*} **evpn**	Configure the route target (RT) for the import and export of EVPN and IPv4 prefixes within the IPv4 address family. If you enter an RT, the following formats are supported: ASN2:NN, ASN4:NN, or IPV4:NN. Manually configured RTs are required to support asymmetric VNIs.
Step 9	**route-target both** *shared-vrf-rt*	Configure a special route target (RT) for the import/export of the shared IPv4 prefixes. An additional import/export map for further qualification is supported.
Step 10	**route-target both** *shared-vrf-rt* **evpn**	Configure a special route target (RT) for the import/export of the shared IPv4 prefixes. An additional import/export map for further qualification is supported.
Step 11	**import vrf default map** *name*	Permits all routes, from VRF default, from being imported into the custom VRF according to the specific route-map.

Configuring Custom VRF Instance in BGP on the Border Node

This procedure applies equally to IPv6.

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *autonomous-system-number*
3. **vrf** *vrf-name*
4. **address-family ipv4 unicast**
5. **advertise l2vpn evpn**
6. **network 0.0.0.0/0**
7. **maximum-paths ibgp** *number*

8. **maximum-paths** *number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**router bgp** *autonomous-system-number*	Configure BGP.
Step 3	**vrf** *vrf-name*	Specify the VRF.
Step 4	**address-family ipv4 unicast**	Configure address family for IPv4.
Step 5	**advertise l2vpn evpn**	Enable the advertisement of EVPN routes within IPv4 address-family.
Step 6	**network 0.0.0.0/0**	Creating IPv4 default-route network statement.
Step 7	**maximum-paths ibgp** *number*	Enabling equal cost multipathing (ECMP) for iBGP prefixes.
Step 8	**maximum-paths** *number*	Enabling equal cost multipathing (ECMP) for eBGP prefixes.

Example - Configuration Centralized VRF Route-Leaking - VRF Default with Custom VRF

An example of Centralized VRF route-leaking with VRF default

Configuring VXLAN BGP EVPN Border Node for VRF Default

The VXLAN BGP EVPN Border Node provides centralized access to VRF default. The leaking configuration is localized such that control-plane leaking and data-path forwarding following the same path. Most significantly is the VRF configuration of the Border Node and the advertisement of the default-route and less specific prefixes (aggregates) into the respective destination VRFs.

```
ip route 0.0.0.0/0 10.9.9.1
!
ip prefix-list PL_DENY_EXPORT seq 5 permit 0.0.0.0/0
!
route-map permit 10
match ip address prefix-list PL_DENY_EXPORT
route-map RM_DENY_EXPORT permit 20
route-map RM_PERMIT_IMPORT permit 10
!
vrf context Blue
  vni 51010
  ip route 0.0.0.0/0 Null0
  rd auto
  address-family ipv4 unicast
    route-target both auto
    route-target both auto evpn
    import vrf default map RM_PERMIT_IMPORT
    export vrf default 100 map RM_DENY_EXPORT allow-vpn
!
vlan 2110
  vn-segment 51010
!
```

```
interface Vlan2110
  no shutdown
  mtu 9216
  vrf member Blue
  no ip redirects
  ip forward
!
vrf context Red
  vni 51020
  ip route 0.0.0.0/0 Null0
  rd auto
  address-family ipv4 unicast
    route-target both auto
    route-target both auto evpn
    import vrf default map RM_PERMIT_IMPORT
    export vrf default 100 map RM_DENY_EXPORT allow-vpn
!
vlan 2120
  vn-segment 51020
!
interface Vlan2120
  no shutdown
  mtu 9216
  vrf member Blue
  no ip redirects
  ip forward
!
interface nve1
  no shutdown
  host-reachability protocol bgp
  source-interface loopback1
  member vni 51010 associate-vrf
  member vni 51020 associate-vrf
!
router bgp 65002
  address-family ipv4 unicast
      aggregate-address 10.10.0.0/16
      aggregate-address 10.20.0.0/16
      maximum-paths 2
      maximum-paths ibgp 2
  vrf Blue
    address-family ipv4 unicast
      advertise l2vpn evpn
      network 0.0.0.0/0
      maximum-paths ibgp 2
      maximum-paths 2
  vrf Red
    address-family ipv4 unicast
      advertise l2vpn evpn
      network 0.0.0.0/0
      maximum-paths ibgp 2
      maximum-paths 2
```

CHAPTER 8

Configuring BGP EVPN Filtering

This chapter contains the following sections:

About BGP EVPN Filtering

This feature describes the requirements for route filtering and attributes handling, arising from the implementation of BGP NLRIs of address family L2VPN EVPN.

EVPN routes are quite different from regular IPv4 and IPv6 routes in NLRI format. They contain many fields and carry attributes specific to EVPN. Using route maps, we can filter routes on the basis of these attributes. The following route-filtering options are available for the routes belonging to the EVPN address family:

- Matching based on the EVPN route type: Six types of NLRIs are available in EVPN. Matching is based on the type specified in the route-map match statement.
- Matching based on the MAC address in the NLRI: This option is similar to matching based on the IP address embedded in the NLRI. EVPN type-2 routes contain a MAC address along with an IP address. This option can be used to filter such routes.
- Matching based on the RMAC extended community: EVPN type-2 and type-5 routes carry the router MAC (RMAC) extended community, which carries a MAC address. The RMAC is advertised as part of the update message to the neighbor along with other extended community information. It specifies the MAC address of the remote next hop of a route. This option allows matching against this RMAC extended community.
- Setting the RMAC extended community: This option allows you to change the RMAC extended community value of an EVPN NLRI.
- Setting the EVPN next-hop IP address: This option sets the next-hop IP address of the EVPN route once the match condition has been met. Setting the next-hop IP address for EVPN routes should be accompanied by setting the RMAC extended community to ensure correctness in forwarding.
- Setting the gateway IP address for route type-5: The gateway IP address encodes an overlay IP index for the IP prefixes that form the type-5 EVPN routes. It gets advertised as part of the EVPN NLRI in the

update message. The default value is 0.0.0.0. When it's set to any other value, the next hop on the route in the VRF context changes to the gateway IP address specified.

- Using table maps: You can configure table maps to filter MAC routes downloaded to the Layer 2 Routing Information Base (L2RIB).

The rest of this chapter provides information on configuring and applying these options.

Guidelines and Limitations for BGP EVPN Filtering

The following are the guidelines and limitations for BGP EVPN filtering:

Cisco Nexus 9000 Series switches support BGP EVPN filtering.

The following match and set options are available for filtering an EVPN address family of routes:

- Matching based on the route type
- Matching based on the MAC address in the NLRI
- Matching based on the RMAC extended community
- Setting the RMAC extended community
- Setting the EVPN next-hop IP address—If more than one next-hop IP address is configured, only the first one is used and processed if using for EVPN. IPv4 and IPv6 can be used as next-hop addresses.
- Setting the gateway IP address for a route type-5—You can set an IPv4 gateway IP address using the **route-map** command.
- Using table maps—A table map for filtering MAC routes is downloaded to the Layer 2 Routing Information Base (L2RIB).

Configuring BGP EVPN Filtering

To perform route filtering for the EVPN address-family routes, you can perform the following tasks:

- Configuring the Route Map with Match and Set Clauses, on page 184
- Applying the Route Map at the Inbound or Outbound Level, on page 188

To configure the table map, you can perform the following tasks:

- Configuring a MAC List and a Route Map that Matches the MAC List, on page 198
- Applying the Table Map, on page 199

Configuring the Route Map with Match and Set Clauses

You can use the existing route-map configuration along with the match and set clauses to decide the kind of filtering that you need.

- Matching Based on EVPN Route Type, on page 185

Matching Based on EVPN Route Type

SUMMARY STEPS

1. **configure terminal**
2. **route-map** *route-map-name*
3. **match evpn route-type {1 | 2 | 2-mac-ip | 2-mac-only | 3 | 4 | 5 | 6}**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enter global configuration mode.
Step 2	**route-map** *route-map-name* **Example:** `switch(config)# route-map ROUTE_MAP_1`	Create a route map.
Step 3	**match evpn route-type {1 | 2 | 2-mac-ip | 2-mac-only | 3 | 4 | 5 | 6}** **Example:** `switch(config-route-map)# match evpn route-type 6`	Match BGP EVPN routes.

Matching Based on MAC Address in the NLRI

SUMMARY STEPS

1. **configure terminal**
2. **mac-list** *list-name* **[seq** *seq-number***] {deny | permit}** *mac-address* **[mac-mask]**
3. **route-map** *route-map-name*
4. **match mac-list** *mac-list-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enter global configuration mode.
Step 2	**mac-list** *list-name* **[seq** *seq-number***] {deny | permit}** *mac-address* **[mac-mask]** **Example:** `switch(config)# mac-list MAC_LIST_1 permit E:E:E`	Build a MAC list.
Step 3	**route-map** *route-map-name* **Example:** `switch(config)# route-map ROUTE_MAP_1`	Create a route map.
Step 4	**match mac-list** *mac-list-name* **Example:** `switch(config-route-map)# match mac-list MAC_LIST_1`	Match entries of MAC lists. The maximum length is 63 characters.

Matching Based on RMAC Extended Community

SUMMARY STEPS

1. **configure terminal**
2. **ip extcommunity-list standard** *list-name* **{deny | permit} rmac** *mac-addr*
3. **route-map** *route-map-name*
4. **match extcommunity** *list-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enter global configuration mode.
Step 2	**ip extcommunity-list standard** *list-name* **{deny | permit} rmac** *mac-addr* **Example:** `switch(config)# ip extcommunity-list standard EXTCOMM_LIST_RMAC permit rmac a8b4.56e4.7edf`	Add an extcommunity list entry. The *list-name* argument must not exceed 63 characters.
Step 3	**route-map** *route-map-name* **Example:** `switch(config)# route-map ROUTE_MAP_1`	Create a route map.

	Command or Action	Purpose
Step 4	**match extcommunity** *list-name* **Example:** `switch(config-route-map)# match extcommunity EXTCOMM_LIST_RMAC`	Match the extended community list name.

Setting the RMAC Extended Community

SUMMARY STEPS

1. **configure terminal**
2. **route-map** *route-map-name*
3. **set extcommunity evpn rmac** *mac-address*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enter global configuration mode.
Step 2	**route-map** *route-map-name* **Example:** `switch(config)# route-map ROUTE_MAP_1`	Create a route map.
Step 3	**set extcommunity evpn rmac** *mac-address* **Example:** `switch(config-route-map)# set extcommunity evpn rmac EEEE.EEEE.EEEE`	Set the BGP RMAC extcommunity attribute.

Setting the EVPN Next-Hop IP Address

SUMMARY STEPS

1. **configure terminal**
2. **route-map** *route-map-name*
3. **set ip next-hop** *next-hop*
4. **set ipv6 next-hop** *next-hop*

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**route-map** *route-map-name* **Example:** `switch(config)# route-map ROUTE_MAP_1`	Create a route map.
Step 3	**set ip next-hop** *next-hop* **Example:** `switch(config-route-map)# set ip next-hop 209.165.200.226`	Set the IP address of the EVPN IP next hop.
Step 4	**set ipv6 next-hop** *next-hop* **Example:** `switch(config-route-map)# set ipv6 next-hop 2001:0DB8::1`	Set the IPv6 next-hop address.

Setting the Gateway IP Address for Route Type-5

SUMMARY STEPS

1. **configure terminal**
2. **route-map** *route-map-name*
3. **set evpn gateway-ip** *gw-ip-address*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enter global configuration mode.
Step 2	**route-map** *route-map-name* **Example:** `switch(config)# route-map ROUTE_MAP_1`	Create a route map.
Step 3	**set evpn gateway-ip** *gw-ip-address* **Example:** `switch(config-route-map)# set evpn gateway-ip 209.165.200.227`	Set the gateway IP address.

Applying the Route Map at the Inbound or Outbound Level

Once you've configured the route map with match and set clauses based on your requirements, use this procedure to apply the route map at the inbound or outbound level.

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *as-num*
3. **neighbor** *address*
4. **address-family l2vpn evpn**
5. **route-map** *route-map* **{in | out}**

DETAILED STEPS

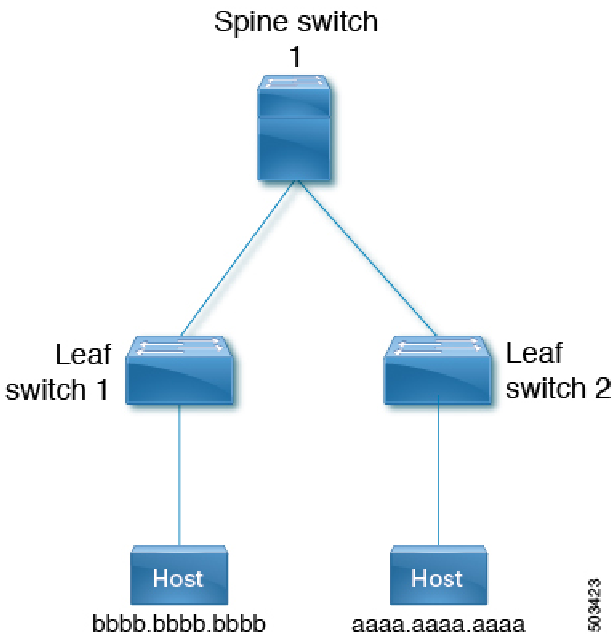
	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enter global configuration mode.
Step 2	**router bgp** *as-num* **Example:** `switch(config)# router bgp 100`	Enables a routing process. The range of *as-num* is from 1 to 65535.
Step 3	**neighbor** *address* **Example:** `switch(config-router)# neighbor 1.1.1.1`	Configure a BGP neighbor.
Step 4	**address-family l2vpn evpn** **Example:** `switch(config-router-neighbor)# address-family l2vpn evpn`	Configure the L2VPN address family.
Step 5	**route-map** *route-map* **{in | out}** **Example:** `switch(config-router-neighbor-af)# route-map ROUTE_MAP_1 in`	Apply the route map to the neighbor.

BGP EVPN Filtering Configuration Examples

This section provides example configurations for filtering EVPN routes.

Example 1

The following example shows how to filter EVPN type-2 routes and set the RMAC extended community as 52fc.c310.2e80.



1. The following output shows the routes in the EVPN table and a type-2 EVPN MAC route before the route map is applied.

```
leaf1(config)# show bgp l2vpn evpn
BGP routing table information for VRF default, address family L2VPN EVPN
BGP table version is 12, Local Router ID is 1.1.1.1
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-injected
Origin codes: i - IGP, e - EGP, ? - incomplete, | - multipath, & - backup, 2 - best2

Network            Next Hop            Metric     LocPrf     Weight Path
Route Distinguisher: 1.1.1.1:32868    (L2VNI 101)
*>i[2]:[0]:[0]:[48]:[aaaa.aaaa.aaaa]:[32]:[101.0.0.3]/272
33.33.33.33                       100          0 i

Route Distinguisher: 3.3.3.3:3
*>i[2]:[0]:[0]:[48]:[52fc.d83a.1b08]:[0]:[0.0.0.0]/216
33.33.33.33                       100          0 i
*>i[5]:[0]:[0]:[24]:[101.0.0.0]/224
3.3.3.3                  0        100          0 ?

Route Distinguisher: 3.3.3.3:32868
*>i[2]:[0]:[0]:[48]:[aaaa.aaaa.aaaa]:[32]:[101.0.0.3]/272
33.33.33.33                       100          0 i

Route Distinguisher: 1.1.1.1:3    (L3VNI 100)
*>i[2]:[0]:[0]:[48]:[52fc.d83a.1b08]:[0]:[0.0.0.0]/216
33.33.33.33                       100          0 i
*>i[2]:[0]:[0]:[48]:[aaaa.aaaa.aaaa]:[32]:[101.0.0.3]/272
33.33.33.33                       100          0 i
*>l[5]:[0]:[0]:[24]:[10.0.0.0]/224
1.1.1.1                  0        100      32768 ?
*>l[5]:[0]:[0]:[24]:[100.0.0.0]/224
1.1.1.1                  0        100      32768 ?
*>i[5]:[0]:[0]:[24]:[101.0.0.0]/224
3.3.3.3                  0        100          0 ?

leaf1(config)# show bgp l2vpn evpn aaaa.aaaa.aaaa
```

```
BGP routing table information for VRF default, address family L2VPN EVPN
Route Distinguisher: 1.1.1.1:32868    (L2VNI 101)
BGP routing table entry for [2]:[0]:[0]:[48]:[aaaa.aaaa.aaaa]:[32]:[101.0.0.3]/2
72, version 12
Paths: (1 available, best #1)
Flags: (0x000212) (high32 00000000) 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, in rib
Imported from 3.3.3.3:32868:[2]:[0]:[0]:[48]:[aaaa.aaaa.aaaa]:[32]:
[101.0.0.3]/272
AS-Path: NONE, path sourced internal to AS
33.33.33.33 (metric 81) from 101.101.101.101 (101.101.101.101)
Origin IGP, MED not set, localpref 100, weight 0
Received label 101 100
Extcommunity: RT:100:100 RT:100:101 SOO:33.33.33.33:0 ENCAP:8
Router MAC:52fc.d83a.1b08
Originator: 3.3.3.3 Cluster list: 101.101.101.101

Path-id 1 not advertised to any peer

Route Distinguisher: 3.3.3.3:32868
BGP routing table entry for [2]:[0]:[0]:[48]:[aaaa.aaaa.aaaa]:[32]:[101.0.0.3]/2
72, version 8
Paths: (1 available, best #1)
Flags: (0x000202) (high32 00000000) on xmit-list, is not in l2rib/evpn, is not in HW

Advertised path-id 1
Path type: internal, path is valid, is best path, no labeled nexthop
Imported to 3 destination(s)
Imported paths list: vni100 default default
AS-Path: NONE, path sourced internal to AS
33.33.33.33 (metric 81) from 101.101.101.101 (101.101.101.101)
Origin IGP, MED not set, localpref 100, weight 0
Received label 101 100
Extcommunity: RT:100:100 RT:100:101 SOO:33.33.33.33:0 ENCAP:8
Router MAC:52fc.d83a.1b08
Originator: 3.3.3.3 Cluster list: 101.101.101.101

Path-id 1 not advertised to any peer

Route Distinguisher: 1.1.1.1:3    (L3VNI 100)
BGP routing table entry for [2]:[0]:[0]:[48]:[aaaa.aaaa.aaaa]:[32]:[101.0.0.3]/2
72, version 11
Paths: (1 available, best #1)
Flags: (0x000202) (high32 00000000) on xmit-list, is not 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 3.3.3.3:32868:[2]:[0]:[0]:[48]:[aaaa.aaaa.aaaa]:[32]:
[101.0.0.3]/272
AS-Path: NONE, path sourced internal to AS
33.33.33.33 (metric 81) from 101.101.101.101 (101.101.101.101)
Origin IGP, MED not set, localpref 100, weight 0
Received label 101 100
Extcommunity: RT:100:100 RT:100:101 SOO:33.33.33.33:0 ENCAP:8
Router MAC:52fc.d83a.1b08
Originator: 3.3.3.3 Cluster list: 101.101.101.101

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

2. The following example shows the route-map configuration.

```
leaf1(config)# show run rpm

!Command: show running-config rpm
!Running configuration last done at: Thu Sep  3 22:32:23 2020
!Time: Thu Sep  3 22:32:31 2020

version 9.3(5) Bios:version
route-map FILTER_EVPN_TYPE2 permit 10
    match evpn route-type 2
    set extcommunity evpn rmac 52fc.c310.2e80
route-map allow permit 10
```

3. The following example shows how to apply the route map to the EVPN peer as an inbound route map.

```
leaf1(config-router-neighbor-af)# show run bgp

!Command: show running-config bgp
!Running configuration last done at: Mon Aug  3 18:08:24 2020
!Time: Mon Aug  3 18:08:28 2020

version 9.3(5) Bios:version
feature bgp

router bgp 100
  event-history detail size large
  neighbor 101.101.101.101
    remote-as 100
    update-source loopback0
    address-family l2vpn evpn
      send-community extended
      route-map FILTER_EVPN_TYPE2 in
  vrf vni100
    address-family ipv4 unicast
      advertise l2vpn evpn
      redistribute direct route-map allow
```

4. The following output shows the routes in the EVPN table and a type-2 EVPN MAC route after the route map is applied.

```
leaf1(config)# show bgp l2vpn evpn
BGP routing table information for VRF default, address family L2VPN EVPN
BGP table version is 19, Local Router ID is 1.1.1.1
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-injected
Origin codes: i - IGP, e - EGP, ? - incomplete, | - multipath, & - backup, 2 - best2

Network            Next Hop            Metric     LocPrf     Weight Path
Route Distinguisher: 1.1.1.1:32868    (L2VNI 101)
*>i[2]:[0]:[0]:[48]:[aaaa.aaaa.aaaa]:[32]:[101.0.0.3]/272
                      33.33.33.33                      100          0 i

Route Distinguisher: 3.3.3.3:3
*>i[2]:[0]:[0]:[48]:[52fc.d83a.1b08]:[0]:[0.0.0.0]/216
                      33.33.33.33                      100          0 i

Route Distinguisher: 3.3.3.3:32868
*>i[2]:[0]:[0]:[48]:[aaaa.aaaa.aaaa]:[32]:[101.0.0.3]/272
                      33.33.33.33                      100          0 i

Route Distinguisher: 1.1.1.1:3    (L3VNI 100)
*>i[2]:[0]:[0]:[48]:[52fc.d83a.1b08]:[0]:[0.0.0.0]/216
                      33.33.33.33                      100          0 i
*>i[2]:[0]:[0]:[48]:[aaaa.aaaa.aaaa]:[32]:[101.0.0.3]/272
```

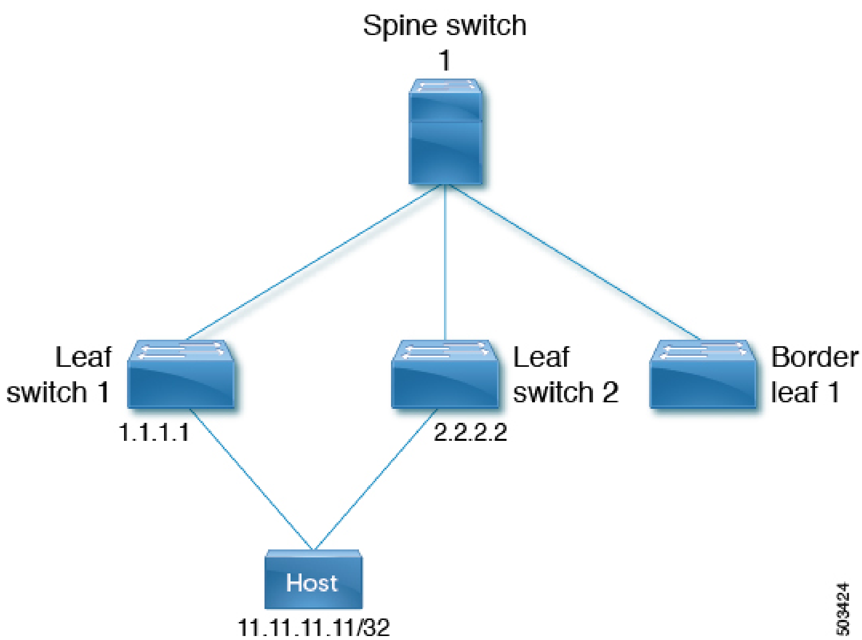
```
                    33.33.33.33                     100          0 i
*>l[5]:[0]:[0]:[24]:[10.0.0.0]/224
                    1.1.1.1              0          100      32768 ?
*>l[5]:[0]:[0]:[24]:[100.0.0.0]/224
                    1.1.1.1              0          100      32768 ?

leaf1(config)# show bgp l2vpn evpn aaaa.aaaa.aaaa
BGP routing table information for VRF default, address family L2VPN EVPN
Route Distinguisher: 1.1.1.1:32868    (L2VNI 101)
BGP routing table entry for [2]:[0]:[0]:[48]:[aaaa.aaaa.aaaa]:[32]:[101.0.0.3]/2
72, version 19
Paths: (1 available, best #1)
Flags: (0x000212) (high32 00000000) 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, in rib
Imported from 3.3.3.3:32868:[2]:[0]:[0]:[48]:[aaaa.aaaa.aaaa]:[32]:
[101.0.0.3]/272
AS-Path: NONE, path sourced internal to AS
33.33.33.33 (metric 81) from 101.101.101.101 (101.101.101.101)
Origin IGP, MED not set, localpref 100, weight 0
Received label 101 100
Extcommunity: RT:100:100 RT:100:101 SOO:33.33.33.33:0 ENCAP:8
Router MAC:52fc.c310.2e80
Originator: 3.3.3.3 Cluster list: 101.101.101.101
Path-id 1 not advertised to any peer

Route Distinguisher: 3.3.3.3:32868
BGP routing table entry for [2]:[0]:[0]:[48]:[aaaa.aaaa.aaaa]:[32]:[101.0.0.3]/2
72, version 15
Paths: (1 available, best #1)
Flags: (0x000202) (high32 00000000) on xmit-list, is not in l2rib/evpn, is not in HW

Advertised path-id 1
Path type: internal, path is valid, is best path, no labeled nexthop
Imported to 3 destination(s)
Imported paths list: vni100 default default
AS-Path: NONE, path sourced internal to AS
33.33.33.33 (metric 81) from 101.101.101.101 (101.101.101.101)
Origin IGP, MED not set, localpref 100, weight 0
Received label 101 100
Extcommunity: RT:100:100 RT:100:101 SOO:33.33.33.33:0 ENCAP:8
Router MAC:52fc.c310.2e80
Originator: 3.3.3.3 Cluster list: 101.101.101.101

Path-id 1 not advertised to any peer

Route Distinguisher: 1.1.1.1:3    (L3VNI 100)
BGP routing table entry for [2]:[0]:[0]:[48]:[aaaa.aaaa.aaaa]:[32]:[101.0.0.3]/2
72, version 18
Paths: (1 available, best #1)
Flags: (0x000202) (high32 00000000) on xmit-list, is not 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 3.3.3.3:32868:[2]:[0]:[0]:[48]:[aaaa.aaaa.aaaa]:[32]:
[101.0.0.3]/272
AS-Path: NONE, path sourced internal to AS
33.33.33.33 (metric 81) from 101.101.101.101 (101.101.101.101)
Origin IGP, MED not set, localpref 100, weight 0
Received label 101 100
Extcommunity: RT:100:100 RT:100:101 SOO:33.33.33.33:0 ENCAP:8
Router MAC:52fc.c310.2e80
Originator: 3.3.3.3 Cluster list: 101.101.101.101
```

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

In a similar manner, you can use the other EVPN-specific match and set clauses with existing route-map options to filter EVPN routes as required.

Example 2

The following example shows how EVPN route filtering can be used to redirect traffic to a different VTEP than the one from which the EVPN route was learned. It involves setting the next-hop IP address and the RMAC of the route to the one corresponding to the other VTEP.



This example demonstrates the following:

- Host 1 belongs to VRF evpn-tenant-0002 and VLAN 3002, and is connected to Leaf 1 and Leaf 2.
- Reachability to Host1 is advertised by Leaf 1 and Leaf 2 to BL1.

At BL1, both routes to 11.11.11.11/32 are received as follows:

- One from 1.1.1.1, which is Leaf 1
- One from 2.2.2.2, which is Leaf 2

1. Initially the best path to reach 11.11.11.11 is through 1.1.1.1:

```
bl1(config)# show bgp l2 e 11.11.11.11
BGP routing table information for VRF default, address family L2VPN EVPN
Route Distinguisher: 1.1.1.1:3
BGP routing table entry for [5]:[0]:[0]:[32]:[11.11.11.11]/224, version 15
Paths: (1 available, best #1)
Flags: (0x000002) (high32 00000000) on xmit-list, is not in l2rib/evpn, is not in HW

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

```
    Imported paths list: evpn-tenant-0002 default
Gateway IP: 0.0.0.0
AS-Path: 150 , path sourced external to AS
  1.1.1.1 (metric 81) from 101.101.101.101 (101.101.101.101)
    Origin incomplete, MED 0, localpref 100, weight 0
    Received label 3003002
    Extcommunity: RT:1:3003002 ENCAP:8 Router MAC:5254.0074.caf5
    Originator: 1.1.1.1 Cluster list: 101.101.101.101

Path-id 1 not advertised to any peer

Route Distinguisher: 2.2.2.2:4
BGP routing table entry for [5]:[0]:[0]:[32]:[11.11.11.11]/224, version 79
Paths: (1 available, best #1)
Flags: (0x000002) (high32 00000000) on xmit-list, is not in l2rib/evpn, is not in HW

Advertised path-id 1
Path type: internal, path is valid, is best path, no labeled nexthop
    Imported to 2 destination(s)
    Imported paths list: evpn-tenant-0002 default
Gateway IP: 0.0.0.0
AS-Path: 150 , path sourced external to AS
  2.2.2.2 (metric 81) from 101.101.101.101 (101.101.101.101)
    Origin incomplete, MED 0, localpref 100, weight 0
    Received label 3003002
    Extcommunity: RT:1:3003002 ENCAP:8 Router MAC:5254.0090.433e
    Originator: 2.2.2.2 Cluster list: 101.101.101.101

Path-id 1 not advertised to any peer

Route Distinguisher: 3.3.3.3:3    (L3VNI 3003002)
BGP routing table entry for [5]:[0]:[0]:[32]:[11.11.11.11]/224, version 80
Paths: (2 available, best #2)Flags: (0x000002) (high32 00000000) on xmit-list, is not
in l2rib/evpn, is not in HW

Path type: internal, path is valid, not best reason: Router Id, no labeled nexthop
    Imported from 2.2.2.2:4:[5]:[0]:[0]:[32]:[11.11.11.11]/224
Gateway IP: 0.0.0.0
AS-Path: 150 , path sourced external to AS
  2.2.2.2 (metric 81) from 101.101.101.101 (101.101.101.101)
    Origin incomplete, MED 0, localpref 100, weight 0
    Received label 3003002
    Extcommunity: RT:1:3003002 ENCAP:8 Router MAC:5254.0090.433e
    Originator: 2.2.2.2 Cluster list: 101.101.101.101

Advertised path-id 1
Path type: internal, path is valid, is best path, no labeled nexthop
    Imported from 1.1.1.1:3:[5]:[0]:[0]:[32]:[11.11.11.11]/224
Gateway IP: 0.0.0.0
AS-Path: 150 , path sourced external to AS
  1.1.1.1 (metric 81) from 101.101.101.101 (101.101.101.101)
    Origin incomplete, MED 0, localpref 100, weight 0
    Received label 3003002
    Extcommunity: RT:1:3003002 ENCAP:8 Router MAC:5254.0074.caf5
    Originator: 1.1.1.1 Cluster list: 101.101.101.101

Path-id 1 not advertised to any peer

Route Distinguisher: 3.3.3.3:4    (L3VNI 3003003)
BGP routing table entry for [5]:[0]:[0]:[32]:[11.11.11.11]/224, version 24
Paths: (1 available, best #1)
Flags: (0x000002) (high32 00000000) on xmit-list, is not in l2rib/evpn

Advertised path-id 1
```

```
Path type: local, path is valid, is best path, no labeled nexthop
Gateway IP: 0.0.0.0
AS-Path: 150 , path sourced external to AS
  3.3.3.3 (metric 0) from 0.0.0.0 (3.3.3.3)
    Origin incomplete, MED 0, localpref 100, weight 0
    Received label 3003003
    Extcommunity: RT:1:3003003 ENCAP:8 Router MAC:5254.006a.435b
    Originator: 1.1.1.1 Cluster list: 101.101.101.101

Path-id 1 advertised to peers:
101.101.101.101

bl1(config)# show ip route 11.11.11.11
IP Route Table for VRF "default"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

11.11.11.11/32, ubest/mbest: 1/0
*via 1.1.1.1, [200/0], 00:02:51, bgp-1, internal, tag 150 (evpn) segid: 3003
002 tunnelid: 0x1010101 encap: VXLAN
```

2. To redirect traffic to the other VTEP leaf-2, you can set the next hop and RMAC on the 11.11.11.11/32 route with a route-map configuration.

```
bl1(config-route-map)# show run rpm

Command: show running-config rpm
!Running configuration last done at: Wed Mar 27 00:12:14 2019
!Time: Wed Mar 27 00:12:17 2019

version 9.2(3) Bios:version
ip prefix-list PFX_LIST1_1 seq 5 permit 11.11.11.11/32
route-map TEST_SET_IP_NEXTHOP permit 10
    match ip address prefix-list PFX_LIST1_1
    set ip next-hop 2.2.2.2
    set extcommunity evpn rmac 5254.0090.433e
```

3. After applying the route map at the inbound level at BL1, the following are the route outputs for route 11.11.11.11/32.

```
bl1(config-router-neighbor-af)# show bgp l2 e 11.11.11.11
BGP routing table information for VRF default, address family L2VPN EVPN
Route Distinguisher: 1.1.1.1:3
BGP routing table entry for [5]:[0]:[0]:[32]:[11.11.11.11]/224, version 81
Paths: (1 available, best #1)
Flags: (0x000002) (high32 00000000) on xmit-list, is not in l2rib/evpn, is not in HW

Advertised path-id 1
Path type: internal, path is valid, is best path, no labeled nexthop
    Imported to 2 destination(s)
    Imported paths list: evpn-tenant-0002 default
Gateway IP: 0.0.0.0
AS-Path: 150 , path sourced external to AS
  2.2.2.2 (metric 81) from 101.101.101.101 (101.101.101.101)
    Origin incomplete, MED 0, localpref 100, weight 0
    Received label 3003002
    Extcommunity: RT:1:3003002 ENCAP:8 Router MAC:5254.0090.433e
    Originator: 1.1.1.1 Cluster list: 101.101.101.101

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

```
Route Distinguisher: 2.2.2.2:4
BGP routing table entry for [5]:[0]:[0]:[32]:[11.11.11.11]/224, version 79
Paths: (1 available, best #1)
Flags: (0x000002) (high32 00000000) on xmit-list, is not in l2rib/evpn, is not in HW

Advertised path-id 1
Path type: internal, path is valid, is best path, no labeled nexthop
    Imported to 2 destination(s)
    Imported paths list: evpn-tenant-0002 default
Gateway IP: 0.0.0.0
AS-Path: 150 , path sourced external to AS
  2.2.2.2 (metric 81) from 101.101.101.101 (101.101.101.101)
    Origin incomplete, MED 0, localpref 100, weight 0
    Received label 3003002
    Extcommunity: RT:1:3003002 ENCAP:8 Router MAC:5254.0090.433e
    Originator: 2.2.2.2 Cluster list: 101.101.101.101

Path-id 1 not advertised to any peer

Route Distinguisher: 3.3.3.3:3    (L3VNI 3003002)
BGP routing table entry for [5]:[0]:[0]:[32]:[11.11.11.11]/224, version 82
Paths: (2 available, best #2)
Flags: (0x000002) (high32 00000000) on xmit-list, is not in l2rib/evpn, is not in HW

Path type: internal, path is valid, not best reason: Router Id, no labeled nexthop
    Imported from 2.2.2.2:4:[5]:[0]:[0]:[32]:[11.11.11.11]/224
Gateway IP: 0.0.0.0
AS-Path: 150 , path sourced external to AS
  2.2.2.2 (metric 81) from 101.101.101.101 (101.101.101.101)
    Origin incomplete, MED 0, localpref 100, weight 0
    Received label 3003002
    Extcommunity: RT:1:3003002 ENCAP:8 Router MAC:5254.0090.433e
    Originator: 2.2.2.2 Cluster list: 101.101.101.101

Advertised path-id 1
Path type: internal, path is valid, is best path, no labeled nexthop
    Imported from 1.1.1.1:3:[5]:[0]:[0]:[32]:[11.11.11.11]/224
Gateway IP: 0.0.0.0
AS-Path: 150 , path sourced external to AS
  2.2.2.2 (metric 81) from 101.101.101.101 (101.101.101.101)
    Origin incomplete, MED 0, localpref 100, weight 0
    Received label 3003002
    Extcommunity: RT:1:3003002 ENCAP:8 Router MAC:5254.0090.433e
    Originator: 1.1.1.1 Cluster list: 101.101.101.101

Path-id 1 not advertised to any peer

Route Distinguisher: 3.3.3.3:4    (L3VNI 3003003)
BGP routing table entry for [5]:[0]:[0]:[32]:[11.11.11.11]/224, version 24
Paths: (1 available, best #1)
Flags: (0x000002) (high32 00000000) on xmit-list, is not in l2rib/evpn

Advertised path-id 1
Path type: local, path is valid, is best path, no labeled nexthop
Gateway IP: 0.0.0.0
AS-Path: 150 , path sourced external to AS
  3.3.3.3 (metric 0) from 0.0.0.0 (3.3.3.3)
    Origin incomplete, MED 0, localpref 100, weight 0
    Received label 3003003
    Extcommunity: RT:1:3003003 ENCAP:8 Router MAC:5254.006a.435b
    Originator: 1.1.1.1 Cluster list: 101.101.101.101

Path-id 1 advertised to peers:
```

```
101.101.101.101


bl1(config-router-neighbor-af)# show ip route 11.11.11.11
IP Route Table for VRF "default"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

11.11.11.11/32, ubest/mbest: 1/0
*via 2.2.2.2, [200/0], 00:02:37, bgp-1, internal, tag 150 (evpn) segid: 3003
002 tunnelid: 0x2020202 encap: VXLAN
```

After the next hop and RMAC value are set using the route map, the traffic that was earlier directed through 1.1.1.1 is now directed through 2.2.2.2.

Configuring a Table Map

Perform these tasks to configure and apply a table map:

- Configuring a MAC List and a Route Map that Matches the MAC List, on page 198
- Applying the Table Map, on page 199

Configuring a MAC List and a Route Map that Matches the MAC List

SUMMARY STEPS

1. **configure terminal**
2. **mac-list** *list-name* **[seq** *seq-number*] **{deny | permit}** *mac-address* **[mac-mask]**
3. **route-map** *route-map-name*
4. **match mac-list** *mac-list-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enter global configuration mode.
Step 2	**mac-list** *list-name* **[seq** *seq-number*] **{deny | permit}** *mac-address* **[mac-mask]** **Example:** `switch(config)# mac-list MAC_LIST_1 permit E:E:E`	Build a MAC list.
Step 3	**route-map** *route-map-name* **Example:** `switch(config)# route-map ROUTE_MAP_1`	Create a route map.

	Command or Action	Purpose
Step 4	**match mac-list** *mac-list-name* **Example:** `switch(config-route-map)# match mac-list MAC_LIST_1`	Match entries of MAC lists. The maximum length is 63 characters.

Applying the Table Map

SUMMARY STEPS

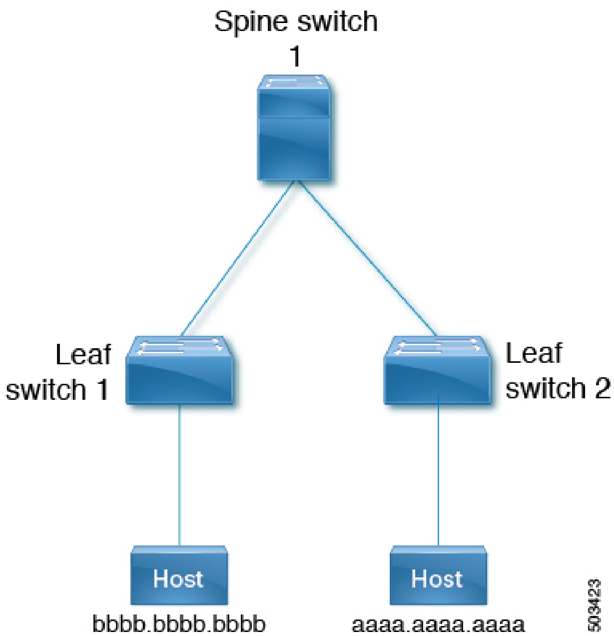
1. **configure terminal**
2. **evpn**
3. **vni** *vni-id* **l2**
4. **table-map** *route-map-name* **[filter]**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enter global configuration mode.
Step 2	**evpn** **Example:** `switch(config)# evpn`	Enter EVPN configuration mode.
Step 3	**vni** *vni-id* **l2** **Example:** `switch(config-evpn)# vni 101 l2`	Configure the Ethernet VPN ID. The range of *vni-id* is from 1 to 16777214.
Step 4	**table-map** *route-map-name* **[filter]** **Example:** `switch(config-evpn-evi)# table-map ROUTE_MAP_1 filter`	Apply table maps at the EVPN VNI configuration level. If the **filter** option is specified, any route that gets denied by the route-map validation isn't downloaded into the L2RIB.

Table Map Configuration Example

The following table-map configuration example shows how to filter MAC route aaaa.aaaa.aaaa from being downloaded into the L2RIB.



1. The following example shows the output for routes in the EVPN table and MAC routes in the L2RIB before the route map is applied.

```
leaf1(config)# show bgp l2vpn evpn
BGP routing table information for VRF default, address family L2VPN EVPN
BGP table version is 25, Local Router ID is 1.1.1.1
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-injected
Origin codes: i - IGP, e - EGP, ? - incomplete, | - multipath, & - backup, 2 - best2

Network            Next Hop            Metric     LocPrf     Weight Path
Route Distinguisher: 1.1.1.1:32868    (L2VNI 101)
*>i[2]:[0]:[0]:[48]:[aaaa.aaaa.aaaa]:[32]:[101.0.0.3]/272
33.33.33.33                        100          0 i

Route Distinguisher: 3.3.3.3:3
*>i[2]:[0]:[0]:[48]:[52fc.d83a.1b08]:[0]:[0.0.0.0]/216
33.33.33.33                        100          0 i

Route Distinguisher: 3.3.3.3:32868
*>i[2]:[0]:[0]:[48]:[aaaa.aaaa.aaaa]:[32]:[101.0.0.3]/272
33.33.33.33                        100          0 i

Route Distinguisher: 1.1.1.1:3    (L3VNI 100)
*>i[2]:[0]:[0]:[48]:[52fc.d83a.1b08]:[0]:[0.0.0.0]/216
33.33.33.33                        100          0 i
*>i[2]:[0]:[0]:[48]:[aaaa.aaaa.aaaa]:[32]:[101.0.0.3]/272
33.33.33.33                        100          0 i
*>l[5]:[0]:[0]:[24]:[10.0.0.0]/224
1.1.1.1                  0         100      32768 ?
*>l[5]:[0]:[0]:[24]:[100.0.0.0]/224
1.1.1.1                  0         100      32768 ?

leaf1(config)# 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, (Orp): Orphan

Topology  Mac Address     Prod  Flags  Seq No  Next-Hops
--------- -----------     ----  -----  ------- ----------
100       52fc.d83a.1b08  VXLAN  Rmac  0       33.33.33.33
101       aaaa.aaaa.aaaa  BGP    Spl   0       33.33.33.33 (Label: 101)

leaf1(config-evpn-evi)# show mac address-table vlan 101
Legend: * - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
age - seconds since last seen,+ - primary entry using vPC Peer-Link,
(T) - True, (F) - False, C - ControlPlane MAC, ~ - vsanVLAN     MAC Address
Type      age     Secure NTFY Ports
---------+-----------------+--------+---------+------+----+------------------
C  101     aaaa.aaaa.aaaa   dynamic  0         F      F    nve1(33.33.33.33)
G  101     521d.7cef.1b08   static   -         F      F    sup-eth1(R)
```

2. The following example shows how to configure the route map to filter MAC route aaaa.aaaa.aaaa.

```
leaf1(config)# show run rpm

!Command: show running-config rpm
!Running configuration last done at: Thu Sep  3 21:47:48 2020
!Time: Thu Sep  3 22:27:57 2020

version 9.4(1) Bios:version
mac-list FILTER_MAC_AAA seq 5 deny aaaa.aaaa.aaaa ffff.ffff.ffff
route-map TABLE_MAP_FILTER permit 10
  match mac-list FILTER_MAC_AAA
```

3. The following example shows how to apply the route map at the BGP EVPN level.

```
leaf1(config-evpn-evi)# show run bgp | section evpn
evpn
    vni 101 l2
      table-map TABLE_MAP_FILTER filter
      rd auto
      route-target import auto
      route-target export auto
      route-target both auto evpn
```

4. The following example shows the output for routes in the EVPN table and MAC routes in the L2RIB after the table map is configured.

```
leaf1(config-evpn-evi)# show bgp l2vpn evpn
BGP routing table information for VRF default, address family L2VPN EVPN
BGP table version is 26, Local Router ID is 1.1.1.1
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-injected
Origin codes: i - IGP, e - EGP, ? - incomplete, | - multipath, & - backup, 2 - best2
Network            Next Hop            Metric     LocPrf     Weight Path
Route Distinguisher: 1.1.1.1:32868    (L2VNI 101)
*>i[2]:[0]:[0]:[48]:[aaaa.aaaa.aaaa]:[32]:[101.0.0.3]/272
33.33.33.33                       100          0 i

Route Distinguisher: 3.3.3.3:3
*>i[2]:[0]:[0]:[48]:[52fc.d83a.1b08]:[0]:[0.0.0.0]/216
33.33.33.33                       100          0 i

Route Distinguisher: 3.3.3.3:32868
*>i[2]:[0]:[0]:[48]:[aaaa.aaaa.aaaa]:[32]:[101.0.0.3]/272
33.33.33.33                       100          0 i
```

```
Route Distinguisher: 1.1.1.1:3    (L3VNI 100)
*>i[2]:[0]:[0]:[48]:[52fc.d83a.1b08]:[0]:[0.0.0.0]/216
33.33.33.33                       100          0 i
*>i[2]:[0]:[0]:[48]:[aaaa.aaaa.aaaa]:[32]:[101.0.0.3]/272
33.33.33.33                       100          0 i
*>l[5]:[0]:[0]:[24]:[10.0.0.0]/224
1.1.1.1                  0        100      32768 ?
*>l[5]:[0]:[0]:[24]:[100.0.0.0]/224
1.1.1.1                  0        100      32768 ?

leaf1(config-evpn-evi)# 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, (Orp): Orphan

Topology    Mac Address    Prod   Flags  Seq No  Next-Hops
----------- -------------- ------ ------ ------- ------------
100         52fc.d83a.1b08 VXLAN  Rmac   0       33.33.33.33

leaf1(config-evpn-evi)# show mac address-table vlan 101
Legend:
* - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
age - seconds since last seen,+ - primary entry using vPC Peer-Link,
(T) - True, (F) - False, C - ControlPlane MAC, ~ - vsan
VLAN     MAC Address      Type      age     Secure NTFY Ports
---------+-----------------+--------+---------+------+----+------------------
G  101     521d.7cef.1b08   static   -         F      F    sup-eth1(R)
```

Verifying BGP EVPN Filtering

To display the status of the BGP EVPN Filtering configuration, enter the following command:

Table 3: Display BGP EVPN Filtering

Command	Purpose
show mac-list	Displays MAC Lists.
show route-map *name*	Displays information about a route map.
show running-config bgp	Displays the BGP configuration.
show running-config rpm	Displays all Route Policy Manager (RPM) information.
show bgp l2vpn evpn	Displays routes in BRIB.

Example of the **show mac-list** command:

```
switch(config)# show mac-list
mac-list list1: 5 entries
   seq 5 deny 0000.836d.f8b7 ffff.ffff.ffff
   seq 6 deny 0000.836d.f8b5 ffff.ffff.ffff
   seq 7 permit 0000.0422.6811 ffff.ffff.ffff
   seq 8 deny 0000.836d.f8b1 ffff.ffff.ffff
```

```
   seq 10 permit 0000.0000.0000 0000.0000.0000
mac-list list2: 3 entries
   seq 5 deny 0000.836e.f8b6 ffff.ffff.ffff
   seq 8 deny 0000.0421.6818 ffff.ffff.ffff
   seq 10 permit 0000.0000.0000 0000.0000.0000
mac-list list3: 2 entries
   seq 5 deny 0000.836d.f8b6 ffff.ffff.ffff
   seq 10 permit 0000.836d.f8b7 ffff.ffff.ffff
```

Example of the **show route-map** command:

```
switch# show route-map pol10
route-map pol10, permit, sequence 10
  Match clauses:
    mac-list: list2
  Set clauses:
    ip next-hop 6.6.6.1 3.3.3.10
    ipv6 next-hop 303:304::1
```

Example of the **show running-config bgp** command:

```
switch# show running-config bgp | beg "5000"
vni 5000 l2
table-map pol1 filter
rd auto
route-target import auto
route-target export auto
vni 5001 l2
rd auto
route-target import auto
route-target export auto
```

Example of the **show running-config rpm** command:

```
switch# show running-config rpm
!Running configuration last done at: Thu May 23 13:58:31 2019
!Time: Thu May 23 13:58:47 2019

version 9.3(1) Bios:version 07.65
feature pbr

mac-list list1 seq 5 permit 0001.0001.0001 ffff.ffff.ffff
mac-list mclist seq 5 permit 0001.0001.0001 ffff.ffff.ffff
route-map test permit 10
match evpn route-type 5
set evpn gateway-ip 1.1.1.2
```

Example of the **show bgp l2vpn evpn aaaa.aaaa.aaaa** command to view detailed information about EVPN route aaaa.aaaa.aaaa:

```
switch(config-evpn-evi)# show bgp l2 e aaaa.aaaa.aaaa

BGP routing table information for VRF default, address family L2VPN EVPN
Route Distinguisher: 1.1.1.1:32868 (L2VNI 101)
BGP routing table entry for [2]:[0]:[0]:[48]:[aaaa.aaaa.aaaa]:[32]:[101.0.0.3]/2
72, version 11
Paths: (1 available, best #1)
Flags: (0x000202) (high32 00000000) on xmit-list, is not in l2rib/evpn, table-ma
p filtered, is not in HW

Advertised path-id 1
Path type: internal, path is valid, is best path, remote nh not installed, no
```

```
labeled nexthop
Imported from 3.3.3.3:32868:[2]:[0]:[0]:[48]:[aaaa.aaaa.aaaa]:[32]:
[101.0.0.3]/272
AS-Path: NONE, path sourced internal to AS
33.33.33.33 (metric 81) from 101.101.101.101 (101.101.101.101)
Origin IGP, MED not set, localpref 100, weight 0
Received label 101 100
Extcommunity: RT:100:100 RT:100:101 SOO:33.33.33.33:0 ENCAP:8
Router MAC:5254.009b.4275
Originator: 3.3.3.3 Cluster list: 101.101.101.101

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

CHAPTER 9

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.

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 4: 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 20: Loopback Message

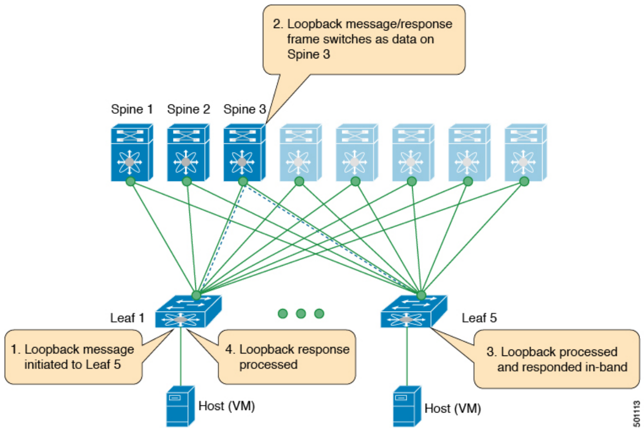
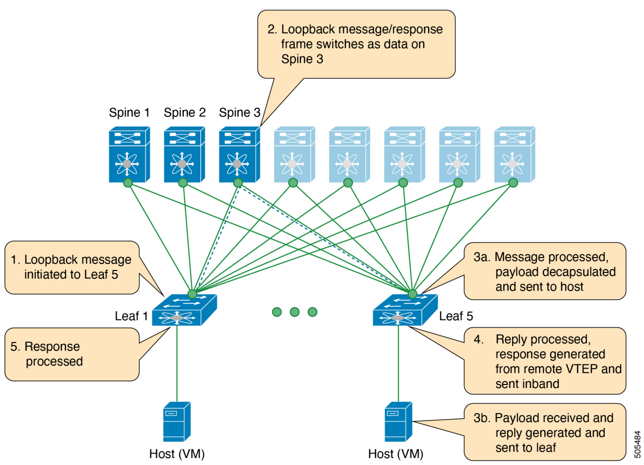


Figure 21: NVO3 Draft Tissa Ping to Remote VM



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:

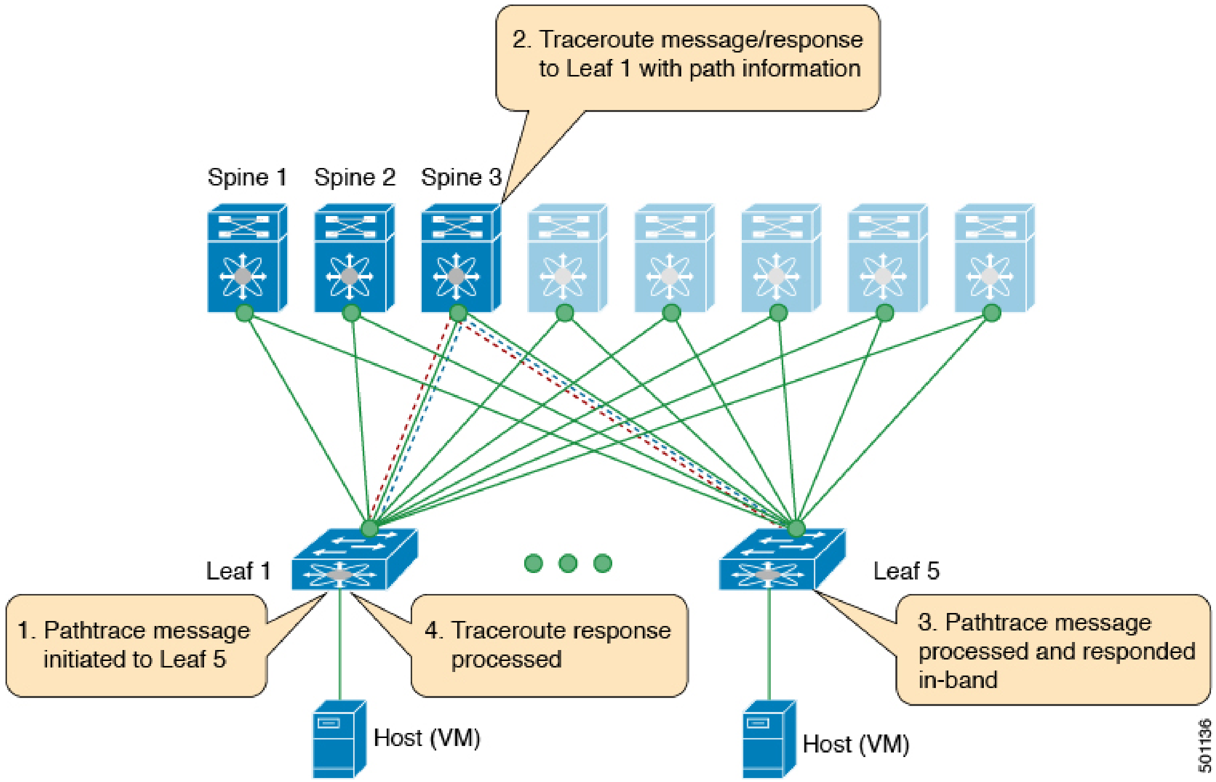
- Traceroute uses the ICMP packets (channel-1), encapsulated in the VXLAN encapsulation to reach the host

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.
- Beginning with NX-OS release 9.3(3), the `Received` field of the **show ngoam pathtrace statistics summary** command indicates all pathtrace requests received by the node on which the command is executed regardless of whether the request was destined to that node.

Figure 22: Traceroute Message



VXLAN EVPN Loop Detection and Mitigation Overview

Causes and Impacts of Loop

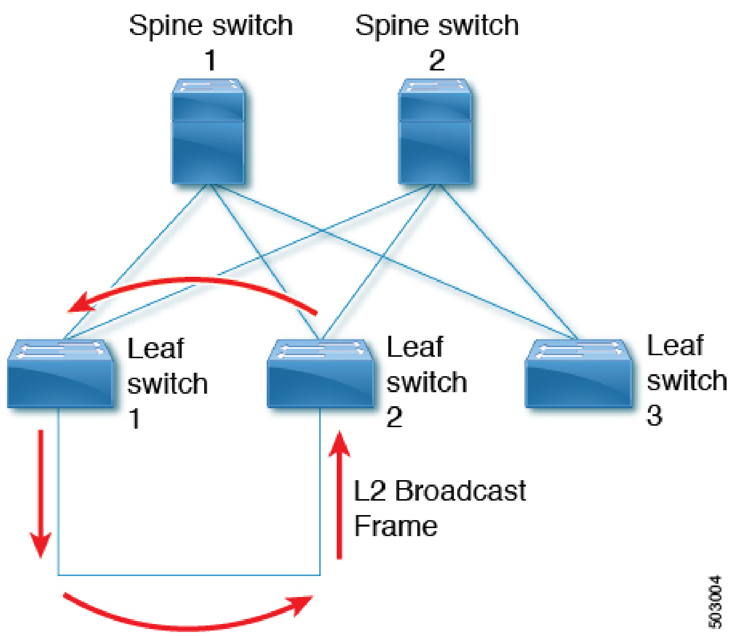
Loops usually occur in a VXLAN EVPN fabric due to incorrect cabling on the south side (access side) of the fabric. When broadcast packets are injected into a network with a loop, the frame remains bridged in the loop. As more broadcast frames enter the loop, they accumulate and can cause a serious disruption of services.

About VXLAN EVPN Loop Detection and Mitigation

Cisco NX-OS Release 9.3(5) introduces VXLAN EVPN loop detection and mitigation. This feature detects Layer 2 loops in a single VXLAN EVPN fabric or a Multi-Site environment. It operates at the port/VLAN level and disables the VLAN(s) on each port where a loop is detected. Administrators are also notified (via syslog) about the condition. In this way, the feature ensures that the network remains up and available.

The following figure shows an EVPN fabric in which two leaf devices (Leaf1 and Leaf2) are directly connected on the south side due to incorrect cabling. In this topology, Leaf3 forwards an L2 broadcast frame to Leaf1. Then the broadcast frame is repeatedly forwarded between Leaf1 and Leaf2 through the south side and the fabric. The forwarding continues until the incorrect cabling is fixed.

Figure 23: Two Leaf Nodes Directly Connected



This feature operates in three phases:

1. Loop Detection: Sends a loop detection probe under the following circumstances: when requested by a client, as part of a periodic probe task, and as soon as any port comes up.

2. Loop Mitigation: Blocks the VLANs on a port once a loop has been discovered and displays a syslog message similar to the following:

   ```
   2020 Jan 14 09:58:44 Leaf1 %NGOAM-4-SLD_LOOP_DETECTED: Loop detected - Blocking vlan
   1001 :: Eth1/3
   ```

   Because loops can lead to incorrect local MAC address learning, this phase also flushes the local and remote MAC addresses. Doing so removes any MAC addresses that are incorrectly learned.

   In the previous figure, MAC addresses can be incorrectly learned because packets from hosts sitting behind the remote leaf (Leaf3) can reach both Leaf1 and Leaf2 from the access side. As a result, the hosts incorrectly appear local to Leaf1 and Leaf2, which causes the leafs to learn their MAC addresses.

3. Loop Recovery: Once a loop is detected on a particular port or VLAN and the recovery interval has passed, recovery probes are sent to determine if the loop still exists. When NGOAM recovers from the loop, a syslog message similar to the following appears:

   ```
   2020 Jan 14 09:59:38 Leaf1 %NGOAM-4-SLD_LOOP_GONE: Loop cleared - Enabling vlan 1001
   :: Eth1/3
   ```

Note The default logging level for NGOAM does not generate a syslog message. Modifying the logging level of NGOAM to 5 with "logging level ngoam 5" will result in a syslog message being generated when a loop is detected.

Guidelines and Limitations for VXLAN NGOAM

VXLAN NGOAM has the following guidelines and limitations:

Supported Platform and Release for VXLAN NGOAM

Supported Release	Supported Platform
9.3(3) and later	Cisco Nexus 9300-FX/FX2/GX Series switches
9.3(5) and later	Cisco Nexus 9300-FX3 Series switches

Guidelines and Limitations for VXLAN EVPN Loop Detection and Mitigation

VXLAN EVPN loop detection and mitigation has the following guidelines and limitations:

- VXLAN EVPN loop detection and mitigation is supported in both STP and STP-less environments.
- To be able to detect loops across sites for VXLAN EVPN Multi-Site deployments, the **ngoam loop-detection** command needs to be configured on all border gateways in the site where the feature is being deployed.
- VXLAN EVPN loop detection and mitigation isn't supported with the following features:
    - Private VLANs
    - VLAN translation
    - ESI-based multihoming
    - VXLAN Cross Connect
    - Q-in-VNI
    - EVPN segment routing (Layer 2)

Note Ports or VLANs configured with these features must be excluded from VXLAN EVPN loop detection and mitigation. You can use the **disable** {**vlan** *vlan-range*} [**port** *port-range*] command to exclude them.

Supported Platform and Release for VXLAN EVPN Loop Detection and Mitigation

Supported Release	Supported Platform
9.3(5) and later	Cisco Nexus 9300-EX/FX/FX2 and 9332C and 9364C Series switches Cisco Nexus 9500 platform switches with 9700-EX/FX line cards
10.1(1) and later	Cisco Nexus 9300-FX3/GX Series switches

Configuring VXLAN OAM

Before you begin

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

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **feature ngoam**
3. switch(config)# **hardware access-list tcam region arp-ether 256 double-wide**
4. switch(config)# **ngoam install acl**
5. (Optional) **bcm-shell module 1 "fp show group 62"**

DETAILED STEPS

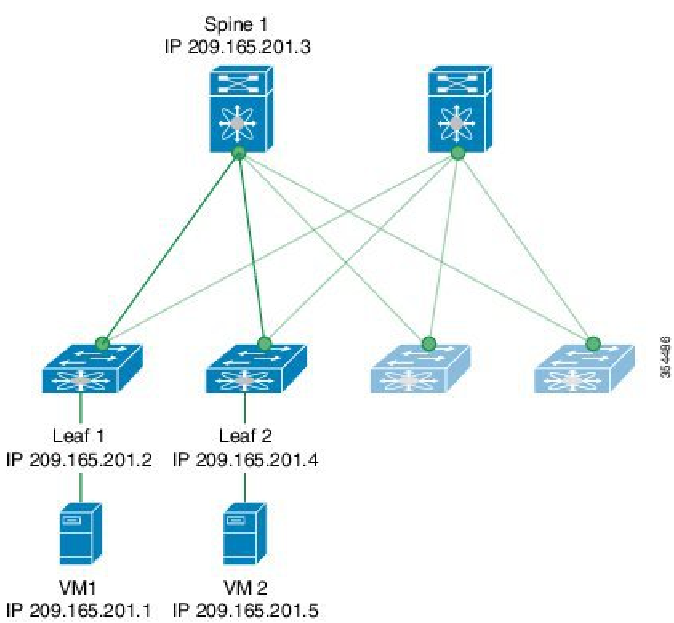
	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **feature ngoam**	Enters the NGOAM feature.
Step 3	switch(config)# **hardware access-list tcam region arp-ether 256 double-wide**	For Cisco Nexus 9300 platform switches with Network 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 prerequisite before installing the ACL rule. **Note** • Configuring the TCAM region requires the node to be rebooted. • This command is not applicable for Cisco Nexus 9300-EX/FX/FX2/GX Series switches.
Step 4	switch(config)# **ngoam install acl**	Installs the NGOAM Access Control List (ACL).

	Command or Action	Purpose
		Note This command is deprecated beginning with Cisco NX-OS Release 9.3(5) and is required only for earlier releases.
Step 5	(Optional) **bcm-shell module 1 "fp show group 62"**	For Cisco Nexus 9300 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.

Example

See the following examples of the configuration topology.

Figure 24: 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 examples display how to ping from Leaf 1 to VM2 via Spine 1 with channel 1 (unique loopback) and with channel 2 (NVO3 Draft Tissa):

```
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
                                           <<<<< add space here
switch# ping nve ip unknown vrf vni-31000 payload ip 209.165.201.5 209.165.201.4 payload-end
 verify-host
<snip>
Sender handle: 34
!!!!!
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
```

Note When the total hop count to final destination is more than 5, the path trace default TTL value is 5. Use **max-ttl** option to finish VXLAN OAM path trace completely.

For example: **pathtrace nve ip unknown vrf vrf-vni13001 payload ip 200.1.1.71 200.1.1.23 payload-end verbose max-ttl 10**

Configuring NGOAM Profile

Complete the following steps to configure NGOAM profile.

SUMMARY STEPS

1. switch(config)# **[no] feature ngoam**
2. switch(config)# **[no] ngoam profile <profile-id>**
3. switch(config-ng-oam-profile)# **?**

DETAILED STEPS

	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.

	Command or Action	Purpose
		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
```

Configuring NGOAM Southbound Loop Detection on Layer-2 Interfaces

Follow these steps to configure NGOAM Southbound loop detection and mitigation.

Before you begin

Enable the NGOAM feature.

Use the following command to create space for the TCAM ing-sup region:

```
hardware access-list tcam region ing-sup 768
```

Note

- Ensure that additional TCAM entries are freed up before increasing the allocation for the ing-sup region.
- Configuring the TCAM region requires the node to be rebooted.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# [**no**] **ngoam loop-detection**
3. (Optional) switch(config-ng-oam-loop-detection)# [**no**] **disable** {**vlan** *vlan-range*} [**port** *port-range*]
4. (Optional) switch(config-ng-oam-loop-detection)# [**no**] **periodic-probe-interval** *value*
5. (Optional) switch(config-ng-oam-loop-detection)# [**no**] **port-recovery-interval** *value*
6. (Optional) switch# **show ngoam loop-detection summary**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# [**no**] **ngoam loop-detection**	Enables NGOAM Southbound loop detection and mitigation for all VLANs or ports. This feature is disabled by default.
Step 3	(Optional) switch(config-ng-oam-loop-detection)# [**no**] **disable** {**vlan** *vlan-range*} [**port** *port-range*]	Disables NGOAM Southbound loop detection and mitigation for specific VLANs or ports and brings up any loop-detected ports. The **no** form of this command resumes active monitoring of these VLANs or ports.
Step 4	(Optional) switch(config-ng-oam-loop-detection)# [**no**] **periodic-probe-interval** *value*	Specifies how often periodic loop-detection probes are sent. The range is from 60 seconds to 3600 seconds (60 minutes). The default value is 300 seconds (5 minutes).
Step 5	(Optional) switch(config-ng-oam-loop-detection)# [**no**] **port-recovery-interval** *value*	Once a port or VLAN is shut down, specifies how often recovery probes are sent. The range is from 300 seconds to 3600 seconds (60 minutes). The default value is 600 seconds (10 minutes).
Step 6	(Optional) switch# **show ngoam loop-detection summary**	Displays the loop-detection configuration and current loop summary.

The following example shows how to configure NGOAM Southbound loop detection and mitigation:

```
switch(config)# ngoam loop-detection
switch(config-ng-oam-loop-detection)# periodic-probe-interval 200
switch(config-ng-oam-loop-detection)# port-recovery-interval 300
```

The following example shows how to disable NGOAM Southbound loop detection and mitigation on specific VLANs or VLAN ports:

```
switch(config-ng-oam-loop-detection)# disable vlan 1200 port ethernet 1/1
switch(config-ng-oam-loop-detection)# disable vlan 1300
```

What to do next

Configure a QoS policy on the spine. (For configuration example, see Configuration Examples for NGOAM Southbound Loop Detection and Mitigation, on page 219).

Detecting Loops and Bringing Up Ports On Demand

Follow the steps in this section to detect loops or bring up blocked ports on demand.

Before you begin

Enable NGOAM Southbound loop detection and mitigation.

SUMMARY STEPS

1. (Optional) switch# **ngoam loop-detection probe** {**vlan** *vlan-range*} [**port** *port-range*]
2. (Optional) switch# **ngoam loop-detection bringup** {**vlan** *vlan-range*} [**port** *port-range*]
3. (Optional) switch# **show ngoam loop-detection status** [**history**] [**vlan** *vlan-range*] [**port** *port-range*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	(Optional) switch# **ngoam loop-detection probe** {**vlan** *vlan-range*} [**port** *port-range*]	Sends a loop-detection probe on the specified VLAN or port and a notification as to whether the probe was successfully sent.
Step 2	(Optional) switch# **ngoam loop-detection bringup** {**vlan** *vlan-range*} [**port** *port-range*]	Brings up the VLANs or ports that were blocked earlier. This command also clears any entries stuck in the NGOAM. **Note** It can take up to two port-recovery intervals for the ports to come up after a loop is cleared. You can speed up the recovery by manually overriding the timer with the **ngoam loop-detection bringup vlan** {**vlan** *vlan-range*} [**port** *port-range*] command.
Step 3	(Optional) switch# **show ngoam loop-detection status** [**history**] [**vlan** *vlan-range*] [**port** *port-range*]	Displays the loop-detection status for the VLAN or port. The status can be one of the following: • BLOCKED–The VLAN or port is shut down because a loop has been detected. • FORWARDING–A loop has not been detected, and the VLAN or port is operational. • RECOVERING–Recovery probes are being sent to determine if a previously detected loop still exists.

	Command or Action	Purpose
		The **history** option displays blocked, forwarding, and recovering ports. Without the **history** option, the command displays only blocked and recovering ports.

Configuration Examples for NGOAM Southbound Loop Detection and Mitigation

The following example hows to configure a QoS policy on the spine and apply it to all of the spine interfaces to which the loop-detection-enabled leaf is connected:

```
class-map type qos match-any Spine-DSCP56
match dscp 56
policy-map type qos Spine-DSCP56
class Spine-DSCP56
set qos-group 7

interface Ethernet1/31
mtu 9216
no link dfe adaptive-tuning
service-policy type qos input Spine-DSCP5663
no ip redirects
ip address 27.4.1.2/24
ip router ospf 200 area 0.0.0.0
ip pim sparse-mode
no shutdown
```

The following sample output shows the loop-detection configuration and current loop summary:

```
switch# show ngoam loop-detection summary
Loop detection:enabled
Periodic probe interval: 200
Port recovery interval: 300
Number of vlans: 1
Number of ports: 1
Number of loops: 1
Number of ports blocked: 1
Number of vlans disabled: 0
Number of ports disabled: 0
Total number of probes sent: 214
Total number of probes received: 102
Next probe window start: Thu May 14 15:14:23 2020 (0 seconds)
Next recovery window start: Thu May 14 15:54:23 2020 (126 seconds)
```

The following sample output shows the loop-detection status for the specified VLANs or ports with and without the **history** option:

```
switch# show ngoam loop-detection status
VlanId Port   Status     NumLoops  Detection Time                ClearedTime
====== ====== ========== ========= ============================= ===============
100    Eth1/3 BLOCKED     1        Tue Apr 14 20:07:50.313 2020  Never

switch# show ngoam loop-detection status history
VlanId Port   Status     NumLoops  Detection Time                ClearedTime
====== ====== ========== ========= ============================= ===============
100    Eth1/3 BLOCKED     1        Tue Apr 14 20:07:50.313 2020  Never
200    Eth1/2 FORWARDING  1        Tue Apr 14 21:19:52.215 2020  May 11 21:30:54.830 2020
```

CHAPTER 10

Configuring vPC Multi-Homing

This chapter contains the following sections:

Advertising Primary IP Address

On a vPC enabled leaf or border leaf switch, by default all Layer-3 routes are advertised with the secondary IP address (VIP) of the leaf switch VTEP as the BGP next-hop IP address. Prefix routes and leaf switch generated routes are not synced between vPC leaf switches. Using the VIP as the BGP next-hop for these types of routes can cause traffic to be forwarded to the wrong vPC leaf or border leaf switch and black-holed. The provision to use the primary IP address (PIP) as the next-hop when advertising prefix routes or loopback interface routes in BGP on vPC enabled leaf or border leaf switches allows users to select the PIP as BGP next-hop when advertising these types of routes, so that traffic will always be forwarded to the right vPC enabled leaf or border leaf switch.

The configuration command for advertising the PIP is **advertise-pip**.

The following is a sample configuration:

```
switch(config)# router bgp 65536
  address-family 12vpn evpn
    advertise-pip
interface nve 1
    advertise virtual-rmac
```

The **advertise-pip** command lets BGP use the PIP as next-hop when advertising externally learned routes or for the redistributed direct routes if vPC is enabled.

VMAC (virtual-mac) is used with VIP and system MAC is used with PIP when the VIP/PIP feature is enabled.

With the **advertise-pip** and **advertise virtual-rmac** commands enabled, type 5 routes are advertised with PIP and type 2 routes are still advertised with VIP. In addition, VMAC will be used with VIP and system MAC will be used with PIP.

Note The **advertise-pip** and **advertise-virtual-rmac** commands must be enabled and disabled together for this feature to work properly. If you enable or disable one and not the other, it is considered an invalid configuration.

BorderPE Switches in a vPC Setup

The two borderPE switches are configured as a vPC. In a VXLAN vPC deployment, a common, virtual VTEP IP address (secondary loopback IP address) is used for communication. The common, virtual VTEP uses a system specific router MAC address. The Layer-3 prefixes or default route from the borderPE switch is advertised with this common virtual VTEP IP (secondary IP) plus the system specific router MAC address as the next hop.

Entering the **advertise-pip** and **advertise virtual-rmac** commands cause the Layer 3 prefixes or default to be advertised with the primary IP and system-specific router MAC address, the MAC addresses to be advertised with the secondary IP, and a router MAC address derived from the secondary IP address.

DHCP Configuration in a vPC Setup

When DHCP or DHCPv6 relay function is configured on leaf switches in a vPC setup, and the DHCP server is in the non default, non management VRF, then configure the **advertise-pip** command on the vPC leaf switches. This allows BGP EVPN to advertise Route-type 5 routes with the next-hop using the primary IP address of the VTEP interface.

The following is a sample configuration:

```
switch(config)# router bgp 100
  address-family l2vpn evpn
    advertise-pip
interface nve 1
  advertise virtual-rmac
```

IP Prefix Advertisement in vPC Setup

There are 3 types of Layer-3 routes that can be advertised by BGP EVPN. They are:

- Local host routes—These routes are learned from the attached servers or hosts.
- Prefix routes—These routes are learned via other routing protocol at the leaf, border leaf and border spine switches.
- Leaf switch generated routes—These routes include interface routes and static routes.

CHAPTER 11

Configuring Multi-Site

This chapter contains the following sections:

About VXLAN EVPN Multi-Site

The VXLAN EVPN Multi-Site solution interconnects two or more BGP-based Ethernet VPN (EVPN) sites/fabrics (overlay domains) in a scalable fashion over an IP-only network. This solution uses border gateways (BGWs) in anycast or vPC mode to terminate and interconnect two sites. The BGWs provide the network control boundary that is necessary for traffic enforcement and failure containment functionality.

In the BGP control plane for releases prior to Cisco NX-OS Release 9.3(5), BGP sessions between the BGWs rewrite the next hop information of EVPN routes and reoriginate them. Beginning with Cisco NX-OS Release 9.3(5), reorigination is always enabled (with either single or dual route distinguishers), and rewrite is not performed. For more information, see Dual RD Support for Multi-Site, on page 224.

VXLAN Tunnel Endpoints (VTEPs) are only aware of their overlay domain internal neighbors, including the BGWs. All routes external to the fabric have a next hop on the BGWs for Layer 2 and Layer 3 traffic.

The BGW is the node that interacts with nodes within a site and with nodes that are external to the site. For example, in a leaf-spine data center fabric, it can be a leaf, a spine, or a separate device acting as a gateway to interconnect the sites.

The VXLAN EVPN Multi-Site feature can be conceptualized as multiple site-local EVPN control planes and IP forwarding domains interconnected via a single common EVPN control and IP forwarding domain. Every

EVPN node is identified with a unique site-scope identifier. A site-local EVPN domain consists of EVPN nodes with the same site identifier. BGWs on one hand are also part of the site-specific EVPN domain and on the other hand a part of a common EVPN domain to interconnect with BGWs from other sites. For a given site, these BGWs facilitate site-specific nodes to visualize all other sites to be reachable only via them. This means:

- Site-local bridging domains are interconnected only via BGWs with bridging domains from other sites.
- Site-local routing domains are interconnected only via BGWs with routing domains from other sites.
- Site-local flood domains are interconnected only via BGWs with flood domains from other sites.

Selective Advertisement is defined as the configuration of the per-tenant information on the BGW. Specifically, this means IP VRF or MAC VRF (EVPN instance). In cases where external connectivity (VRF-lite) and EVPN Multi-Site coexist on the same BGW, the advertisements are always enabled.

Dual RD Support for Multi-Site

Beginning with Cisco NX-OS Release 9.3(5), VXLAN EVPN Multi-Site supports route reorigination with dual route distinguishers (RDs). This behavior is enabled automatically.

Each VRF or L2VNI tracks two RDs: a primary RD (which is unique) and a secondary RD (which is the same across BGWs). Reoriginated routes are advertised with the secondary type-0 RD (site-id:VNI). All other routes are advertised with the primary RD. The secondary RD is allocated automatically once the router is in Multi-Site BGW mode.

If the site ID is greater than 2 bytes, the secondary RD can't be generated automatically on the Multi-Site BGW, and the following message appears:

```
%BGP-4-DUAL_RD_GENERATION_FAILED: bgp- [12564] Unable to generate dual RD on EVPN multisite
 border gateway. This may increase memory consumption on other BGP routers receiving
re-originated EVPN routes. Configure router bgp <asn> ; rd dual id <id> to avoid it.
```

In this case, you can either manually configure the secondary RD value or disable dual RDs. For more information, see Configuring Dual RD Support for Multi-Site, on page 229.

Guidelines and Limitations for VXLAN EVPN Multi-Site

VXLAN EVPN Multi-Site has the following configuration guidelines and limitations:

- The following switches support VXLAN EVPN Multi-Site:
    - Cisco Nexus 9300-EX and 9300-FX platform switches (except Cisco Nexus 9348GC-FXP platform switches)
    - Cisco Nexus 9300-FX2 platform switches
    - Cisco Nexus 9300-FX3 platform switches
    - Cisco Nexus 9300-GX platform switches
    - Cisco Nexus 9500 platform switches with -EX or -FX or -GX line cards

Note Cisco Nexus 9500 platform switches with -R/RX line cards don't support VXLAN EVPN Multi-Site.

- The **evpn multisite fabric-tracking** is mandatory only for anycast BGWs. For vPC based BGWs, this command is not mandatory. The NVE Interface will be brought up with just the dci tracked link in the up state.
- Cisco Nexus 9332C and 9364C platform switches can be BGWs.
- In a VXLAN EVPN Multi-Site deployment, when you use the ttag feature, make sure that the ttag is stripped (**ttag-strip**) on BGW's DCI interfaces that connect to the cloud. To elaborate, if the ttag is attached to non-Nexus 9000 devices that do not support EtherType 0x8905, stripping of the ttag is required. However, BGW back-to-back model of DCI does not require ttag stripping.
- VXLAN EVPN Multi-Site and Tenant Routed Multicast (TRM) are supported between sources and receivers deployed across different sites.
- The Multi-Site BGW allows the coexistence of Multi-Site extensions (Layer 2 unicast/multicast and Layer 3 unicast) as well as Layer 3 unicast and multicast external connectivity.
- In TRM with multi-site deployments, all BGWs receive traffic from fabric. However, only the designated forwarder (DF) BGW forwards the traffic. All other BGWs drop the traffic through a default drop ACL. This ACL is programmed in all DCI tracking ports. Don't remove the **evpn multisite dci-tracking** configuration from the DCI uplink ports. If you do, you remove the ACL, which creates a nondeterministic traffic flow in which packets can be dropped or duplicated instead of deterministically forwarded by only one BGW, the DF.
- Anycast mode can support up to six BGWs per site.
- BGWs in a vPC topology are supported.
- Multicast Flood Domain between inter-site/fabric BGWs isn't supported.
- iBGP EVPN Peering between BGWs of different fabrics/sites isn't supported.
- The **peer-type fabric-external** command configuration is required only for VXLAN Multi-Site BGWs (this command must not be used when peering with non-Cisco equipment).

Note The **peer-type fabric-external** command configuration is not required for pseudo BGWs.

- Anycast mode can support only Layer 3 services that are attached to local interfaces.
- In Anycast mode, BUM is replicated to each border leaf. DF election between the border leafs for a particular site determines which border leaf forwards the inter-site traffic (fabric to DCI and conversely) for that site.
- In Anycast mode, all Layer 3 services are advertised in BGP via EVPN Type-5 routes with their physical IP as the next hop.
- vPC mode can support only two BGWs.

- vPC mode can support both Layer 2 hosts and Layer 3 services on local interfaces.
- In vPC mode, BUM is replicated to either of the BGWs for traffic coming from the external site. Hence, both BGWs are forwarders for site external to site internal (DCI to fabric) direction.
- In vPC mode, BUM is replicated to either of the BGWs for traffic coming from the local site leaf for a VLAN using Ingress Replication (IR) underlay. Both BGWs are forwarders for site internal to site external (fabric to DCI) direction for VLANs using the IR underlay.
- In vPC mode, BUM is replicated to both BGWs for traffic coming from the local site leaf for a VLAN using the multicast underlay. Therefore, a decapper/forwarder election happens, and the decapsulation winner/forwarder only forwards the site-local traffic to external site BGWs for VLANs using the multicast underlay.
- In vPC mode, all Layer 3 services/attachments are advertised in BGP via EVPN Type-5 routes with their virtual IP as next hop. If the VIP/PIP feature is configured, they are advertised with PIP as the next hop.
- If different Anycast Gateway MAC addresses are configured across sites, enable ARP suppression for all VLANs that have been extended.
- Bind NVE to a loopback address that is separate from loopback addresses that are required by Layer 3 protocols. A best practice is to use a dedicated loopback address for the NVE source interface (PIP VTEP) and multi-site source interface (anycast and virtual IP VTEP).
- PIM BiDir is not supported for fabric underlay multicast replication with VXLAN Multi-Site.
- PIM is not supported on Multi-Site VXLAN DCI links.
- FEX is not supported on a vPC BGW and Anycast BGW.
- Beginning with Cisco NX-OS Release 9.3(5), VTEPs support VXLAN-encapsulated traffic over parent interfaces if subinterfaces are configured. This feature is supported for VXLAN EVPN Multi-Site and DCI. DCI tracking can be enabled only on the parent interface.
- Beginning with Cisco NX-OS Release 9.3(5), VXLAN EVPN Multi-Site supports asymmetric VNIs. For more information, see Multi-Site with Asymmetric VNIs and Configuration Example for Multi-Site with Asymmetric VNIs, on page 239.
- The following guidelines and limitations apply to dual RD support for Multi-Site:
  - Dual RD are supported beginning with Cisco NX-OS Release 9.3(5).
  - Dual RD is enabled automatically for Cisco Nexus 9332C, 9364C, 9300-EX, and 9300-FX/FX2 platform switches and Cisco Nexus 9500 platform switches with -EX/FX line cards that have VXLAN EVPN Multi-Site enabled.
  - To use CloudSec or other features that require PIP advertisement for multi-site reoriginated routes, configure BGP additional paths on the route server if dual RD are enabled on the BGW, or disable dual RD.
  - Sending secondary RD additional paths at the BGW node isn't supported.
  - During an ISSU, the number of paths for the leaf nodes might double temporarily while all BGWs are being upgraded.
- Beginning with Cisco NX-OS Release 9.3(5), if you disable the **host-reachability protocol bgp** command under the NVE interface in a VXLAN EVPN Multi-Site topology, the NVE interface stays operationally down.

- Beginning with Cisco NX-OS Release 9.3(5), Multi-Site Border Gateways re-originate incoming remote routes when advertising to the site's local spine/leaf switches. These re-originated routes modify the following fields:
  - RD value changes to [Multisite Site ID:L3 VNID].
  - It is mandatory that Route-Targets are defined on all VTEP that are participating in a given VRF, this includes and is explicitly required for the BGW to extend the given VRF. Prior to Cisco NX-OS Release 9.3(5), Route-Targets from intra-site VTEPs were inadvertently kept across the site boundary, even if not defined on the BGW. Starting from Cisco NX-OS Release 9.3(5) the mandatory behavior is enforced. By adding the necessary Route-Targets to the BGW, the change from inadvertent Route-Target advertisement to explicit Route-Target advertisement can be performed.
  - Path type changes from external to local.
- To improve the convergence in case of fabric link failure and avoid issues in case of fabric link flapping, ensure to configure multi-hop BFD between loopbacks of spines and BGWs.

  In the specific scenario where a BGW node becomes completely isolated from the fabric due to all its fabric links failing, the use of multi-hop BFD ensures that the BGP sessions between the spines and the isolated BGW can be immediately brought down, without relying on the configured BGP hold-time value.
- In a VXLAN Multi-Site environment, a border gateway device that uses ECMP for routing through both a VXLAN overlay and an L3 prefix to access remote site subnets might encounter adjacency resolution failure for one of these routes. If the switch attempts to use this unresolved prefix, it will result in traffic being dropped.
- For SVI-related triggers (such as shut/unshut or PIM enable/disable), a 30-second delay was added, allowing the Multicast FIB (MFIB) Distribution module (MFDM) to clear the hardware table before toggling between L2 and L3 modes or vice versa.

Enabling VXLAN EVPN Multi-Site

This procedure enables the VXLAN EVPN Multi-Site feature. Multi-Site is enabled on the BGWs only. The site-id must be the same on all BGWs in the fabric/site.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**evpn multisite border-gateway** *ms-id* **Example:** `switch(config)# evpn multisite border-gateway 100`	Configures the site ID for a site/fabric. The range of values for *ms-id* is 1 to 2,814,749,767,110,655. The *ms-id* must be the same in all BGWs within the same fabric/site.

	Command or Action	Purpose
Step 3	**interface nve 1** **Example:** `switch(config-evpn-msite-bgw)# interface nve 1`	Creates a VXLAN overlay interface that terminates VXLAN tunnels. **Note** Only one NVE interface is allowed on the switch.
Step 4	**source-interface loopback** *src-if* **Example:** `switch(config-if-nve)# source-interface loopback 0`	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 transient devices in the transport network and the remote VTEPs. This requirement is accomplished by advertising it through a dynamic routing protocol in the transport network.
Step 5	**host-reachability protocol bgp** **Example:** `switch(config-if-nve)# host-reachability protocol bgp`	Defines BGP as the mechanism for host reachability advertisement.
Step 6	**multisite border-gateway interface loopback** *vi-num* **Example:** `switch(config-if-nve)# multisite border-gateway interface loopback 100`	Defines the loopback interface used for the BGW virtual IP address (VIP). The border-gateway 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 transient devices in the transport network and the remote VTEPs. This requirement is accomplished by advertising it through a dynamic routing protocol in the transport network. This loopback must be different than the source interface loopback. The range of *vi-num* is from 0 to 1023.
Step 7	**no shutdown** **Example:** `switch(config-if-nve)# no shutdown`	Negates the **shutdown** command.
Step 8	**exit** **Example:** `switch(config-if-nve)# exit`	Exits the NVE configuration mode.
Step 9	**interface loopback** *loopback-number* **Example:** `switch(config)# interface loopback 0`	Configures the loopback interface.
Step 10	**ip address** *ip-address* **Example:** `switch(config-if)# ip address 198.0.2.0/32`	Configures the IP address for the loopback interface.

Configuring Dual RD Support for Multi-Site

Follow these steps if you need to manually configure the secondary RD value or disable dual RDs.

Before you begin

Enable VXLAN EVPN Multi-Site.

Procedure

	Command or Action	Purpose

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

Step 2	**router bgp** *as-num*
	Example:
	`switch(config)# router bgp 100`
	`switch(config-router)#`
		Configures the autonomous system number. The range for *as-num* is from 1 to 4,294,967,295.

Step 3	[**no**] **rd dual id** [*2-bytes*] **Example:** `switch(config-router)# rd dual id 1`	Defines the first 2 bytes of the secondary RD. The ID must be the same across the Multi-Site BGWs. The range is from 1 to 65535. **Note** If necessary, you can use the **no rd dual** command to disable dual RDs and fall back to a single RD.
Step 4	(Optional) **show bgp evi** *evi-id* **Example:** `switch(config-router)# show bgp evi 100`	Displays the secondary RD configured as part of the **rd dual id** [*2-bytes*] command for the specified EVI.

Example

The following example shows sample output for the **show bgp evi** *evi-id* command:

```
switch# show bgp evi 100
-----------------------------------------------
  L2VNI ID                      : 100 (L2-100)
  RD                            : 3.3.3.3:32867
  Secondary RD                  : 1:100
  Prefixes (local/total)        : 1/6
  Created                       : Jun 23 22:35:13.368170
  Last Oper Up/Down             : Jun 23 22:35:13.369005 / never
  Enabled                       : Yes

  Active Export RT list         :
        100:100
  Active Import RT list         :
```

```
100:100
```

Configuring VNI Dual Mode

This procedure describes the configuration of the BUM traffic domain for a given VLAN. Support exists for using multicast or ingress replication inside the fabric/site and ingress replication across different fabrics/sites.

Note If you have multiple VRFs and only one is extended to ALL leaf switches, you can add a dummy loopback to that one extended VRF and advertise through BGP. Otherwise, you'll need to check how many VRFs are extended and to which switches, and then add a dummy loopback to the respective VRFs and advertise them as well. Therefore, use the **advertise-pip** command to prevent potential user errors in the future.

For more information about configuring multicast or ingress replication for a large number of VNIs, see Example of VXLAN BGP EVPN (EBGP), on page 113.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**interface nve 1** **Example:** `switch(config)# interface nve 1`	Creates a VXLAN overlay interface that terminates VXLAN tunnels. **Note** Only one NVE interface is allowed on the switch.
Step 3	**member vni** *vni-range* **Example:** `switch(config-if-nve)# member vni 200`	Configures the virtual network identifier (VNI). The range for *vni-range* is from 1 to 16,777,214. The value of *vni-range* can be a single value like 5000 or a range like 5001-5008. **Note** Enter one of the Step 4 or Step 5 commands.
Step 4	**mcast-group** *ip-addr* **Example:** `switch(config-if-nve-vni)# mcast-group 255.0.4.1`	Configures the NVE Multicast group IP prefix within the fabric.
Step 5	**ingress-replication protocol bgp** **Example:** `switch(config-if-nve-vni)# ingress-replication protocol bgp`	Enables BGP EVPN with ingress replication for the VNI within the fabric.
Step 6	**multisite ingress-replication** **Example:**	Defines the Multi-Site BUM replication method for extending the Layer 2 VNI.

	Command or Action	Purpose
	`switch(config-if-nve-vni)# multisite ingress-replication`	

Configuring Fabric/DCI Link Tracking

This procedure describes the configuration to track all DCI-facing interfaces and site internal/fabric facing interfaces. Tracking is mandatory and is used to disable reorigination of EVPN routes either from or to a site if all the DCI/fabric links go down.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**interface ethernet** *port* **Example:** `switch(config)# interface ethernet1/1`	Enters interface configuration mode for the DCI or fabric interface. **Note** Enter one of the following commands in Step 3 or Step 4.
Step 3	**evpn multisite dci-tracking** **Example:** `switch(config-if)# evpn multisite dci-tracking`	Configures DCI interface tracking.
Step 4	(Optional) **evpn multisite fabric-tracking** **Example:** `switch(config-if)# evpn multisite fabric-tracking`	Configures EVPN Multi-Site fabric tracking. The **evpn multisite fabric-tracking** is mandatory for anycast BGWs and vPC BGW fabric links.
Step 5	**ip address** *ip-addr* **Example:** `switch(config-if)# ip address 192.1.1.1`	Configures the IP address.
Step 6	**no shutdown** **Example:** `switch(config-if)# no shutdown`	Negates the **shutdown** command.

Configuring Fabric External Neighbors

This procedure describes the configuration of fabric external/DCI neighbors for communication to other site/fabric BGWs.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**router bgp** *as-num* **Example:** `switch(config)# router bgp 100`	Configures the autonomous system number. The range for *as-num* is from 1 to 4,294,967,295.
Step 3	**neighbor** *ip-addr* **Example:** `switch(config-router)# neighbor 100.0.0.1`	Configures a BGP neighbor.
Step 4	**remote-as** *value* **Example:** `switch(config-router-neighbor)# remote-as 69000`	Configures remote peer's autonomous system number.
Step 5	**peer-type fabric-external** **Example:** `switch(config-router-neighbor)# peer-type fabric-external`	Enables the next hop rewrite for Multi-Site. Defines site external BGP neighbors for EVPN exchange. The default for **peer-type** is **fabric-internal**. **Note** The **peer-type fabric-external** command is required only for VXLAN Multi-Site BGWs. It is not required for pseudo BGWs.
Step 6	**address-family l2vpn evpn** **Example:** `switch(config-router-neighbor)# address-family l2vpn evpn`	Configures the address family Layer 2 VPN EVPN under the BGP neighbor.
Step 7	**rewrite-evpn-rt-asn** **Example:** `switch(config-router-neighbor)# rewrite-evpn-rt-asn`	Rewrites the route target (RT) information to simplify the MAC-VRF and IP-VRF configuration. BGP receives a route, and as it processes the RT attributes, it checks if the AS value matches the peer AS that is sending that route and replaces it. Specifically, this command changes the incoming route target's AS number to match the BGP-configured neighbor's remote AS number. You can see the modified RT value in the receiver router.

Configuring VXLAN EVPN Multi-Site Storm Control

VXLAN EVPN Multi-Site Storm Control allows rate limiting of multidestination (BUM) traffic on Multi-Site BGWs. You can control BUM traffic sent over the DCI link using a policer on fabric links in the ingress direction.

Remote peer reachability must be only through DCI links. Appropriate routing configuration must ensure that remote site routes are not advertised over Fabric links.

Multicast traffic is policed only on DCI interfaces, while unknown unicast and broadcast traffic is policed on both DCI and fabric interfaces.

Cisco NX-OS Release 9.3(6) and later releases optimize rate granularity and accuracy. Bandwidth is calculated based on the accumulated DCI uplink bandwidth, and only interfaces tagged with DCI tracking are considered. (Prior releases also include fabric-tagged interfaces.) In addition, granularity is enhanced by supporting two digits after the decimal point. These enhancements apply to the Cisco Nexus 9300-EX, 9300-FX/FX2/FX3, and 9300-GX platform switches.

Note For information on access port storm control, see the Cisco Nexus 9000 Series NX-OS Layer 2 Configuration Guide.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **evpn storm-control** {**broadcast** | **multicast** | **unicast**} {**level** *level*}

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	[**no**] **evpn storm-control** {**broadcast** | **multicast** | **unicast**} {**level** *level*}
	Example:
	`switch(config)# evpn storm-control unicast level 10`
	Example:
	`switch(config)# evpn storm-control unicast level 10.20`
		Configures the storm suppression level as a number from 0–100.
		0 means that all traffic is dropped, and 100 means that all traffic is allowed. For any value in between, the unknown unicast traffic rate is restricted to a percentage of available bandwidth. For example, a value of 10 means that the traffic rate is restricted to 10% of the available bandwidth, and anything above that rate is dropped.
		Beginning with Cisco NX-OS Release 9.3(6), you can configure the level as a fractional value by adding two digits after the decimal point. For example, you can enter a value of 10.20.

Verifying VXLAN EVPN Multi-Site Storm Control

To display EVPN storm control setting information, enter the following command:

Command	Purpose
slot 1 show hardware vxlan storm-control	Displays the status of EVPN storm control setting.

Note Once the Storm control hits the threshold, a message is logged as stated below:

```
BGWY-1 %ETHPORT-5-STORM_CONTROL_ABOVE_THRESHOLD: Traffic in port Ethernet1/32 exceeds the
configured threshold , action - Trap (message repeated 38 times)
```

Multi-Site with vPC Support

About Multi-Site with vPC Support

The BGWs can be in a vPC complex. In this case, it is possible to support dually-attached directly-connected hosts that might be bridged or routed as well as dually-attached firewalls or service attachments. The vPC BGWs have vPC-specific multihoming techniques and do not rely on EVPN Type 4 routes for DF election or split horizon.

Guidelines and Limitations for Multi-Site with vPC Support

Multi-Site with vPC support has the following configuration guidelines and limitations:

- 4000 VNIs for vPC are not supported.
- For BUM with continued VIP use, the MCT link is used as transport upon core isolation or fabric isolation, and for unicast traffic in fabric isolation.
- Beginning with Cisco NX-OS Release 10.1(2), TRM Multisite with vPC BGW is supported.
- The routes to remote Multisite BGW loopback addresses must always prioritize the DCI link path over the iBGP protocol between vPC Border Gateway switches configured using the backup SVI. The backup SVI should be used strictly in the event of a DCI link failure.

Configuring Multi-Site with vPC Support

This procedure describes the configuration of Multi-Site with vPC support:

- Configure vPC domain.
- Configure port channels.
- Configuring vPC Peer Link.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**feature vpc** **Example:** `switch(config)# feature vpc`	Enables vPCs on the device.
Step 3	**feature interface-vlan** **Example:** `switch(config)# feature interface-vlan`	Enables the interface VLAN feature on the device.
Step 4	**feature lacp** **Example:** `switch(config)# feature lacp`	Enables the LACP feature on the device.
Step 5	**feature pim** **Example:** `switch(config)# feature pim`	Enables the PIM feature on the device.
Step 6	**feature ospf** **Example:** `switch(config)# feature ospf`	Enables the OSPF feature on the device.
Step 7	**ip pim rp-address** *address* **group-list** *range* **Example:** `switch(config)# ip pim rp-address 100.100.100.1 group-list 224.0.0/4`	Defines a PIM RP address for the underlay multicast group range.
Step 8	**vpc domain** *domain-id* **Example:** `switch(config)# vpc domain 1`	Creates a vPC domain on the device and enters vpn-domain configuration mode for configuration purposes. There is no default. The range is from 1 to 1000.
Step 9	**peer switch** **Example:** `switch(config-vpc-domain)# peer switch`	Defines the peer switch.
Step 10	**peer gateway** **Example:** `switch(config-vpc-domain)# peer gateway`	Enables Layer 3 forwarding for packets destined to the gateway MAC address of the vPC.
Step 11	**peer-keepalive destination** *ip-address* **Example:**	Configures the IPv4 address for the remote end of the vPC peer-keepalive link.

	Command or Action	Purpose
	`switch(config-vpc-domain)# peer-keepalive destination 172.28.230.85`	**Note** The system does not form the vPC peer link until you configure a vPC peer-keepalive link. The management ports and VRF are the defaults.
Step 12	**ip arp synchronize** **Example:** `switch(config-vpc-domain)# ip arp synchronize`	Enables IP ARP synchronize under the vPC domain to facilitate faster ARP table population following device reload.
Step 13	**ipv6 nd synchronize** **Example:** `switch(config-vpc-domain)# ipv6 nd synchronize`	Enables IPv6 ND synchronization under the vPC domain to facilitate faster ND table population following device reload.

Step 14	Create the vPC peer-link.
	Example:
	`switch(config)# interface port-channel 1`
	`switch(config)# switchport`
	`switch(config)# switchport mode trunk`
	`switch(config)# switchport trunk allowed vlan 1,10,100-200`
	`switch(config)# mtu 9216`
	`switch(config)# vpc peer-link`
	`switch(config)# no shut`
	
	`switch(config)# interface Ethernet 1/1, 1/21`
	`switch(config)# switchport`
	`switch(config)# mtu 9216`
	`switch(config)# channel-group 1 mode active`
	`switch(config)# no shutdown`
		Creates the vPC peer-link port-channel interface and adds two member interfaces to it.

Step 15	**system nve infra-vlans** *range* **Example:** `switch(config)# system nve infra-vlans 10`	Defines a non-VXLAN-enabled VLAN as a backup routed path.
Step 16	**vlan** *number* **Example:** `switch(config)# vlan 10`	Creates the VLAN to be used as an infra-VLAN.

Step 17	Create the SVI.
	Example:
	`switch(config)# interface vlan 10`
	`switch(config)# ip address 10.10.10.1/30`
	`switch(config)# ip router ospf process UNDERLAY area 0`
	`switch(config)# ip pim sparse-mode`
	`switch(config)# no ip redirects`
	`switch(config)# mtu 9216`
	`switch(config)# no shutdown`
		Creates the SVI used for the backup routed path over the vPC peer-link.

	Command or Action	Purpose
Step 18	(Optional) **delay restore interface-vlan** *seconds* **Example:** `switch(config-vpc-domain)# delay restore interface-vlan 45`	Enables the delay restore timer for SVIs. We recommend tuning this value when the SVI/VNI scale is high. For example, when the SCI count is 1000, we recommend that you set the delay restore to 45 seconds.
Step 19	**evpn multisite border-gateway** *ms-id* **Example:** `switch(config)# evpn multisite border-gateway 100`	Configures the site ID for a site/fabric. The range of values for *ms-id* is 1 to 281474976710655. The *ms-id* must be the same in all BGWs within the same fabric/site.
Step 20	**interface nve 1** **Example:** `switch(config-evpn-msite-bgw)# interface nve 1`	Creates a VXLAN overlay interface that terminates VXLAN tunnels. **Note** Only one NVE interface is allowed on the switch.
Step 21	**source-interface loopback** *src-if* **Example:** `switch(config-if-nve)# source-interface loopback 0`	Defines the source interface, which must be a loopback interface with a valid /32 IP address. This /32 IP address must be known by the transient devices in the transport network and the remote VTEPs. This requirement is accomplished by advertising the address through a dynamic routing protocol in the transport network.
Step 22	**host-reachability protocol bgp** **Example:** `switch(config-if-nve)# host-reachability protocol bgp`	Defines BGP as the mechanism for host reachability advertisement.
Step 23	**multisite border-gateway interface loopback** *vi-num* **Example:** `switch(config-if-nve)# multisite border-gateway interface loopback 100`	Defines the loopback interface used for the BGW virtual IP address (VIP). The BGW 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 transient devices in the transport network and the remote VTEPs. This requirement is accomplished by advertising the address through a dynamic routing protocol in the transport network. This loopback must be different than the source interface loopback. The range of *vi-num* is from 0 to 1023.
Step 24	**no shutdown** **Example:** `switch(config-if-nve)# no shutdown`	Negates the **shutdown** command.
Step 25	**exit** **Example:** `switch(config-if-nve)# exit`	Exits the NVE configuration mode.
Step 26	**interface loopback** *loopback-number* **Example:**	Configures the loopback interface.

	Command or Action	Purpose
	switch(config)# **interface loopback 0**	
Step 27	**ip address** *ip-address* **Example:** switch(config-if)# **ip address 198.0.2.0/32**	Configures the primary IP address for the loopback interface.
Step 28	**ip address** *ip-address* **secondary** **Example:** switch(config-if)# **ip address 198.0.2.1/32** secondary	Configures the secondary IP address for the loopback interface.
Step 29	**ip pim sparse-mode** **Example:** switch(config-if)# **ip pim sparse-mode**	Configures PIM sparse mode on the loopback interface.

Verifying the Multi-Site with vPC Support Configuration

To display Multi-Site with vPC support information, enter one of the following commands:

show vpc brief	Displays general vPC and CC status.
show vpc consistency-parameters global	Displays the status of those parameters that must be consistent across all vPC interfaces.
show vpc consistency-parameters vni	Displays configuration information for VNIs under the NVE interface that must be consistent across both vPC peers.

Output example for the **show vpc brief** command:

```
switch# show vpc brief
Legend:
                (*) - local vPC is down, forwarding via vPC peer-link

vPC domain id                     : 1
Peer status                       : peer adjacency formed ok     (<--- peer up)
vPC keep-alive status             : peer is alive
Configuration consistency status  : success (<----- CC passed)
Per-vlan consistency status       : success                       (<---- per-VNI CCpassed)
Type-2 consistency status         : success
vPC role                          : secondary
Number of vPCs configured         : 1
Peer Gateway                      : Enabled
Dual-active excluded VLANs        : -
Graceful Consistency Check        : Enabled
Auto-recovery status              : Enabled, timer is off.(timeout = 240s)
Delay-restore status              : Timer is off.(timeout = 30s)
Delay-restore SVI status          : Timer is off.(timeout = 10s)
Operational Layer3 Peer-router    : Disabled
[...]
```

Output example for the **show vpc consistency-parameters global** command:

```
switch# show vpc consistency-parameters global

    Legend:
        Type 1 : vPC will be suspended in case of mismatch

Name                        Type  Local Value            Peer Value
-------------               ----  ---------------------- -----------------------
[...]
Nve1 Adm St, Src Adm St,    1     Up, Up, 2.1.44.5, CP,  Up, Up, 2.1.44.5, CP,
Sec IP, Host Reach, VMAC          TRUE, Disabled,        TRUE, Disabled,
Adv, SA,mcast l2, mcast           0.0.0.0, 0.0.0.0,      0.0.0.0, 0.0.0.0,
l3, IR BGP,MS Adm St, Reo         Disabled, Up,          Disabled, Up,
                                  200.200.200.200        200.200.200.200
[...]
```

Output example for the **show vpc consistency-parameters vni** command:

```
switch(config-if-nve-vni)# show vpc consistency-parameters vni

    Legend:
        Type 1 : vPC will be suspended in case of mismatch

Name                        Type  Local Value            Peer Value
-------------               ----  ---------------------- -----------------------
Nve1 Vni, Mcast, Mode,      1     11577, 234.1.1.1,      11577, 234.1.1.1,
Type, Flags                       Mcast, L2, MS IR       Mcast, L2, MS IR
Nve1 Vni, Mcast, Mode,      1     11576, 234.1.1.1,      11576, 234.1.1.1,
Type, Flags                       Mcast, L2, MS IR       Mcast, L2, MS IR
[...]
```

Configuration Example for Multi-Site with Asymmetric VNIs

The following example shows how two sites with different sets of VNIs can connect to the same MAC VRF or IP VRF. One site uses VNI 200 internally, and the other site uses VNI 300 internally. Route-target auto no longer matches because the VNI values are different. Therefore, the route-target values must be manually configured. In this example, the value 222:333 stitches together the two VNIs from different sites.

The BGW of site 1 has L2VNI 200 and L3VNI 201.

The BGW of site 2 has L2VNI 300 and L3VNI 301.

Note This configuration example assumes that basic Multi-Site configurations are already in place.

Note You must have VLAN-to-VRF mapping on the BGW. This requirement is necessary to maintain L2VNI-to-L3VNI mapping, which is needed for reorigination of MAC-IP routes at BGWs.

Layer 3 Configuration

In the BGW node of site 1, configure the common RT 201:301 for stitching the two sites using L3VNI 201 and L3VNI 301:

```
vrf context vni201
  vni 201
  address-family ipv4 unicast
```

```
    route-target both auto evpn
    route-target import 201:301 evpn
    route-target export 201:301 evpn
```

In the BGW node of site 2, configure the common RT 201:301 for stitching the two sites using L3VNI 201 and L3VNI 301:

```
vrf context vni301
  vni 301
  address-family ipv4 unicast
    route-target both auto evpn
    route-target import 201:301 evpn
    route-target export 201:301 evpn
```

Layer 2 Configuration

In the BGW node of site 1, configure the common RT 222:333 for stitching the two sites using L2VNI 200 and L2VNI 300:

```
evpn
  vni 200 l2
    rd auto
    route-target import auto
    route-target import 222:333
    route-target export auto
    route-target export 222:333
```

For proper reorigination of L3 labels of MAC-IP routes, associate the VRF (L3VNI) to the L2VNI:

```
interface Vlan 200
  vrf member vni201
```

In the BGW node of site 2, configure the common RT 222:333 for stitching the two sites using L2VNI 200 and L2VNI 300:

```
evpn
  vni 300 l2
    rd auto
    route-target import auto
    route-target import 222:333
    route-target export auto
    route-target export 222:333
```

For proper reorigination of L3 labels of MAC-IP routes, associate the VRF (L3VNI) to the L2VNI:

```
interface vlan 300
  vrf member vni301
```

TRM with Multi-Site

This section contains the following topics:

Information About Configuring TRM with Multi-Site

Tenant Routed Multicast (TRM) with Multi-Site enables multicast forwarding across multiple VXLAN EVPN fabrics that are connected via Multi-Site. This feature provides Layer 3 multicast services across sites for sources and receivers across different sites. It addresses the requirement of East-West multicast traffic between sites.

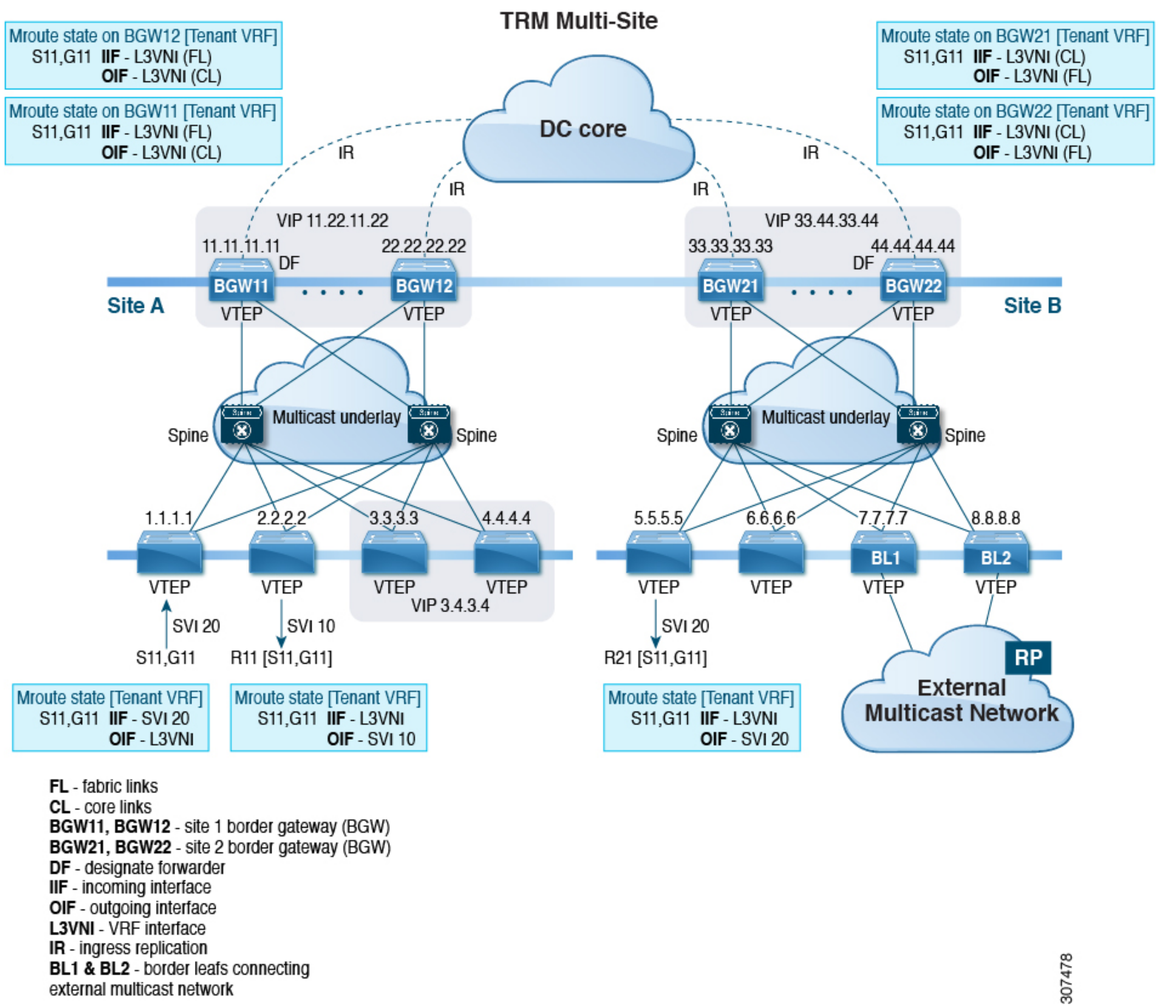
Each TRM site is operating independently. Border gateways on each site allow stitching across the sites. There can be multiple border gateways for each site. Multicast source and receiver information across sites is propagated by BGP on the border gateways that are configured with TRM. The border gateway on each site receives the multicast packet and re-encapsulates the packet before sending it to the local site. Beginning with Cisco NX-OS Release 10.1(2), TRM with Multi-Site supports both Anycast Border Gateway and vPC Border Gateway.

The border gateway that is elected as Designated Forwarder (DF) for the L3VNI forwards the traffic from fabric toward the core side. In the TRM Multicast-Anycast Gateway model, we use the VIP-R based model to send traffic toward remote sites. The IR destination IP is the VIP-R of the remote site. Each site that has the receiver gets only one copy from the source site. DF forwarding is applicable only on Anycast Border Gateways.

Note Only the DF sends the traffic toward remote sites.

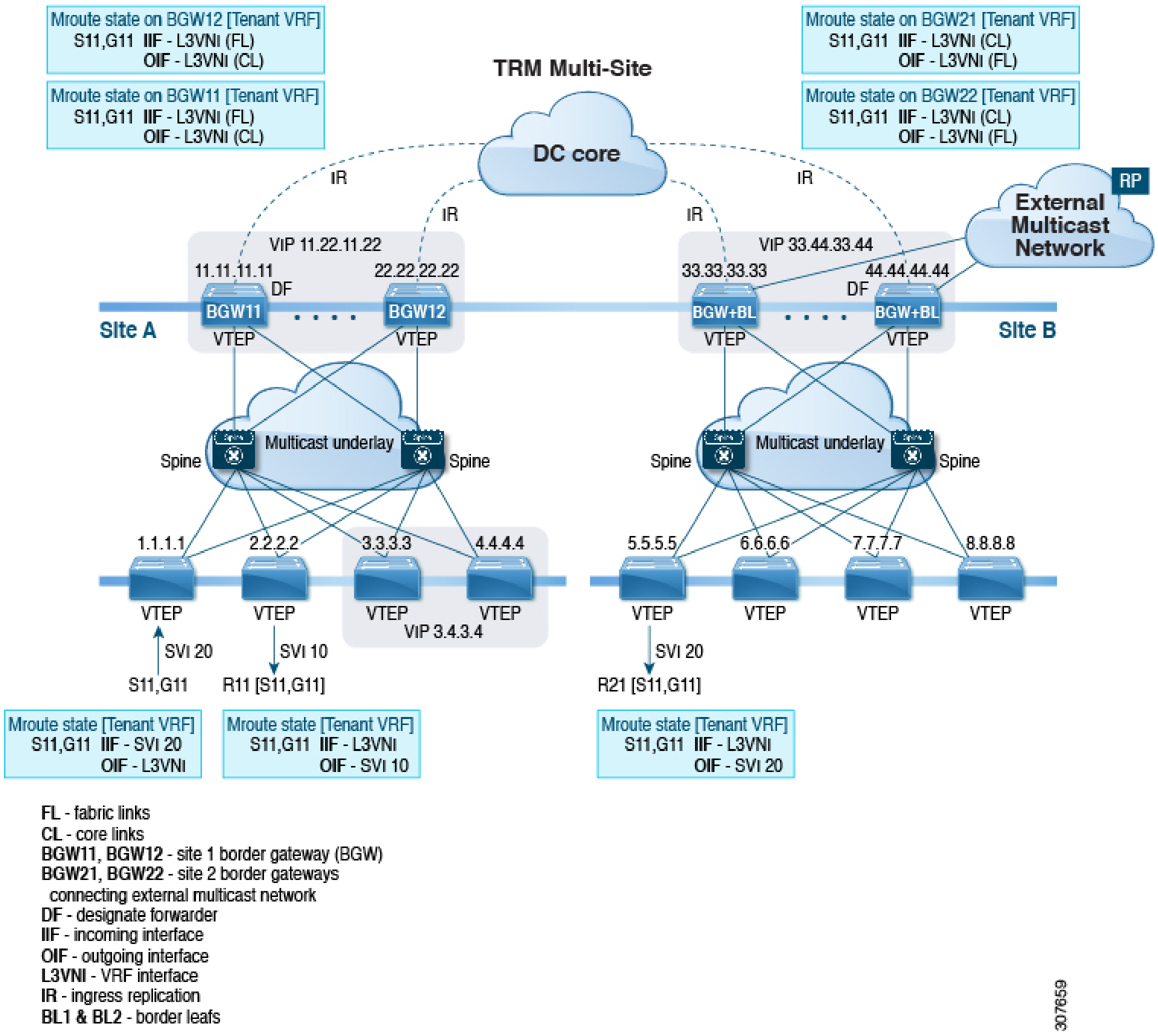
On the remote site, the BGW that receives the inter-site multicast traffic from the core forwards the traffic toward the fabric side. The DF check is not done from the core to fabric direction because non-DF can also receive the VIP-R copy from the source site.

Figure 25: TRM with Multi-Site Topology, BL External Multicast Connectivity



Beginning with Cisco NX-OS Release 9.3(3), TRM with Multi-Site supports BGW connections to the external multicast network in addition to the BL connectivity, which is supported in previous releases. Forwarding occurs as documented in the previous example, except the exit point to the external multicast network can optionally be provided through the BGW.

Figure 26: TRM with Multi-Site Topology, BGW External Multicast Connectivity



Guidelines and Limitations for TRM with Multi-Site

TRM with Multi-Site has the following guidelines and limitations:

- The following platforms support TRM with Multi-Site:
  - Cisco Nexus 9300-EX platform switches
  - Cisco Nexus 9300-FX/FX2/FX3 platform switches
  - Cisco Nexus 9300-GX platform switches
  - Cisco Nexus 9500 platform switches with -EX/FX line cards
- Beginning with Cisco NX-OS Release 9.3(3), a border leaf and Multi-Site border gateway can coexist on the same node for multicast traffic.

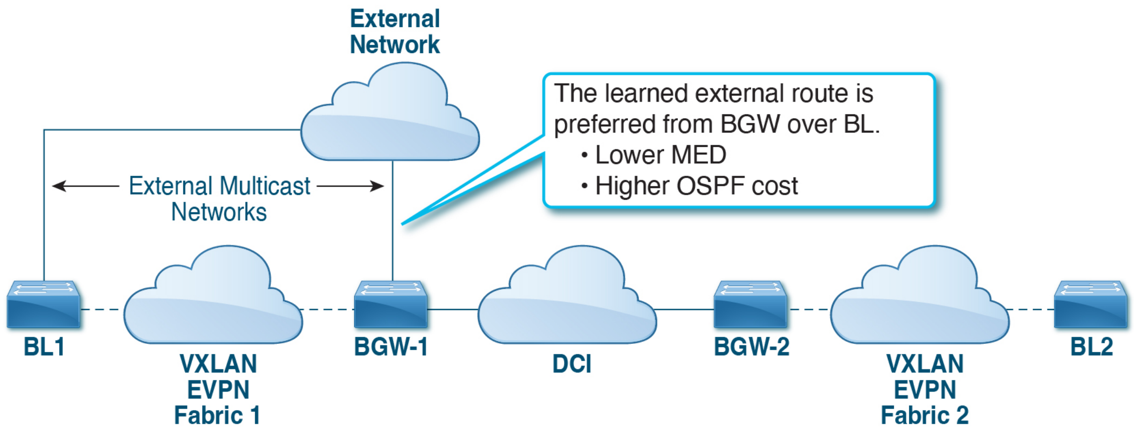
- Beginning with Cisco NX-OS Release 9.3(3), all border gateways for a given site must run the same Cisco NX-OS 9.3(x) image.
- Cisco NX-OS Release 10.1(2) has the following guidelines and limitations:
  - You need to add a VRF lite link (per Tenant VRF) between the vPC peers in order to support the L3 hosts attached to the vPC primary and secondary peers.
  - Backup SVI is needed between the two vPC peers.
  - Orphan ports attached with L2 and L3 are supported with vPC BGW.
  - TRM multi-site with vPC BGW is not supported with vMCT.

  For details on TRM and Configuring TRM with vPC Support, see Configuring Tenant Routed Multicast.
- TRM with Multi-Site supports the following features:
  - TRM Multi-Site with vPC Border Gateway.
  - PIM ASM multicast underlay in the VXLAN fabric
  - TRM with Multi-Site Layer 3 mode only
  - TRM with Multi-Site with Anycast Gateway
  - Terminating VRF-lite at the border leaf
  - The following RP models with TRM Multi-Site:
    - External RP
    - RP Everywhere
    - Internal RP
- Only one pair of vPC BGW can be configured on one site.
- A pair of vPC BGW and Anycast BGW cannot co-exist on the same site.
- Border routers reoriginate MVPN routes from fabric to core and from core to fabric.
- Only eBGP peering between border gateways of different sites is supported.
- Each site must have a local RP for the TRM underlay.
- Keep each site's underlay unicast routing isolated from another site's underlay unicast routing. This requirement also applies to Multi-Site.
- MVPN address family must be enabled between BGWs.
- When configuring BGW connections to the external multicast fabric, be aware of the following:
  - The multicast underlay must be configured between all BGWs on the fabric side even if the site doesn’t have any leafs in the fabric site.
  - Sources and receivers that are Layer-3 attached through VRF-Lite links to the BGW of a single site acting therefore also as Border Leaf (BL) node need to have reachability through the external Layer-3 network. If there's a Layer-3 attached source on BGW BL Node-1 and a Layer-3 attached receiver

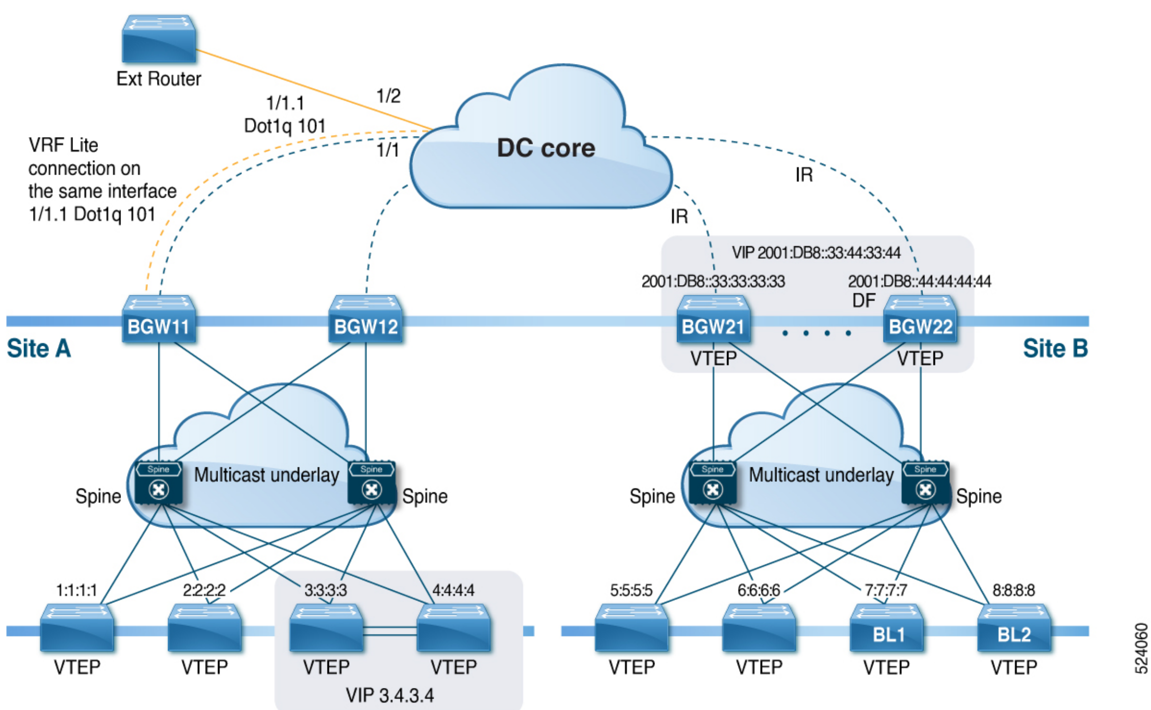
on BGW BL Node-2 for the same site, the traffic between these two endpoints flows through the external Layer-3 network and not through the fabric.

- External multicast networks should be connected only through the BGW or BL. If a deployment requires external multicast network connectivity from both the BGW and BL at the same site, make sure that external routes that are learned from the BGW are preferred over the BL. To do so, the BGW must have a lower MED and a higher OSPF cost (on the external links) than the BL.

  The following figure shows a site with external network connectivity through BGW-BLs and an internal leaf (BL1). The path to the external source should be through BGW-1 (rather than through BL1) to avoid duplication on the remote site receiver.



- The BGW supports VRF-lite hand-off and Multi-site configuration on the same physical interface as shown in the diagram.



- MED is supported for iBGP only.

Configuring TRM with Multi-Site

Before you begin

The following must be configured:

- VXLAN TRM
- VXLAN Multi-Site

This section provides the configuration procedure for Anycast BGW with TRM. For vPC BGW with TRM, vPC must be configured along with VxLAN TRM and VxLAN Multi-site.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**interface nve1** **Example:** `switch(config)# interface nve1`	Configures the NVE interface.
Step 3	**no shutdown** **Example:** `switch(config-if-nve)# no shutdown`	Brings up the NVE interface.
Step 4	**host-reachability protocol bgp** **Example:** `switch(config-if-nve)# host-reachability protocol bgp`	Defines BGP as the mechanism for host reachability advertisement.
Step 5	**source-interface loopback** *src-if* **Example:** `switch(config-if-nve)# source-interface loopback 0`	Defines the source interface, which must be a loopback interface with a valid /32 IP address. This /32 IP address must be known by the transient devices in the transport network and the remote VTEPs. This requirement is accomplished by advertising the address through a dynamic routing protocol in the transport network.
Step 6	**multisite border-gateway interface loopback** *vi-num* **Example:** `switch(config-if-nve)# multisite border-gateway interface loopback 1`	Defines the loopback interface used for the border gateway virtual IP address (VIP). The border-gateway 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 transient devices in the transport network and the remote VTEPs. This requirement is accomplished by advertising the address through a dynamic routing protocol

	Command or Action	Purpose
		in the transport network. This loopback must be different than the source interface loopback. The range of *vi-num* is from 0 to 1023.
Step 7	**member vni** *vni-range* **associate-vrf** **Example:** `switch(config-if-nve)# member vni 10010 associate-vrf`	Configures the virtual network identifier (VNI). The range for *vni-range* is from 1 to 16,777,214 The value of *vni-range* can be a single value like 5000 or a range like 5001-5008.
Step 8	**mcast-group** *ip-addr* **Example:** `switch(config-if-nve-vni)# mcast-group 225.0.0.1`	Configures the NVE multicast group IP prefix within the fabric.
Step 9	**multisite ingress-replication optimized** **Example:** `switch(config-if-nve-vni)# multisite ingress-replication optimized`	Defines the Multi-Site BUM replication method for extending the Layer 2 VNI.

Verifying TRM with Multi-Site Configuration

To display the status for the TRM with Multi-Site configuration, enter the following command:

Command	Purpose
show nve vni *virtual-network-identifier*	Displays the L3VNI. **Note** For this feature, optimized IR is the default setting for the Multi-Site extended L3VNI. MS-IR flag inherently means that it's MS-IR optimized.

Example of the **show nve vni** command:

For IPv4

```
switch(config)# show nve vni 51001
Codes: CP - Control Plane        DP - Data Plane
       UC - Unconfigured         SA - Suppress ARP
       SU - Suppress Unknown Unicast
       Xconn - Crossconnect
       MS-IR - Multisite Ingress Replication

Interface VNI      Multicast-group   State Mode Type [BD/VRF]      Flags
--------- -------- ----------------- ----- ---- ------------------ -----
nve1      51001    226.0.0.1         Up    CP   L3 [cust_1]        MS-IR
```

CHAPTER 12

Configuring Tenant Routed Multicast (TRM)

This chapter contains the following sections:

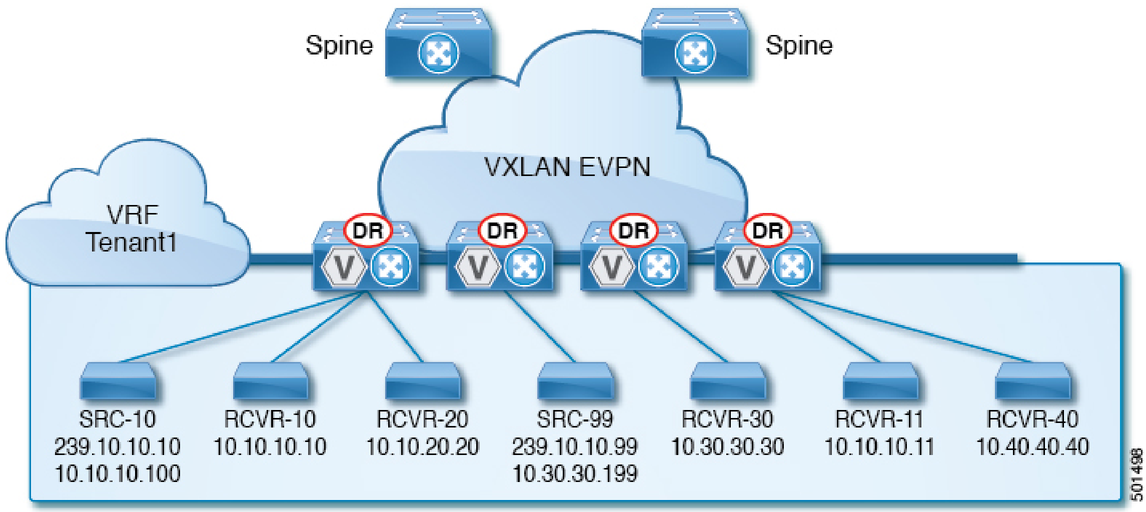
About Tenant Routed Multicast

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 27: VXLAN EVPN TRM

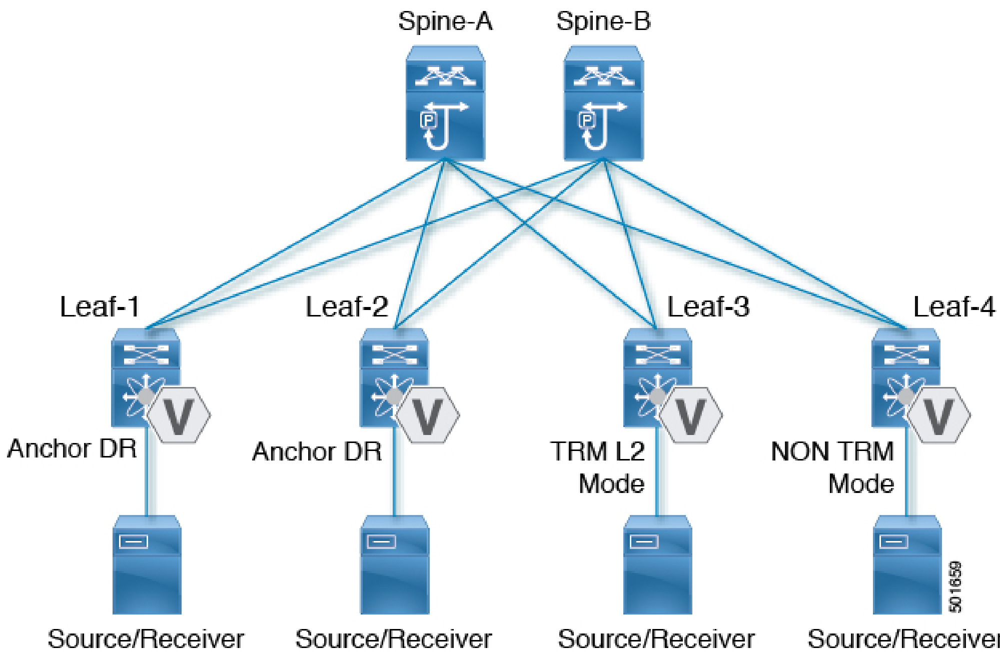


With TRM enabled, multicast forwarding in the underlay is leveraged to replicate VXLAN encapsulated routed multicast traffic. A Default Multicast Distribution Tree (Default-MDT) is built per-VRF. This is an addition to the existing multicast groups for Layer-2 VNI Broadcast, Unknown Unicast, and Layer-2 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.

About Tenant Routed Multicast Mixed Mode

Figure 28: TRM Layer 2/Layer 3 Mixed Mode



Guidelines and Limitations for Tenant Routed Multicast

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

- Beginning with Cisco NX-OS Release 10.1(2), TRM Multisite with vPC BGW is supported.
- With Tenant Routed Multicast enabled, FEX is not supported.
- If VXLAN TRM feature is enabled on a VTEP, it would stop to send IGMP messages to the VXLAN fabric.
- The Guidelines and Limitations for VXLAN also apply to TRM.
- With TRM enabled, SVI as a core link is not supported.
- If TRM is configured, ISSU is disruptive.
- 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.
- In a Transit Routing Multicast (TRM) deployment, the RP-on-stick model can sometimes lead to traffic drops if there is flapping on the Protocol Independent Multicast (PIM) enabled interface. Use the **ip pim spt-switch-graceful** command on the turnaround router that leads to the RP. This command allows for a graceful switch to the Shortest Path Tree (SPT) during flapping, which can minimize traffic drops.
- TRM with Multi-Site is not supported on Cisco Nexus 9504-R platforms.
- TRM supports multiple border nodes. Reachability to an external RP/source via multiple border leaf switches is supported with ECMP and requires symmetric unicast routing.
- Both PIM and **ip igmp snooping vxlan** must be enabled on the L3 VNI's VLAN in a VXLAN vPC setup.
- For traffic streams with an internal source and external L3 receiver using an external RP, the external L3 receiver might send PIM S,G join requests to the internal source. Doing so triggers the recreation of S,G on the fabric FHR, and it can take up to 10 minutes for this S,G to be cleared.

Guidelines and Limitations for Layer 3 Tenant Routed Multicast

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

- When upgrading from Cisco NX-OS Release 9.3(3) to Cisco NX-OS Release 9.3(6), if you do not retain configurations of the TRM enabled VRFs from Cisco NX-OS Release 9.3(3), or if you create new VRFs after the upgrade, the auto-generation of **ip multicast multipath s-g-hash next-hop-based** CLI, when **feature ngmvpn** is enabled, will not happen. You must enable the CLI manually for each TRM enabled VRF.
- Layer 3 TRM is supported for Cisco Nexus 9200, 9300-EX, and 9300-FX/FX2/FX3/FXP and 9300-GX platform switches.
- Beginning with Cisco NX-OS Release 9.3(7), Cisco Nexus N9K-C9316D-GX, N9K-C9364C-GX, and N9K-X9716D-GX platform switches support the combination of Layer 3 TRM and EVPN Multi-Site.
- Cisco Nexus 9300-GX platform switches do not support the combination of Layer 3 TRM and EVPN Multi-Site in Cisco NX-OS Release 9.3(5).
- Beginning with Cisco NX-OS Release 9.3(3), the Cisco Nexus 9504 and 9508 platform switches with -R/RX line cards support TRM in Layer 3 mode. This feature is supported on IPv4 overlays only. Layer 2 mode and L2/L3 mixed mode are not supported.

  The Cisco Nexus 9504 and 9508 platform switches with -R/RX line cards can function as a border leaf for Layer 3 unicast traffic. For Anycast functionality, the RP can be internal, external, or RP everywhere.
- When configuring TRM VXLAN BGP EVPN, the following platforms are supported:
  - Cisco Nexus 9200, 9332C, 9364C, 9300-EX, and 9300-FX/FX2/FX3/FXP platform switches.
  - Cisco Nexus 9500 platform switches with 9700-EX line cards, 9700-FX line cards, or a combination of both line cards.

- Layer 3 TRM and VXLAN EVPN Multi-Site are supported on the same physical switch. For more information, see Configuring Multi-Site.
- TRM Multi-Site functionality is not supported on Cisco Nexus 9504 platform switches with -R/RX line cards.
- If one or both VTEPs is a Cisco Nexus 9504 or 9508 platform switch with -R/RX line cards, the packet TTL is decremented twice, once for routing to the L3 VNI on the source leaf and once for forwarding from the destination L3 VNI to the destination VLAN on the destination leaf.
- TRM with vPC border leafs is supported only for Cisco Nexus 9200, 9300-EX, and 9300-FX/FX2/FX3/GX platform switches and Cisco Nexus 9500 platform switches with -EX/FX or -R/RX line cards. The **advertise-pip** and **advertise virtual-rmac** commands must be enabled on the border leafs to support this functionality. For configuration information, see the "Configuring VIP/PIP" section.
- 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.

Guidelines and Limitations for Layer 2/Layer 3 Tenant Routed Multicast (Mixed Mode)

Layer 2/Layer 3 Tenant Routed Multicast (TRM) has the following configuration guidelines and limitations:

- All TRM Layer 2/Layer 3 configured switches must be Anchor DR. This is because in TRM Layer 2/Layer 3, you can have switches configured with TRM Layer 2 mode that co-exist in the same topology. This mode is necessary if non-TRM and Layer 2 TRM mode edge devices (VTEPs) are present in the same topology.
- Anchor DR is required to be an RP in the overlay.
- An extra loopback is required for anchor DRs.
- Non-TRM and Layer 2 TRM mode edge devices (VTEPs) require an IGMP snooping querier configured per multicast-enabled VLAN. Every non-TRM and Layer 2 TRM mode edge device (VTEP) requires this IGMP snooping querier configuration because in TRM multicast control-packets are not forwarded over VXLAN.
- The IP address for the IGMP snooping querier can be re-used on non-TRM and Layer 2 TRM mode edge devices (VTEPs).
- The IP address of the IGMP snooping querier in a VPC domain must be different on each VPC member device.
- When interface NVE is brought down on the border leaf, the internal overlay RP per VRF should be brought down.
- The NVE interface must be shut and unshut while configuring the **ip multicast overlay-distributed-dr** command.
- Beginning with Cisco NX-OS Release 9.2(1), TRM with vPC border leafs is supported. Advertise-PIP and Advertise Virtual-Rmac need to be enabled on border leafs to support with functionality. For configuring advertise-pip and advertise virtual-rmac, see the "Configuring VIP/PIP" section.

- Anchor DR is supported only on the following hardware platforms:
  - Cisco Nexus 9200, 9300-EX, and 9300-FX/FX2 platform switches
  - Cisco Nexus 9500 platform switches with 9700-EX line cards, 9700-FX line cards, or a combination of both line cards
- Layer 2/Layer 3 Tenant Routed Multicast (TRM) is not supported on Cisco Nexus 9300-FX3/GX platform switches.

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 L2 Mode	N/A	N/A	N/A
TRM L3 Mode	7.0(3)I7(1), 9.2(x)	7.0(3)I7(4), 9.2(3)	Supported in 7.0(3)I7(x) releases starting from 7.0(3)I7(5) Not supported in 9.2(x) Supported in NX-OS releases beginning with 9.3(1) for the following Nexus 9000 switches: • Cisco Nexus 9200 Series switches • Cisco Nexus 9364C platform switches • Cisco Nexus 9300-EX/FX/FX2 platform switches (excluding the Cisco Nexus 9300-FXP platform switch) Supported for Cisco Nexus 9300-FX3 platform switches beginning with Cisco NX-OS Release 9.3(5)
TRM L2L3 Mode	7.0(3)I7(1), 9.2(x)	N/A	N/A

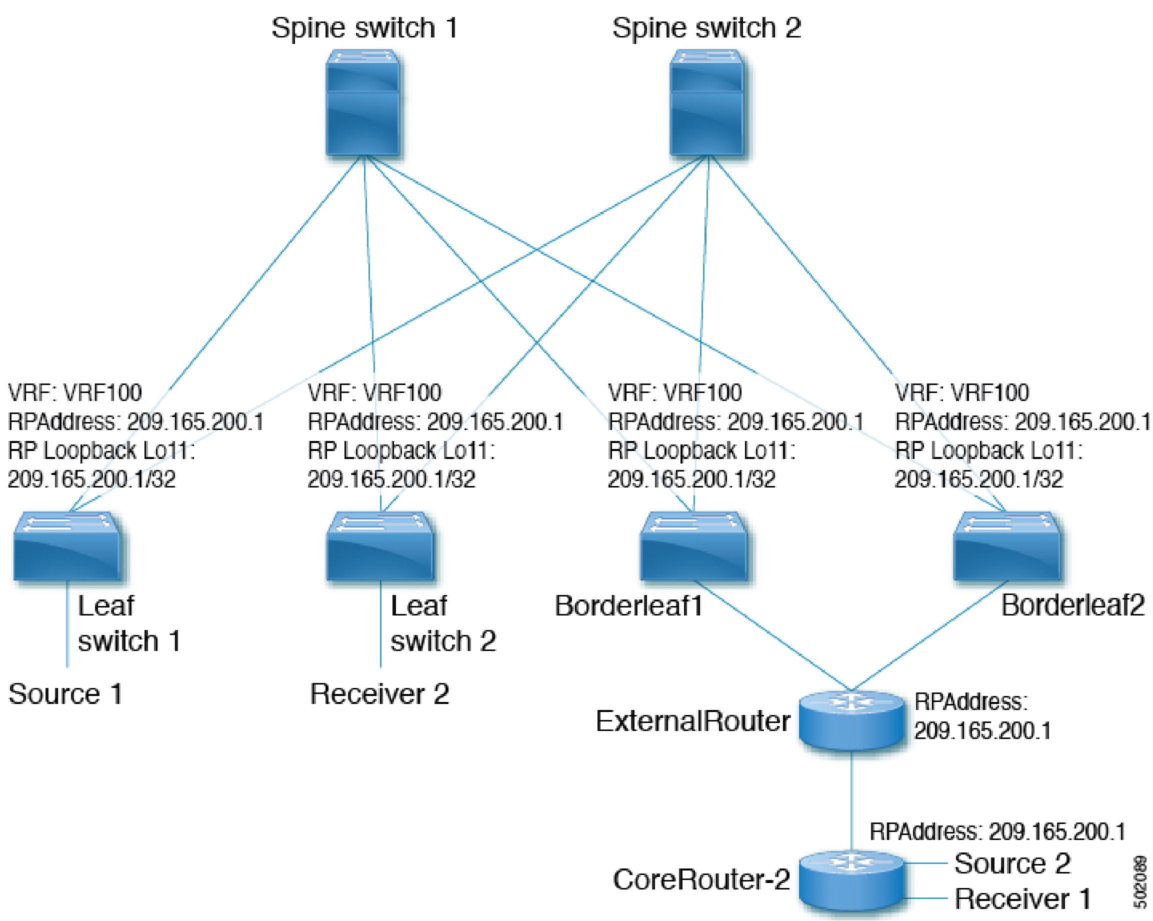
Configuring a Rendezvous Point for Tenant Routed Multicast

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

- Configuring a Rendezvous Point Inside the VXLAN Fabric, on page 255
- Configuring an External Rendezvous Point, on page 256
- Configuring RP Everywhere with PIM Anycast, on page 258
- Configuring RP Everywhere with MSDP Peering, on page 264

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).



SUMMARY STEPS

1. **configure terminal**

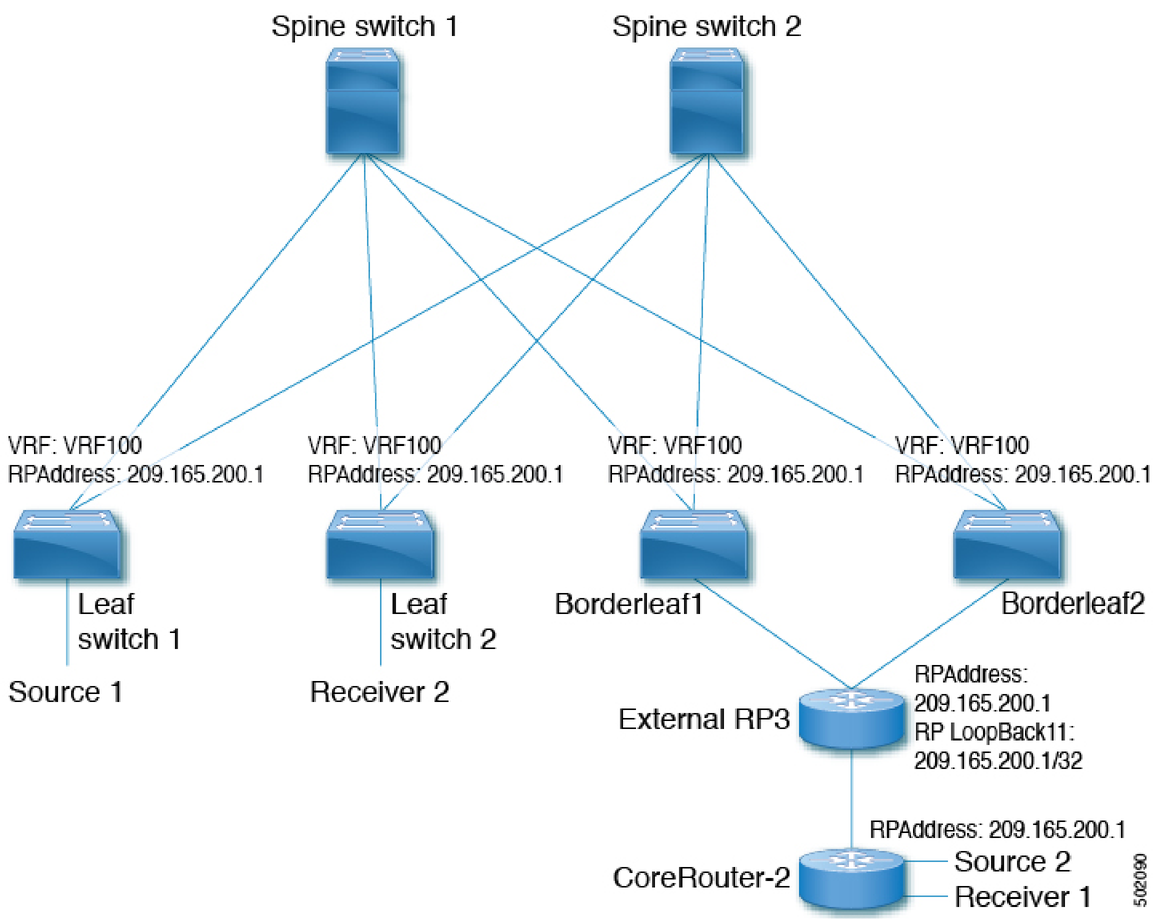
2. **interface loopback** *loopback_number*
3. **vrf member** *vxlan-number*
4. **ip address** *ip-address*
5. **ip pim sparse-mode**
6. **vrf context** *vrf-name*
7. **ip pim rp-address** *ip-address-of-router* **group-list** *group-range-prefix*

DETAILED STEPS

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



SUMMARY STEPS

1. **configure terminal**
2. **vrf context vrf100**
3. **ip pim rp-address** *ip-address-of-router* **group-list** *group-range-prefix*

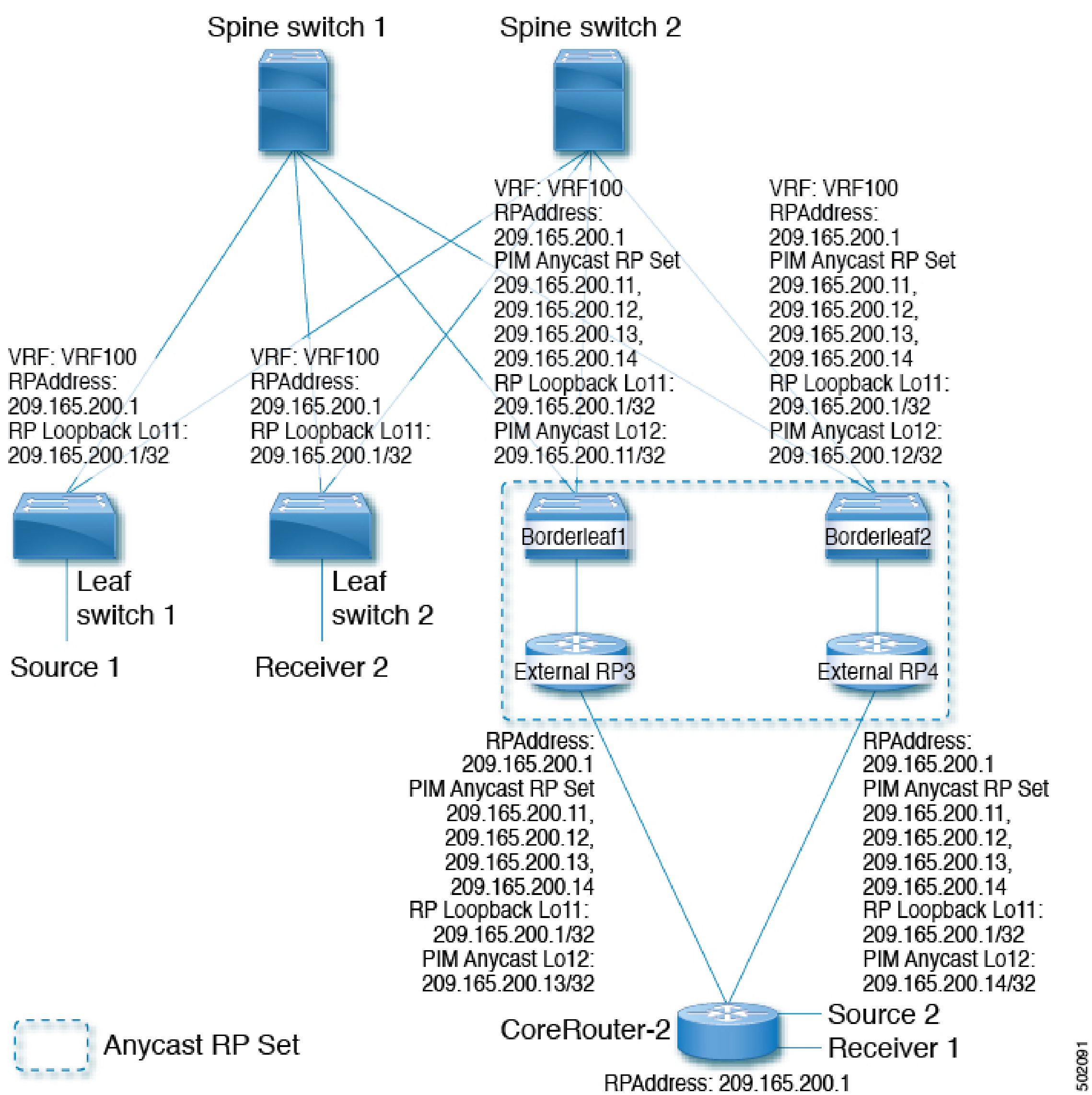
DETAILED STEPS

	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.

	Command or Action	Purpose
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

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

SUMMARY STEPS

1. **configure terminal**
2. **interface loopback** *loopback_number*
3. **vrf member** *vrf-name*
4. **ip address** *ip-address*
5. **ip pim sparse-mode**
6. **vrf context** *vxlan*
7. **ip pim rp-address** *ip-address-of-router* **group-list** *group-range-prefix*

DETAILED STEPS

	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.

	Command or Action	Purpose
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.

SUMMARY STEPS

1. **configure terminal**
2. **{ip | ipv6} pim evpn-border-leaf**
3. **interface loopback** *loopback_number*
4. **vrf member** *vrf-name*
5. **ip address** *ip-address*
6. **ipv6 pim sparse-mode**
7. **interface loopback** *loopback_number*
8. **vrf member** *vxlan-number*
9. **ipv6 address** *ipv6-address*
10. **ipv6 pim sparse-mode**
11. **vrf context** *vrf-name*
12. **ipv6 pim rp-address** *ipv6-address-of-router* **group-list** *group-range-prefix*
13. **ipv6 pim anycast-rp** *anycast-rp-address address-of-rp*
14. **ipv6 pim anycast-rp** *anycast-rp-address address-of-rp*
15. **ipv6 pim anycast-rp** *anycast-rp-address address-of-rp*
16. **ipv6 pim anycast-rp** *anycast-rp-address address-of-rp*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enter configuration mode.
Step 2	**{ip | ipv6} pim evpn-border-leaf** **Example:** `switch(config)# ipv6 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.

	Command or Action	Purpose
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	**ipv6 pim sparse-mode** **Example:** `switch(config-if)# ipv6 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	**ipv6 address** *ipv6-address* **Example:** `switch(config-if)# ip address 209.165.200.11/32`	Specify IP address.
Step 10	**ipv6 pim sparse-mode** **Example:** `switch(config-if)# ipv6 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	**ipv6 pim rp-address** *ipv6-address-of-router* **group-list** *group-range-prefix* **Example:** `switch(config-vrf)# ipv6 pim rp-address 2090:165:200::1 group ff1e::/16`	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	**ipv6 pim anycast-rp** *anycast-rp-address address-of-rp* **Example:** `switch(config-vrf)# ipv6 pim anycast-rp 2090:165:2000::1 2090:165:2000::11`	Configure PIM Anycast RP set.
Step 14	**ipv6 pim anycast-rp** *anycast-rp-address address-of-rp* **Example:**	Configure PIM Anycast RP set.

	Command or Action	Purpose
	`switch(config-vrf)# ipv6 pim anycast-rp 2090:165:2000::1 2090:165:2000::12`	
Step 15	**ipv6 pim anycast-rp** *anycast-rp-address address-of-rp* **Example:** `switch(config-vrf)# ipv6 pim anycast-rp 2090:165:2000::1 2090:165:2000::13`	Configure PIM Anycast RP set.
Step 16	**ipv6 pim anycast-rp** *anycast-rp-address address-of-rp* **Example:** `switch(config-vrf)# ipv6 pim anycast-rp 2090:165:2000::1 2090:165:2000::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.

SUMMARY STEPS

1. **configure terminal**
2. **interface loopback** *loopback_number*
3. **vrf member** *vrf-name*
4. **ip address** *ip-address*
5. **ip pim sparse-mode**
6. **interface loopback** *loopback_number*
7. **vrf member** *vxlan-number*
8. **ip address** *ip-address*
9. **ip pim sparse-mode**
10. **vrf context** *vxlan*
11. **ip pim rp-address** *ip-address-of-router* **group-list** *group-range-prefix*
12. **ip pim anycast-rp** *anycast-rp-address address-of-rp*
13. **ip pim anycast-rp** *anycast-rp-address address-of-rp*
14. **ip pim anycast-rp** *anycast-rp-address address-of-rp*
15. **ip pim anycast-rp** *anycast-rp-address address-of-rp*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enter configuration mode.
Step 2	**interface loopback** *loopback_number* **Example:**	Configure the loopback interface on all VXLAN VTEP devices.

	Command or Action	Purpose
	switch(config)# **interface loopback 11**	
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.
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.

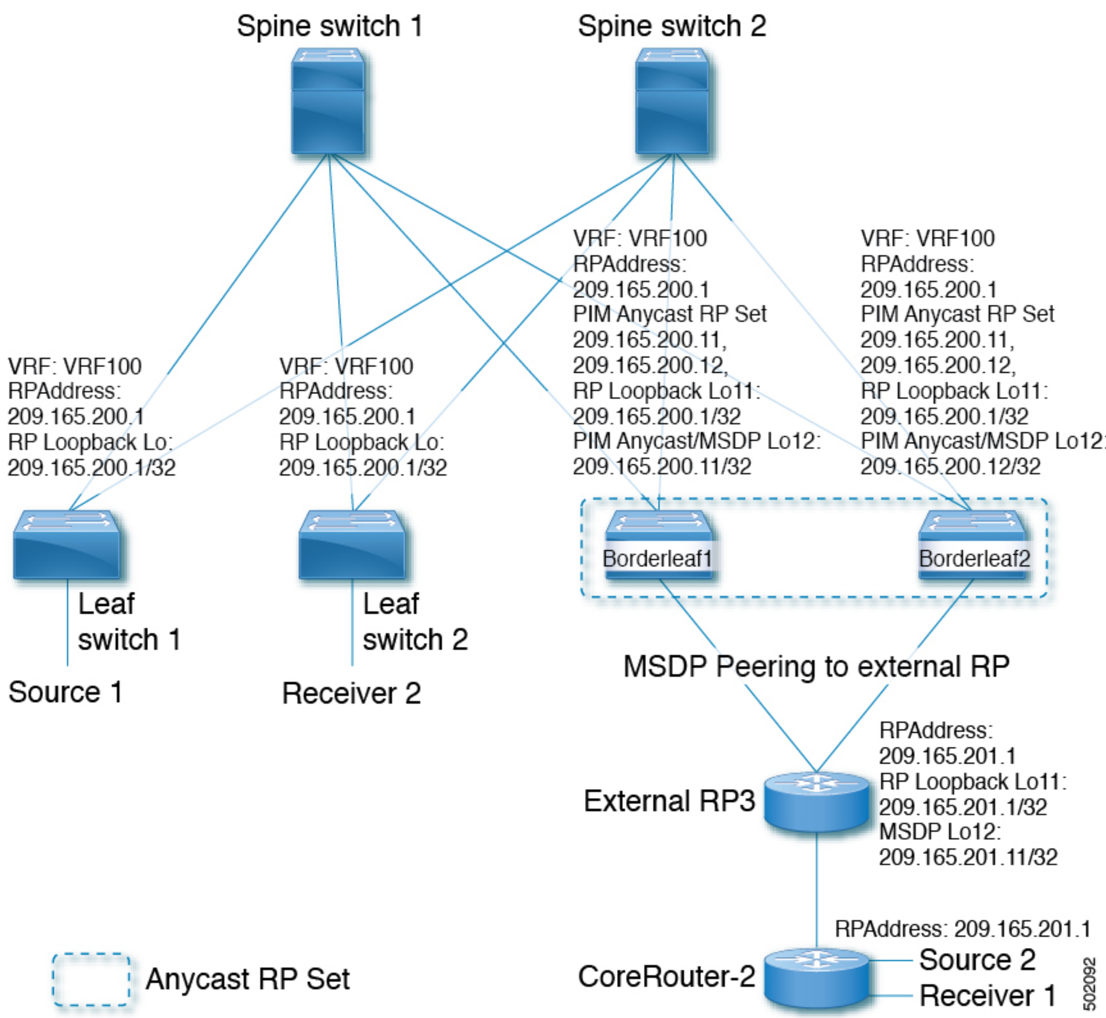
	Command or Action	Purpose
Step 13	**ip pim anycast-rp** *anycast-rp-address address-of-rp* **Example:** `switch(config-vrf)# ip pim anycast-rp 209.165.200.1 209.165.200.12`	Configure PIM Anycast RP set.
Step 14	**ip pim anycast-rp** *anycast-rp-address address-of-rp* **Example:** `switch(config-vrf)#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

The following figure represents the 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 265
- Configuring a TRM Border Leaf Node for RP Everywhere with MSDP Peering, on page 266
- Configuring an External Router for RP Everywhere with MSDP Peering, on page 269



Configuring a TRM Leaf Node for RP Everywhere with MSDP Peering

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

SUMMARY STEPS

1. **configure terminal**
2. **interface loopback** *loopback_number*
3. **vrf member** *vrf-name*
4. **ip address** *ip-address*
5. **ip pim sparse-mode**
6. **vrf context** *vrf-name*
7. **ip pim rp-address** *ip-address-of-router* **group-list** *group-range-prefix*

DETAILED STEPS

	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** *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.

SUMMARY STEPS

1. **configure terminal**
2. **feature msdp**
3. **ip pim evpn-border-leaf**
4. **interface loopback** *loopback_number*
5. **vrf member** *vrf-name*
6. **ip address** *ip-address*
7. **ip pim sparse-mode**

8. **interface loopback** *loopback_number*
9. **vrf member** *vrf-name*
10. **ip address** *ip-address*
11. **ip pim sparse-mode**
12. **vrf context** *vrf-name*
13. **ip pim rp-address** *ip-address-of-router* **group-list** *group-range-prefix*
14. **ip pim anycast-rp** *anycast-rp-address address-of-rp*
15. **ip pim anycast-rp** *anycast-rp-address address-of-rp*
16. **ip msdp originator-id** *loopback*
17. **ip msdp peer** *ip-address* **connect-source** *loopback*

DETAILED STEPS

	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:** `switch(config)# ip pim evpn-border-leaf`	Configure VXLAN VTEP as TRM border leaf node,
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.

	Command or Action	Purpose
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.
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

SUMMARY STEPS

1. **configure terminal**
2. **feature msdp**
3. **interface loopback** *loopback_number*
4. **vrf member** *vrf-name*
5. **ip address** *ip-address*
6. **ip pim sparse-mode**
7. **interface loopback** *loopback_number*
8. **vrf member** *vrf-name*
9. **ip address** *ip-address*
10. **ip pim sparse-mode**
11. **vrf context** *vrf-name*
12. **ip pim rp-address** *ip-address-of-router* **group-list** *group-range-prefix*
13. **ip msdp originator-id loopback12**
14. **ip msdp peer** *ip-address* **connect-source loopback12**

DETAILED STEPS

	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.
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.

	Command or Action	Purpose
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:** `switch(config-vrf)# ip msdp peer 209.165.200.11 connect-source loopback12`	Configure MSDP peering between external RP router and all TRM border nodes.

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. You will get a syslog message when you enable this command. The message informs you that **ip multicast multipath s-g-hash next-hop-based** is the recommended multipath hashing algorithm and you need enable it for the TRM enabled VRFs. The auto-generation of **ip multicast multipath s-g-hash next-hop-based** command does not happen after you enable the **feature ngmvpn** command. You need to configure **ip multicast multipath s-g-hash next-hop-based** as part of the VRF configuration.
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.
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:**	Builds the default multicast distribution tree for the VRF VNI (Layer 3 VNI).

	Command or Action	Purpose
	`switch(config-if-nve-vni)# **mcast-group 225.3.3.3**`	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 <as-number>** **Example:** `switch(config)# **router bgp 100**`	Set autonomous system number.
Step 10	**neighbor** *ip-addr* **Example:** `switch(config-router)# **neighbor 1.1.1.1**`	Configure IP address of the neighbor.
Step 11	**address-family ipv4 mvpn** **Example:** `switch(config-router-neighbor)# **address-family ipv4 mvpn**`	Configure multicast VPN.
Step 12	**send-community extended** **Example:** `switch(config-router-neighbor-af)# **send-community extended**`	Enables ngMVPN for address family signalization. The **send community extended** command ensures that extended communities are exchanged for this address family.
Step 13	**exit** **Example:** `switch(config-router-neighbor-af)# **exit**`	Exits command mode.
Step 14	**exit** **Example:** `switch(config-router)# **exit**`	Exits command mode.
Step 15	**vrf context** *vrf_name* **Example:** `switch(config-router)# **vrf context vrf100**`	Configures VRF name.

	Command or Action	Purpose
Step 16	**ip multicast multipath s-g-hash next-hop-based** **Example:** `switch(config-vrf)# ip multicast multipath s-g-hash next-hop-based`	Configures multicast multipath and initiates S, G, nexthop hashing (rather than the default of S/RP, G-based hashing) to select the RPF interface.
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 255 section.
Step 18	**address-family ipv4 unicast** **Example:** `switch(config-vrf)# address-family ipv4 unicast`	Configures 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 9000 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.
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`	Configures VRF name.
Step 24	**ipv6 address** *ipv6_address* **Example:** `switch(config-if)# ip address 11.1.1.1/24`	Configures IP address.
Step 25	**ipv6 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.

	Command or Action	Purpose
Step 26	**fabric forwarding mode anycast-gateway** **Example:** `switch(config-if)# fabric forwarding mode anycast-gateway`	Configures Anycast Gateway Forwarding Mode.
Step 27	**ip pim neighbor-policy** *route-map-name* **Example:** `switch(config-if)# ip pim neighbor-policy route-map1`	Creates an IP PIM neighbor policy with a suitable route-map to deny any IPv4 addresses, preventing PIM from establishing PIM neighborship on the L2VNI SVI. **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`	Configures Layer 3 VNI.
Step 30	**no shutdown** **Example:** `switch(config-if)# no shutdown`	Disable an interface.
Step 31	**vrf member vrf100** **Example:** `switch(config-if)# vrf member vrf100`	Configures VRF name.
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`	Configures 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. See Configuring iBGP for EVPN on the Spine, on page 92.

SUMMARY STEPS

1. **configure terminal**
2. **route-map permitall permit 10**
3. **set ip next-hop unchanged**
4. **exit**
5. **router bgp [autonomous system]** *number*
6. **address-family ipv4 mvpn**
7. **retain route-target all**
8. **neighbor** *ip-address* [**remote-as** *number*]
9. **address-family ipv4 mvpn**
10. **disable-peer-as-check**
11. **rewrite-rt-asn**
12. **send-community extended**
13. **route-reflector-client**
14. **route-map permitall out**

DETAILED STEPS

	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.
Step 5	**router bgp [autonomous system]** *number* **Example:** `switch(config)# router bgp 65002`	Specify BGP.

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

	Command or Action	Purpose
	switch(config-router-neighbor-af)# **route-map permitall out**	

Configuring Tenant Routed Multicast in Layer 2/Layer 3 Mixed Mode

This procedure enables the Tenant Routed Multicast (TRM) feature. This enables both Layer 2 and Layer 3 multicast BGP signaling. This mode is only necessary if non-TRM edge devices (VTEPs) are present in the Cisco Nexus 9000 Series switches (1st generation). Only the Cisco Nexus 9000-EX and 9000-FX switches can do Layer 2/Layer 3 mode (Anchor-DR).

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.

All Cisco Nexus 9300-EX and 9300-FX platform switches must be in Layer 2/Layer 3 mode.

Before you begin

VXLAN 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	**advertise evpn multicast** **Example:** switch(config)# **advertise evpn multicast**	Advertises IMET and SMET routes into BGP EVPN towards non-TRM capable switches.
Step 4	**ip igmp snooping vxlan** **Example:** switch(config)# **ip igmp snooping vxlan**	Configure IGMP snooping for VXLAN VLANs.
Step 5	**ip multicast overlay-spt-only** **Example:** switch(config)# **ip multicast overlay-spt-only**	Gratuitously originate (S,A) route when source is locally connected. The **ip multicast overlay-spt-only** command is enabled by default on all MVPN-enabled Cisco Nexus 9000 Series switches (typically leaf nodes).

	Command or Action	Purpose
Step 6	**ip multicast overlay-distributed-dr** **Example:** `switch(config)# ip multicast overlay-distributed-dr`	Enables distributed anchor DR function on this VTEP. **Note** The NVE interface must be shut and unshut while configuring this command.
Step 7	**interface nve1** **Example:** `switch(config)# interface nve 1`	Configure the NVE interface.
Step 8	**[no] shutdown** **Example:** `switch(config-if-nve)# shutdown`	Shuts down the NVE interface. The **no shutdown** command brings up the interface.
Step 9	**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 10	**mcast-group** *ip-prefix* **Example:** `switch(config-if-nve-vni)# mcast-group 225.3.3.3`	Configures the multicast group on distributed anchor DR.
Step 11	**exit** **Example:** `switch(config-if-nve-vni)# exit`	Exits command mode.
Step 12	**interface loopback** *loopback_number* **Example:** `switch(config-if-nve)# interface loopback 10`	Configure the loopback interface on all distributed anchor DR devices.
Step 13	**ip address** *ip_address* **Example:** `switch(config-if)# ip address 100.100.1.1/32`	Configure IP address. This IP address is the same on all distributed anchor DR.
Step 14	**ip router ospf** *process-tag* **area** *ospf-id* **Example:** `switch(config-if)# ip router ospf 100 area 0.0.0.0`	OSPF area ID in IP address format.
Step 15	**ip pim sparse-mode** **Example:** `switch(config-if)# ip pim sparse-mode`	Configure sparse-mode PIM on interface.
Step 16	**interface nve1** **Example:** `switch(config-if)# interface nve1`	Configure NVE interface.

	Command or Action	Purpose
Step 17	**shutdown** **Example:** `switch(config-if-nve)# shutdown`	Disable the interface.
Step 18	**mcast-routing override source-interface loopback** *int-num* **Example:** `switch(config-if-nve)# mcast-routing override source-interface loopback 10`	Enables that TRM is using a different loopback interface than the VTEPs default source-interface. The *loopback10* variable must be configured on every TRM-enabled VTEP (Anchor DR) in the underlay with the same IP address. This loopback and the respective **override** command are needed to serve TRM VTEPs in co-existence with non-TRM VTEPs.
Step 19	**exit** **Example:** `switch(config-if-nve)# exit`	Exits command mode.
Step 20	**router bgp 100** **Example:** `switch(config)# router bgp 100`	Set autonomous system number.
Step 21	**neighbor** *ip-addr* **Example:** `switch(config-router)# neighbor 1.1.1.1`	Configure IP address of the neighbor.
Step 22	**address-family ipv4 mvpn** **Example:** `switch(config-router-neighbor)# address-family ipv4 mvpn`	Configure multicast VPN.
Step 23	**send-community extended** **Example:** `switch(config-router-neighbor-af)# send-community extended`	Send community attribute.
Step 24	**exit** **Example:** `switch(config-router-neighbor-af)# exit`	Exits command mode.
Step 25	**exit** **Example:** `switch(config-router)# exit`	Exits command mode.
Step 26	**vrf** *vrf_name* **vrf100** **Example:** `switch(config)# vrf context vrf100`	Configure VRF name.

	Command or Action	Purpose
Step 27	**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 255 - Internal RP section.
Step 28	**address-family ipv4 unicast** **Example:** `switch(config-vrf)# address-family ipv4 unicast`	Configure unicast address family.
Step 29	**route-target both auto mvpn** **Example:** `switch(config-vrf-af-ipv4)# route-target both auto mvpn`	Specify target for mvpn routes.
Step 30	**exit** **Example:** `switch(config-vrf-af-ipv4)# exit`	Exits command mode.
Step 31	**exit** **Example:** `switch(config-vrf)# exit`	Exits command mode.
Step 32	**interface** *vlan_id* **Example:** `switch(config)# interface vlan11`	Configure Layer 2 VNI.
Step 33	**no shutdown** **Example:** `switch(config-if)# no shutdown`	Disable an interface.
Step 34	**vrf member vrf100** **Example:** `switch(config-if)# vrf member vrf100`	Configure VRF name.
Step 35	**ip address** *ip_address* **Example:** `switch(config-if)# ip address 11.1.1.1/24`	Configure IP address.
Step 36	**ip pim sparse-mode** **Example:** e `switch(config-if)# ip pim sparse-mode`	Configure sparse-mode PIM on the interface.

	Command or Action	Purpose
Step 37	**fabric forwarding mode anycast-gateway** **Example:** `switch(config-if)# fabric forwarding mode anycast-gateway`	Configure Anycast Gateway Forwarding Mode.
Step 38	**ip pim neighbor-policy** *route-map-name* **Example:** `switch(config-if)# ip pim neighbor-policy route-map1`	Creates an IP PIM neighbor policy with a suitable route-map to deny any IPv4 addresses, preventing PIM from establishing PIM neighborship on the L2VNI SVI.
Step 39	**exit** **Example:** `switch(config-if)# exit`	Exits command mode.
Step 40	**interface** *vlan_id* **Example:** `switch(config)# interface vlan100`	Configure Layer 3 VNI.
Step 41	**no shutdown** **Example:** `switch(config-if)# no shutdown`	Disable an interface.
Step 42	**vrf member vrf100** **Example:** `switch(config-if)# vrf member vrf100`	Configure VRF name.
Step 43	**ip forward** **Example:** `switch(config-if)# ip forward`	Enable IP forwarding on interface.
Step 44	**ip pim sparse-mode** **Example:** `switch(config-if)# ip pim sparse-mode`	Configure sparse-mode PIM on the interface.

Configuring Layer 2 Tenant Routed Multicast

This procedure enables the Tenant Routed Multicast (TRM) feature. This enables Layer 2 multicast BGP signaling.

IGMP Snooping Querier must be configured per multicast-enabled VXLAN VLAN on all Layer-2 TRM leaf switches.

Before you begin

VXLAN EVPN 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 EVPN/MVPN feature.
Step 3	**advertise evpn multicast** **Example:** `switch(config)# advertise evpn multicast`	Advertise L2 multicast capability.
Step 4	**ip igmp snooping vxlan** **Example:** `switch(config)# ip igmp snooping vxlan`	Configure IGMP snooping for VXLANs.
Step 5	**vlan configuration** *vlan-id* **Example:** `switch(config)# vlan configuration 101`	Enter configuration mode for VLAN 101.
Step 6	**ip igmp snooping querier** *querier-ip-address* **Example:** `switch(config-vlan-config)# ip igmp snooping querier 2.2.2.2`	Configure IGMP snooping querier for each multicast-enabled VXLAN VLAN.

Configuring TRM with vPC Support

This section provides steps to configure TRM with vPC support. Beginning with Cisco NX-OS Release 10.1(2), TRM Multisite with vPC BGW is supported.

SUMMARY STEPS

1. **configure terminal**
2. **feature vpc**
3. **feature interface-vlan**
4. **feature lacp**
5. **feature pim**
6. **feature ospf**
7. **ip pim rp-address** *address* **group-list** *range*
8. **vpc domain** *domain-id*
9. **peer switch**
10. **peer gateway**

11. **peer-keepalive destination** *ipaddress*
12. **ip arp synchronize**
13. **ipv6 nd synchronize**
14. Create vPC peer-link.
15. **system nve infra-vlans** *range*
16. **vlan** *number*
17. Create the SVI.
18. (Optional) **delay restore interface-vlan** *seconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enter global configuration mode.
Step 2	**feature vpc** **Example:** `switch(config)# feature vpc`	Enables vPCs on the device.
Step 3	**feature interface-vlan** **Example:** `switch(config)# feature interface-vlan`	Enables the interface VLAN feature on the device.
Step 4	**feature lacp** **Example:** `switch(config)# feature lacp`	Enables the LACP feature on the device.
Step 5	**feature pim** **Example:** `switch(config)# feature pim`	Enables the PIM feature on the device.
Step 6	**feature ospf** **Example:** `switch(config)# feature ospf`	Enables the OSPF feature on the device.
Step 7	**ip pim rp-address** *address* **group-list** *range* **Example:** `switch(config)# ip pim rp-address 100.100.100.1 group-list 224.0.0/4`	Defines a PIM RP address for the underlay multicast group range.
Step 8	**vpc domain** *domain-id* **Example:** `switch(config)# vpc domain 1`	Creates a vPC domain on the device and enters vpn-domain configuration mode for configuration purposes. There is no default. The range is from 1 to 1000.

	Command or Action	Purpose
Step 9	**peer switch** **Example:** `switch(config-vpc-domain)# peer switch`	Defines the peer switch.
Step 10	**peer gateway** **Example:** `switch(config-vpc-domain)# peer gateway`	To enable Layer 3 forwarding for packets destined to the gateway MAC address of the virtual port channel (vPC), use the **peer-gateway** command.
Step 11	**peer-keepalive destination** *ipaddress* **Example:** `switch(config-vpc-domain)# peer-keepalive destination 172.28.230.85`	Configures the IPv4 address for the remote end of the vPC peer-keepalive link. **Note** The system does not form the vPC peer link until you configure a vPC peer-keepalive link. The management ports and VRF are the defaults. **Note** We recommend that you configure a separate VRF and use a Layer 3 port from each vPC peer device in that VRF for the vPC peer-keepalive link. For more information about creating and configuring VRFs, see the Cisco Nexus 9000 NX-OS Series Unicast Routing Config Guide, 9.3(x).
Step 12	**ip arp synchronize** **Example:** `switch(config-vpc-domain)# ip arp synchronize`	Enables IP ARP synchronize under the vPC Domain to facilitate faster ARP table population following device reload.
Step 13	**ipv6 nd synchronize** **Example:** `switch(config-vpc-domain)# ipv6 nd synchronize`	Enables IPv6 nd synchronization under the vPC domain to facilitate faster nd table population following device reload.

Step 14	Create vPC peer-link.
	Example:
	`switch(config)# interface port-channel 1`
	`switch(config)# switchport`
	`switch(config)# switchport mode trunk`
	`switch(config)# switchport trunk allowed vlan 1,10,100-200`
	`switch(config)# mtu 9216`
	`switch(config)# vpc peer-link`
	`switch(config)# no shut`
	
	`switch(config)# interface Ethernet 1/1, 1/21`
	`switch(config)# switchport`
	`switch(config)# mtu 9216`
	`switch(config)# channel-group 1 mode active`
	`switch(config)# no shutdown`
		Creates the vPC peer-link port-channel interface and adds two member interfaces to it.

	Command or Action	Purpose
Step 15	**system nve infra-vlans** *range* **Example:** `switch(config)# system nve infra-vlans 10`	Defines a non-VXLAN enabled VLAN as a backup routed path.
Step 16	**vlan** *number* **Example:** `switch(config)# vlan 10`	Creates the VLAN to be used as an infra-VLAN.

Step 17	Create the SVI.
	Example:
	`switch(config)# interface vlan 10`
	`switch(config)# ip address 10.10.10.1/30`
	`switch(config)# ip router ospf process UNDERLAY area 0`
	`switch(config)# ip pim sparse-mode`
	`switch(config)# no ip redirects`
	`switch(config)# mtu 9216`
	`switch(config)# no shutdown`
		Creates the SVI used for the backup routed path over the vPC peer-link.

Step 18	(Optional) **delay restore interface-vlan** *seconds* **Example:** `switch(config-vpc-domain)# delay restore interface-vlan 45`	Enables the delay restore timer for SVIs. We recommend tuning this value when the SVI/VNI scale is high. For example, when the SCI count is 1000, we recommend that you set the delay restore for **interface-vlan** to 45 seconds.

Configuring TRM with vPC Support (Cisco Nexus 9504-R and 9508-R)

SUMMARY STEPS

1. **configure terminal**
2. **feature vpc**
3. **feature interface-vlan**
4. **feature lacp**
5. **feature pim**
6. **feature ospf**
7. **ip pim rp-address** *address* **group-list** *range*
8. **vpc domain** *domain-id*
9. **hardware access-list tcam region mac-ifacl**
10. **hardware access-list tcam region vxlan 10**
11. **reload**
12. **peer switch**
13. **peer gateway**
14. **peer-keepalive destination** *ipaddress*

15. **ip arp synchronize**
16. **ipv6 nd synchronize**
17. Create vPC peer-link.
18. **system nve infra-vlans** *range*
19. **vlan** *number*
20. Create the SVI.
21. (Optional) **delay restore interface-vlan** *seconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enter global configuration mode.
Step 2	**feature vpc** **Example:** `switch(config)# feature vpc`	Enables vPCs on the device.
Step 3	**feature interface-vlan** **Example:** `switch(config)# feature interface-vlan`	Enables the interface VLAN feature on the device.
Step 4	**feature lacp** **Example:** `switch(config)# feature lacp`	Enables the LACP feature on the device.
Step 5	**feature pim** **Example:** `switch(config)# feature pim`	Enables the PIM feature on the device.
Step 6	**feature ospf** **Example:** `switch(config)# feature ospf`	Enables the OSPF feature on the device.
Step 7	**ip pim rp-address** *address* **group-list** *range* **Example:** `switch(config)# ip pim rp-address 100.100.100.1 group-list 224.0.0/4`	Defines a PIM RP address for the underlay multicast group range.
Step 8	**vpc domain** *domain-id* **Example:** `switch(config)# vpc domain 1`	Creates a vPC domain on the device and enters vpn-domain configuration mode for configuration purposes. There is no default. The range is 1–1000.
Step 9	**hardware access-list tcam region mac-ifacl** **Example:**	Carves the TCAM region for the ACL database.

	Command or Action	Purpose
	`switch(config)# hardware access-list tcam region mac-ifacl 0`	**Note** This TCAM carving command is required to enable TRM forwarding for N9K-X9636C-RX line cards only. With no TCAM region carved for **mac-ifacl**, the TCAM resources are used for TRM instead.
Step 10	**hardware access-list tcam region vxlan 10** **Example:** `switch(config)# hardware access-list tcam region vxlan 10`	Assigns the the TCAM region for use by a VXLAN. **Note** This TCAM carving command is required to enable TRM forwarding for N9K-X9636C-RX line cards only.
Step 11	**reload** **Example:** `switch(config)# reload`	Reloads the switch config for the TCAM assignments to become active.
Step 12	**peer switch** **Example:** `switch(config-vpc-domain)# peer switch`	Defines the peer switch.
Step 13	**peer gateway** **Example:** `switch(config-vpc-domain)# peer gateway`	To enable Layer 3 forwarding for packets that are destined to the gateway MAC address of the virtual port channel (vPC), use the **peer-gateway** command.
Step 14	**peer-keepalive destination** *ipaddress* **Example:** `switch(config-vpc-domain)# peer-keepalive destination 172.28.230.85`	Configures the IPv4 address for the remote end of the vPC peer-keepalive link. **Note** The system does not form the vPC peer link until you configure a vPC peer-keepalive link. The management ports and VRF are the defaults. **Note** We recommend that you configure a separate VRF and use a Layer 3 port from each vPC peer device in that VRF for the vPC peer-keepalive link. For more information about creating and configuring VRFs, see the Cisco Nexus 9000 NX-OS Series Unicast Routing Config Guide, 9.3(x).
Step 15	**ip arp synchronize** **Example:** `switch(config-vpc-domain)# ip arp synchronize`	Enables IP ARP synchronize under the vPC Domain to facilitate faster ARP table population following device reload.
Step 16	**ipv6 nd synchronize** **Example:** `switch(config-vpc-domain)# ipv6 nd synchronize`	Enables IPv6 and synchronization under the vPC domain to facilitate faster and table population following device reload.

	Command or Action	Purpose

Step 17	Create vPC peer-link.
	Example:
	`switch(config)# interface port-channel 1`
	`switch(config)# switchport`
	`switch(config)# switchport mode trunk`
	`switch(config)# switchport trunk allowed vlan 1,10,100-200`
	`switch(config)# mtu 9216`
	`switch(config)# vpc peer-link`
	`switch(config)# no shut`
	
	`switch(config)# interface Ethernet 1/1, 1/21`
	`switch(config)# switchport`
	`switch(config)# mtu 9216`
	`switch(config)# channel-group 1 mode active`
	`switch(config)# no shutdown`
		Creates the vPC peer-link port-channel interface and adds two member interfaces to it.

Step 18	**system nve infra-vlans** *range* **Example:** `switch(config)# system nve infra-vlans 10`	Defines a non-VXLAN enabled VLAN as a backup routed path.
Step 19	**vlan** *number* **Example:** `switch(config)# vlan 10`	Creates the VLAN to be used as an infra-VLAN.

Step 20	Create the SVI.
	Example:
	`switch(config)# interface vlan 10`
	`switch(config)# ip address 10.10.10.1/30`
	`switch(config)# ip router ospf process UNDERLAY area 0`
	`switch(config)# ip pim sparse-mode`
	`switch(config)# no ip redirects`
	`switch(config)# mtu 9216`
	`switch(config)# no shutdown`
		Creates the SVI used for the backup routed path over the vPC peer-link.

Step 21	(Optional) **delay restore interface-vlan** *seconds* **Example:** `switch(config-vpc-domain)# delay restore interface-vlan 45`	Enables the delay restore timer for SVIs. We recommend tuning this value when the SVI/VNI scale is high. For example, when the SCI count is 1000, we recommend that you set the delay restore for **interface-vlan** to 45 seconds.

CHAPTER 13

Configuring Cross Connect

This chapter contains the following sections:

About VXLAN Cross Connect

This feature provides point-to-point tunneling of data and control packet from one VTEP to another. Every attachment circuit will be part of a unique provider VNI. BGP EVPN signaling will discover these end-points based on how the provider VNI is stretched in the fabric. All inner customer .1q tags will be preserved, as is, and packets will be encapsulated in the provider VNI at the encapsulation VTEP. On the decapsulation end-point, the provider VNI will forward the packet to its attachment circuit while preserving all customer .1q tags in the packets.

Note Cross Connect and xconnect are synonymous.

VXLAN Cross Connect supports vPC fabric peering.

VXLAN Cross Connect enables VXLAN point-to-point functionality on the following switches:

- Cisco Nexus 9332PQ
- Cisco Nexus 9336C-FX2

- Cisco Nexus 9372PX
- Cisco Nexus 9372PX-E
- Cisco Nexus 9372TX
- Cisco Nexus 9372TX-E
- Cisco Nexus 93120TX
- Cisco Nexus 93108TC-EX
- Cisco Nexus 93108TC-FX
- Cisco Nexus 93180LC-EX
- Cisco Nexus 93180YC-EX
- Cisco Nexus 93180YC-FX
- Cisco Nexus 93240YC-FX2
- Cisco Nexus N9K-C93180YC-FX3S
- Cisco Nexus 9316D-GX
- Cisco Nexus 9364C-GX
- Cisco Nexus 93600CD-GX

VXLAN Cross Connect enables tunneling of all control frames (CDP, LLDP, LACP, STP, BFD, and PAGP) and data across the VXLAN cloud.

Guidelines and Limitations for VXLAN Cross Connect

VXLAN Cross Connect has the following guidelines and limitations:

- When an upgrade is performed non-disruptively from Cisco NX-OS Release 7.0(3)I7(4) to Cisco NX-OS Release 9.2(x) code, and if a VLAN is created and configured as xconnect, you must enter the **copy running-config startup-config** command and reload the switch. If the box was upgraded disruptively to Cisco NX-OS Release 9.2(x) code, a reload is not needed on configuring a VLAN as xconnect.
- MAC learning will be disabled on the xconnect VNIs and none of the host MAC will be learned on the tunnel access ports.
- Only supported on a BGP EVPN topology.
- LACP bundling of attachment circuits is not supported.
- Only one attachment circuit can be configured for a provider VNI on a given VTEP.
- A VNI can only be stretched in a point-to-point fashion. Point-to-multipoint is not supported.
- SVI on an xconnect VLAN is not supported.
- ARP suppression is not supported on an xconnect VLAN VNI. If ARP Suppression is enabled on a VLAN, and you enable xconnect on the VLAN, the xconnect feature takes precedence.
- Xconnect is not supported on the following switches:

  - Cisco Nexus 9504
  - Cisco Nexus 9508
  - Cisco Nexus 9516

- Scale of xconnect VLANs depends on the number of ports available on the switch. Every xconnect VLAN can tunnel all 4k customer VLANs.
- Xconnect or Crossconnect feature on vpc-vtep needs backup-svi as native VLAN on the vPC peer-link.
- Make sure that the NGOAM xconnect hb-interval is set to 5000 milliseconds on all VTEPs before attempting ISSU/patch activation to avoid link flaps.
- Before activating the patch for the cfs process, you must move the NGOAM xconnect hb-interval to the maximum value of 5000 milliseconds. This prevents interface flaps during the patch activation.
- The vPC orphan tunneled port per VNI should be either on the vPC primary switch or secondary switch, but not both.
- Configuring a static MAC on xconnect tunnel interfaces is not supported.
- xconnect is not supported on FEX ports.
- On vpc-vtep, spanning tree must be disabled on both vPC peers for xconnect VLANs.
- Xconnect access ports need to be flapped after disabling NGOAM on all the VTEPs.
- After deleting and adding a VLAN, or removing xconnect from a VLAN, physical ports need to be flapped with NGOAM.
- Beginning with Cisco NX-OS Release 9.3(3), support is added for the following switches:
  - Cisco Nexus C93600CD-GX

    Cisco Nexus C9364C-GX

    Cisco Nexus C9316D-GX

- VXLAN Cross Connect is not supported as part of multi-site solution.

Configuring VXLAN Cross Connect

This procedure describes how to configure the VXLAN Cross Connect feature.

SUMMARY STEPS

1. **configure terminal**
2. **vlan** *vlan-id*
3. **vn-segment** *vnid*
4. **xconnect**
5. **exit**
6. **interface** *type port*
7. **switchport mode dot1q-tunnel**

8. **switchport access vlan** *vlan-id*
9. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**vlan** *vlan-id* **Example:** `switch(config)# vlan 10`	Specifies VLAN.
Step 3	**vn-segment** *vnid* **Example:** `switch(config-vlan)# vn-segment 10010`	Specifies VXLAN VNID (Virtual Network Identifier).
Step 4	**xconnect** **Example:** `switch(config-vlan)# xconnect`	Defines the provider VLAN with the attached VNI to be in cross connect mode.
Step 5	**exit** **Example:** `switch(config-vlan)# exit`	Exits command mode.
Step 6	**interface** *type port* **Example:** `switch(config)# interface ethernet 1/1`	Enters interface configuration mode.
Step 7	**switchport mode dot1q-tunnel** **Example:** `switch(config-if)# switchport mode dot1q-tunnel`	Creates a 802.1q tunnel on the port. The port will do down and reinitialize (port flap) when the interface mode is changed. BPDU filtering is enabled and CDP is disabled on tunnel interfaces.
Step 8	**switchport access vlan** *vlan-id* **Example:** `switch(config-if)# switchport access vlan 10`	Sets the interface access VLAN.
Step 9	**exit** **Example:** `switch(config-vlan)# exit`	Exits command mode.

Example

This example shows how to configure VXLAN Cross Connect.

```
switch# configure terminal
switch(config)# vlan 10
switch(config)# vn-segment 10010
switch(config)# xconnect
switch(config)# vlan 20
switch(config)# vn-segment 10020
switch(config)# xconnect
switch(config)# vlan 30
switch(config)# vn-segment 10030
switch(config)# xconnect
```

This example shows how to configure access ports:

```
switch# configure terminal
switch(config)# interface ethernet1/1
switch(config-if)# switchport mode dot1q-tunnel
switch(config-if)# switchport access vlan 10
switch(config-if)# exit
switch(config)# interface ethernet1/2
switch(config-if)# switchport mode dot1q-tunnel
switch(config-if)# switchport access vlan 20
switch(config-if)# exit
switch(config)# interface ethernet1/3
switch(config-if)# switchport mode dot1q-tunnel
switch(config-if)# switchport access vlan 30
```

Verifying VXLAN Cross Connect Configuration

To display the status for the VXLAN Cross Connect configuration, enter one of the following commands:

Table 5: Display VXLAN Cross Connect Information

Command	Purpose
show running-config vlan *session-num*	Displays VLAN information.
show nve vni	Displays VXLAN VNI status.
show nve vni *session-num*	Displays VXLAN VNI status per VNI.

Example of the **show run vlan 503** command:

```
switch(config)# sh run vlan 503

!Command: show running-config vlan 503
!Running configuration last done at: Mon Jul  9 13:46:03 2018
!Time: Tue Jul 10 14:12:04 2018

version 9.2(1) Bios:version 07.64
vlan 503
vlan 503
  vn-segment 5503
  xconnect
```

Example of the **show nve vni 5503** command:

```
switch(config)# sh nve vni 5503
Codes: CP - Control Plane        DP - Data Plane
       UC - Unconfigured         SA - Suppress ARP
       SU - Suppress Unknown Unicast
Interface VNI      Multicast-group   State Mode Type [BD/VRF]      Flags
--------- -------- ----------------- ----- ---- ------------------ -----
nve1      5503     225.5.0.3         Up    CP   L2 [503]           SA       Xconn
```

Example of the **show nve vni** command:

```
switch(config)# sh nve vni
Codes: CP - Control Plane        DP - Data Plane
       UC - Unconfigured         SA - Suppress ARP
       SU - Suppress Unknown Unicast
Interface VNI      Multicast-group   State Mode Type [BD/VRF]      Flags
--------- -------- ----------------- ----- ---- ------------------ -----
nve1      5501     225.5.0.1         Up    CP   L2 [501]           SA
nve1      5502     225.5.0.2         Up    CP   L2 [502]           SA
nve1      5503     225.5.0.3         Up    CP   L2 [503]           SA       Xconn
nve1      5504     UnicastBGP        Up    CP   L2 [504]           SA       Xconn
nve1      5505     225.5.0.5         Up    CP   L2 [505]           SA       Xconn
nve1      5506     UnicastBGP        Up    CP   L2 [506]           SA       Xconn
nve1      5507     225.5.0.7         Up    CP   L2 [507]           SA       Xconn
nve1      5510     225.5.0.10        Up    CP   L2 [510]           SA       Xconn
nve1      5511     225.5.0.11        Up    CP   L2 [511]           SA       Xconn
nve1      5512     225.5.0.12        Up    CP   L2 [512]           SA       Xconn
nve1      5513     UnicastBGP        Up    CP   L2 [513]           SA       Xconn
nve1      5514     225.5.0.14        Up    CP   L2 [514]           SA       Xconn
nve1      5515     UnicastBGP        Up    CP   L2 [515]           SA       Xconn
nve1      5516     UnicastBGP        Up    CP   L2 [516]           SA       Xconn
nve1      5517     UnicastBGP        Up    CP   L2 [517]           SA       Xconn
nve1      5518     UnicastBGP        Up    CP   L2 [518]           SA       Xconn
```

Configuring NGOAM for VXLAN Cross Connect

This procedure describes how to configure NGOAM for VXLAN Cross Connect.

SUMMARY STEPS

1. **configure terminal**
2. **feature ngoam**
3. **ngoam install acl**
4. (Optional) **ngoam xconnect hb-interval** *interval*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**feature ngoam** **Example:** `switch(config)# feature ngoam`	Enters the NGOAM feature.

	Command or Action	Purpose
Step 3	**ngoam install acl** **Example:** `switch(config)# ngoam install acl`	Installs NGOAM Access Control List (ACL).
Step 4	(Optional) **ngoam xconnect hb-interval** *interval* **Example:** `switch(config)# ngoam xconnect hb-interval 5000`	Configures the heart beat interval. Range of *interval* is 150 to 5000. The default value is 190.

Verifying NGOAM for VXLAN Cross Connect

To display the NGOAM status for the VXLAN Cross Connect configuration, enter one of the following commands:

Table 6: Display VXLAN Cross Connect Information

Command	Purpose
show ngoam xconnect session all	Displays the summary of xconnect sessions.
show ngoam xconnect session *session-num*	Displays detailed xconnect information for the session.

Example of the **show ngoam xconnect session all** command:

```
switch(config)# sh ngoam xconnect session all

States: LD = Local interface down, RD = Remote interface Down
          HB = Heartbeat lost, DB = Database/Routes not present
          * - Showing Vpc-peer interface info
Vlan          Peer-ip/vni      XC-State       Local-if/State    Rmt-if/State
===============================================================================
507       6.6.6.6 /  5507      Active        Eth1/7 / UP        Eth1/5 / UP
508       7.7.7.7 /  5508      Active        Eth1/8 / UP        Eth1/5 / UP
509       7.7.7.7 /  5509      Active        Eth1/9 / UP        Eth1/9 / UP
510       6.6.6.6 /  5510      Active         Po303 / UP         Po103 / UP
513       6.6.6.6 /  5513      Active        Eth1/6 / UP        Eth1/8 / UP
```

Example of the **show ngoam xconnect session 507** command:

```
switch(config)# sh ngoam xconnect session 507
Vlan ID: 507
Peer IP: 6.6.6.6  VNI : 5507
State: Active
Last state update: 07/09/2018 13:47:03.849
Local interface: Eth1/7  State:  UP
Local vpc interface Unknown  State:  DOWN
Remote interface: Eth1/5  State:  UP
Remote vpc interface: Unknown  State:  DOWN
switch(config)#
```

NGOAM Authentication

NGOAM provides the interface statistics in the pathtrace response. 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
```

Guidelines and Limitations for Q-in-VNI

Q-in-VNI has the following guidelines and limitations:

- Q-in-VNI and selective Q-in-VNI are supported with VXLAN Flood and Learn with Ingress Replication and VXLAN EVPN with Ingress Replication.
- Q-in-VNI, selective Q-in-VNI, and QinQ-QinVNI are not supported with the multicast underlay on Cisco Nexus 9000-EX platform switches.
- The **system dot1q-tunnel transit** [**vlan** *vlan-range*] command is required when running this feature on vPC VTEPs.
- Port VLAN mapping and Q-in-VNI cannot coexist on the same port.
- Port VLAN mapping and Q-in-VNI cannot coexist on a switch if the **system dot1q-tunnel transit** command is enabled. Beginning with Cisco NX-OS Release 9.3(5), port VLAN mapping and Q-in-VNI can coexist on the same switch but on different ports and different provider VLANs, which are configured using the **system dot1q-tunnel transit vlan** *vlan-range* command.
- Beginning with Cisco NX-OS Release 10.1(1), Selective Q-in-VNI and VXLAN VLAN on Same Port feature is supported on Cisco Nexus 9300-FX3 platform switches.
- For proper operation during L3 uplink failure scenarios on vPC VTEPs, configure a backup SVI and enter the **system nve infra-vlans** *backup-svi-vlan* command. On Cisco Nexus 9000-EX platform switches, the backup SVI VLAN needs to be the native VLAN on the peer-link.
- Q-in-VNI only supports VXLAN bridging. It does not support VXLAN routing.
- The dot1q tunnel mode does not support ALE ports on Cisco Nexus 9300 Series and Cisco Nexus 9500 platform switches.
- Q-in-VNI does not support FEX.
- When configuring access ports and trunk ports for Cisco Nexus 9000 Series switches with a Network Forwarding Engine (NFE) or a Leaf Spine Engine (LSE), 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.
- Disable ARP suppression on the provider VNI for ARP traffic originated from a customer VLAN in order to flow.

```
switch(config)# interface nve 1
switch(config-if-nve)# member VNI 10000011
switch(config-if-nve-vni)# no suppress-arp
```

- Cisco Nexus 9300 platform switches support single tag. You can enable it by entering the **no overlay-encapsulation vxlan-with-tag** command for the NVE interface:

```
switch(config)# interface nve 1
switch(config-if-nve)# no overlay-encapsulation vxlan-with-tag
switch# show run int nve 1

!Command: show running-config interface nve1
!Time: Wed Jul 20 23:26:25 2016

version 7.0(3u)I4(2u)

interface nve1
  no shutdown
  source-interface loopback0
  host-reachability protocol bgp
  member vni 900001 associate-vrf
  member vni 2000980
    mcast-group 225.4.0.1
```

- Cisco Nexus 9500 platform switches do not support single tag. They support only double tag.
- Cisco Nexus 9300-EX platform switches do not support double tag. They support only single tag.
- Cisco Nexus 9300-EX platform switches do not support traffic between ports configured for Q-in-VNI and ports configured for trunk.
- Q-in-VNI cannot coexist with a VTEP that has Layer 3 subinterfaces configured. Beginning with Cisco NX-OS Release 9.3(5), this limitation no longer applies to Cisco Nexus 9332C, 9364C, 9300-FX/FX2, and 9300-GX platform switches.
- When VLAN1 is configured as the native VLAN with selective Q-in-VNI with the multiple provider tag, traffic on the native VLAN gets dropped. Do not configure VLAN1 as the native VLAN when the port is configured with selective Q-in-VNI. When VLAN1 is configured as a customer VLAN, the traffic on VLAN1 gets dropped.
- 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.
- If you have Q-in-VNI on one Cisco Nexus 9300-EX Series switch VTEP and trunk on another Cisco Nexus 9300-EX Series switch VTEP, the bidirectional traffic will not be sent between the two ports.
- Cisco Nexus 9300-EX Series of switches performing VXLAN and Q-in-Q, a mix of provider interface and VXLAN uplinks is not considered. The VXLAN uplinks have to be separated from the Q-in-Q provider or customer interface.

  For vPC use cases, the following considerations must be made when VXLAN and Q-in-Q are used on the same switch.

  - The vPC peer-link has to be specifically configured as a provider interface to ensure orphan-to-orphan port communication. In these cases, the traffic is sent with two IEEE 802.1q tags (double dot1q tagging). The inner dot1q is the customer VLAN ID while the outer dot1q is the provider VLAN ID (access VLAN).

- The vPC peer-link is used as backup path for the VXLAN encapsulated traffic in the case of an uplink failure. In Q-in-Q, the vPC peer-link also acts as the provider interface (orphan-to-orphan port communication). In this combination, use the native VLAN as the backup VLAN for traffic to handle uplink failure scenarios. Also make sure the backup VLAN is configured as a system infra VLAN (system nve infra-vlans).

- Beginning with Cisco NX-OS Release 9.3(5), Q-in-VNI is supported on Cisco Nexus 9300-GX platform switches.
- Beginning with Cisco NX-OS Release 9.3(5), Q-in-VNI supports vPC Fabric Peering.
- BPDU filter is required for Selective Q-in-VNI, as we do not support tunneling STP BPDU.

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.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *type port*
3. **switchport mode dot1q-tunnel**
4. **switchport access vlan** *vlan-id*
5. **spanning-tree bpdufilter enable**

DETAILED STEPS

	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 is an example of configuring 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)#
```

Configuring Selective Q-in-VNI

Selective Q-in-VNI is a VXLAN tunneling feature that allows a user specific range of customer VLANs on a port to be associated with one specific provider VLAN. Packets that come in with a VLAN tag that matches any of the configured customer VLANs on the port are tunneled across the VXLAN fabric using the properties of the service provider VNI. The VXLAN encapsulated packet carries the customer VLAN tag as part of the L2 header of the inner packet.

The packets that come in with a VLAN tag that is not present in the range of the configured customer VLANs on a selective Q-in-VNI configured port are dropped. This includes the packets that come in with a VLAN tag that matches the native VLAN on the port. Packets coming untagged or with a native VLAN tag are L3 routed using the native VLAN's SVI that is configured on the selective Q-in-VNI port (no VXLAN).

See the following guidelines for selective Q-in-VNI:

- Selective Q-in-VNI is supported on both vPC and non-vPC ports on Cisco Nexus 9300-EX and 9300-FX/FXP/FX2/FX3 and 9300-GX platform switches. This feature is not supported on Cisco Nexus 9200 and 9300 platform switches.
- Beginning with Cisco NX-OS Release 9.3(5), selective Q-in-VNI supports vPC Fabric Peering.
- Configuring selective Q-in-VNI on one VTEP and configuring plain Q-in-VNI on the VXLAN peer is supported. Configuring one port with selective Q-in-VNI and the other port with plain Q-in-VNI on the same switch is supported.
- Selective Q-in-VNI is an ingress VLAN tag-policing feature. Only ingress VLAN tag policing is performed with respect to the selective Q-in-VNI configured range.

  For example, selective Q-in-VNI customer VLAN range of 100-200 is configured on VTEP1 and customer VLAN range of 200-300 is configured on VTEP2. When traffic with VLAN tag of 175 is sent from VTEP1 to VTEP2, the traffic is accepted on VTEP1, since the VLAN is in the configured range and it is forwarded to the VTEP2. On VTEP2, even though VLAN tag 175 is not part of the configured range, the packet egresses out of the selective Q-in-VNI port. If a packet is sent with VLAN tag 300 from VTEP1, it is dropped because 300 is not in VTEP1's selective Q-in-VNI configured range.
- Beginning with Cisco NX-OS Release 10.1(1), Selective Q-in-VNI and Advertise PIP on a VTEP feature is supported on Cisco Nexus 9300-FX3 platform switches.
- Beginning with Cisco NX-OS Release 9.3(5), the **advertise-pip** command is supported with selective Q-in-VNI on a VTEP.
- Port VLAN mapping and selective Q-in-VNI cannot coexist on the same port.
- Port VLAN mapping and selective Q-in-VNI cannot coexist on a switch if the **system dot1q-tunnel transit** command is enabled. Beginning with Cisco NX-OS Release 9.3(5), port VLAN mapping and Q-in-VNI can coexist on the same switch but on different ports and different provider VLANs, which are configured using the **system dot1q-tunnel transit vlan** *vlan-range* command.
- Configure the **system dot1q-tunnel transit** [**vlan** *vlan-id*] command on vPC switches with selective Q-in-VNI configurations. This command is required to retain the inner Q-tag as the packet goes over the vPC peer link when one of the vPC peers has an orphan port. With this CLI configuration, the **vlan**

dot1Q tag native functionality does not work. Prior to Cisco NX-OS Release 9.3(5), every VLAN created on the switch is a provider VLAN and cannot be used for any other purpose.

Beginning with Cisco NX-OS Release 9.3(5), selective Q-in-VNI and VXLAN VLANs can be supported on the same port. With the [**vlan** *vlan-range*] option, you can specify the provider VLANs and allow other VLANs to be used for regular VXLAN traffic. In the following example, the VXLAN VLAN is 50, the provider VLAN is 501, the customer VLANs are 31-40, and the native VLAN is 2400.

```
system dot1q-tunnel transit vlan 501
interface Ethernet1/1/2
  switchport
  switchport mode trunk
  switchport trunk native vlan 2400
  switchport vlan mapping 31-40 dot1q-tunnel 501
  switchport trunk allowed vlan 50,501,2400
  spanning-tree port type edge trunk
  mtu 9216
  no shutdown
```

- The native VLAN configured on the selective Q-in-VNI port cannot be a part of the customer VLAN range. If the native VLAN is part of the customer VLAN range, the configuration is rejected.

  The provider VLAN can overlap with the customer VLAN range. For example, **switchport vlan mapping 100-1000 dot1q-tunnel 200**.

- By default, the native VLAN on any port is VLAN 1. If VLAN 1 is configured as part of the customer VLAN range using the **switchport vlan mapping** <*range*>**dot1q-tunnel** <*sp-vlan*> CLI command, the traffic with customer VLAN 1 is not carried over as VLAN 1 is the native VLAN on the port. If customer wants VLAN 1 traffic to be carried over the VXLAN cloud, they should configure a dummy native VLAN on the port whose value is outside the customer VLAN range.

- To remove some VLANs or a range of VLANs from the configured switchport VLAN mapping range on the selective Q-in-VNI port, use the **no** form of the **switchport vlan mapping** <*range*>**dot1q-tunnel** <*sp-vlan*> command.

  For example, VLAN 100-1000 is configured on the port. To remove VLAN 200-300 from the configured range, use the **no switchport vlan mapping** <*200-300*> **dot1q-tunnel** <*sp-vlan*> command.

```
interface Ethernet1/32
  switchport
  switchport mode trunk
  switchport trunk native vlan 4049
  switchport vlan mapping 100-1000 dot1q-tunnel 21
  switchport trunk allowed vlan 21,4049
  spanning-tree bpdufilter enable
  no shutdown

switch(config-if)# no sw vlan mapp 200-300 dot1q-tunnel 21
switch(config-if)# sh run int e 1/32

version 7.0(3)I5(2)

interface Ethernet1/32
  switchport
  switchport mode trunk
  switchport trunk native vlan 4049
  switchport vlan mapping 100-199,301-1000 dot1q-tunnel 21
  switchport trunk allowed vlan 21,4049
  spanning-tree bpdufilter enable
  no shutdown
```

See the following configuration examples.

- See the following example for the provider VLAN configuration:

```
vlan 50
 vn-segment 10050
```

- See the following example for configuring VXLAN Flood and Learn with Ingress Replication:

```
member vni 10050
    ingress-replication protocol static
      peer-ip 100.1.1.3
      peer-ip 100.1.1.5
      peer-ip 100.1.1.10
```

- See the following example for the interface nve configuration:

```
interface nve1
 no shutdown
 source-interface loopback0  member vni 10050
mcast-group 230.1.1.1
```

- See the following example for configuring an SVI in the native VLAN to routed traffic.

```
vlan 150
interface vlan150
 no shutdown
 ip address 150.1.150.6/24
 ip pim sparse-mode
```

- See the following example for configuring selective Q-in-VNI on a port. In this example, native VLAN 150 is used for routing the untagged packets. Customer VLANs 200-700 are carried across the dot1q tunnel. The native VLAN 150 and the provider VLAN 50 are the only VLANs allowed.

```
switch# config terminal
switch(config)#interface Ethernet 1/31
switch(config-if)#switchport
switch(config-if)#switchport mode trunk
switch(config-if)#switchport trunk native vlan 150
switch(config-if)#switchport vlan mapping 200-700 dot1q-tunnel 50
switch(config-if)#switchport trunk allowed vlan 50,150
switch(config-if)#no shutdown
```

- Disable ARP suppression on the provider VNI for ARP traffic originated from a customer VLAN in order to flow.

```
switch(config)# interface nve 1
switch(config-if-nve)# member VNI 10000011
 switch(config-if-nve-vni)# no suppress-arp
```

Configuring Q-in-VNI with LACP Tunneling

Q-in-VNI can be configured to tunnel LACP packets.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *type port*
3. **switchport mode dot1q-tunnel**
4. **switchport access vlan** *vlan-id*
5. **interface nve** *x*

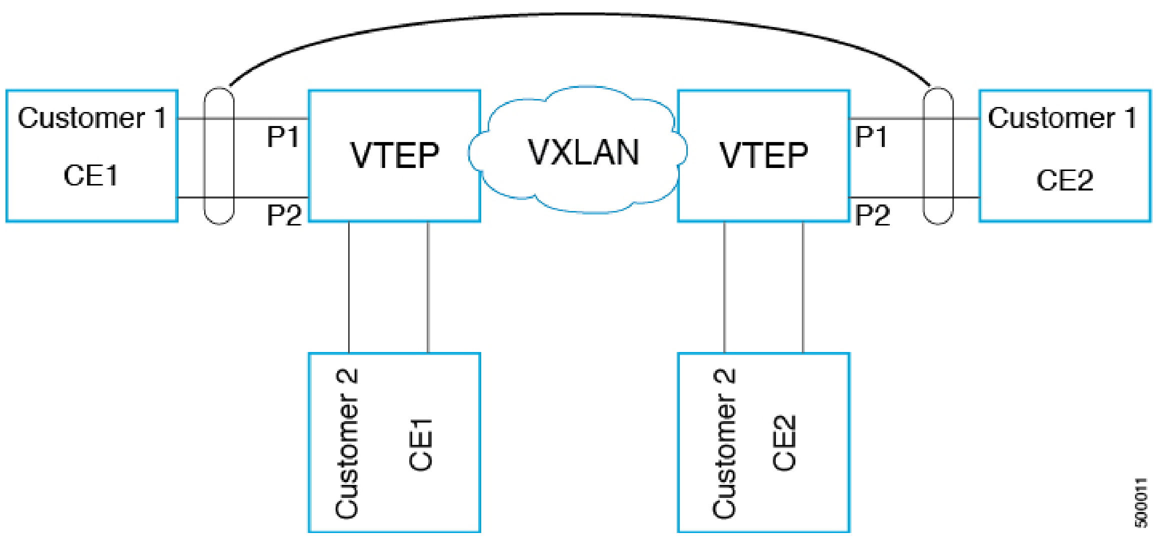
DETAILED STEPS

	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**	Enables dot1q-tunnel mode.
Step 4	**switchport access vlan** *vlan-id*	Specifies the port assigned to a VLAN.
Step 5	**interface nve** *x*	Creates a VXLAN overlay interface that terminates VXLAN tunnels.

Example

- The following is an example topology that pins each port of a port-channel pair to a unique VM. The port-channel is stretched from the CE perspective. There is no port-channel on VTEP. The traffic on P1 of CE1 transits to P1 of CE2 using Q-in-VNI.

Figure 29: LACP Tunneling Over VXLAN P2P Tunnels



Note

- Q-in-VNI can be configured to tunnel LACP packets. (Able to provide port-channel connectivity across data-centers.)
    - Gives impression of L1 connectivity and co-location across data-centers.
    - Exactly two sites. Traffic coming from P1 of CE1 goes out of P1 of CE2. If P1 of CE1 goes down, LACP provides coverage (over time) to redirect traffic to P2.
- Uses static ingress replication with VXLAN with flood and learn. Each port of the port channel is configured with Q-in-VNI. There are multiple VNIs for each member of a port-channel and each port is pinned to specific VNI.
    - To avoid saturating the MAC, you should turn off/disable learning of VLANS.
- Configuring Q-in-VNI to tunnel LACP packets is not supported for VXLAN EVPN.
- The number of port-channel members supported is the number of ports supported by the VTEP.

Selective Q-in-VNI with Multiple Provider VLANs

About Selective Q-in-VNI with Multiple Provider VLANs

Selective Q-in-VNI with multiple provider VLANs is a VXLAN tunneling feature. This feature allows a user specific range of customer VLANs on a port to be associated with one specific provider VLAN. It also enables you to have multiple customer-VLAN to provider-VLAN mappings on a port. Packets that come in with a VLAN tag which matches any of the configured customer VLANs on the port are tunneled across the VXLAN fabric using the properties of the service provider VNI. The VXLAN encapsulated packet carries the customer VLAN tag as part of the Layer 2 header of the inner packet.

Guidelines and Limitations for Selective Q-in-VNI with Multiple Provider VLANs

Selective Q-in-VNI with multiple provider VLANs has the following guidelines and limitations:

- All the existing guidelines and limitations for Selective Q-in-VNI apply.
- This feature is supported with VXLAN BGP EVPN IR mode only.
- When enabling multiple provider VLANs on a vPC port channel, make sure that the configuration is consistent across the vPC peers.
- Port VLAN mapping and selective Q-in-VNI cannot coexist on the same port.
- Port VLAN mapping and selective Q-in-VNI cannot coexist on a switch if the **system dot1q-tunnel transit** command is enabled. Beginning with Cisco NX-OS Release 9.3(5), port VLAN mapping and selective Q-in-VNI can coexist on the same switch but on different ports and different provider VLANs, which are configured using the **system dot1q-tunnel transit vlan** *vlan-range* command.

- The **system dot1q-tunnel transit** [**vlan** *vlan-range*] command is required when using this feature on vPC VTEPs.
- For proper operation during Layer 3 uplink failure scenarios on vPC VTEPs, configure the backup SVI and enter the **system nve infra-vlans** *backup-svi-vlan* command. On Cisco Nexus 9000-EX platform switches, the backup SVI VLAN must be the native VLAN on the peer-link.
- As a best practice, do not allow provider VLANs on a regular trunk.
- We recommend not creating or allowing customer VLANs on the switch where customer-VLAN to provider-VLAN mapping is configured.
- We do not support specific native VLAN configuration when the **switchport vlan mapping all dot1q-tunnel** command is entered.
- Beginning with Cisco NX-OS Release 9.3(5), selective Q-in-VNI with a multiple provider tag supports vPC Fabric Peering.
- Disable ARP suppression on the provider VNI for ARP traffic originated from a customer VLAN in order to flow.

```
switch(config)# interface nve 1
switch(config-if-nve)# member VNI 10000011
switch(config-if-nve-vni)# no suppress-arp
```

- All incoming traffic should be tagged when the interface is configured with the **switchport vlan mapping all dot1q-tunnel** command.

Configuring Selective Q-in-VNI with Multiple Provider VLANs

You can configure selective Q-in-VNI with multiple provider VLANs.

Before you begin

You must configure provider VLANs and associate the VLAN to a vn-segment.

SUMMARY STEPS

1. Enter global configuration mode.
2. Configure Layer 2 VLANs and associate them to a vn-segment.
3. Enter interface configuration mode where the traffic comes in with a dot1Q VLAN tag.

DETAILED STEPS

Step 1 Enter global configuration mode.

```
switch# configure terminal
```

Step 2 Configure Layer 2 VLANs and associate them to a vn-segment.

```
switch(config)# vlan 10
vn-segment 10000010
switch(config)# vlan 20
 vn-segment 10000020
```

Step 3 Enter interface configuration mode where the traffic comes in with a dot1Q VLAN tag.

```
switch(config)# interf port-channel 10
switch(config-if)# switchport
switch(config-if)# switchport mode trunk
switch(config-if)# switchport trunk native vlan 3962
switch(config-if)# switchport vlan mapping 2-400 dot1q-tunnel 10
switch(config-if)# switchport vlan mapping 401-800 dot1q-tunnel 20
switch(config-if)# switchport vlan mapping 801-1200 dot1q-tunnel 30
switch(config-if)# switchport vlan mapping 1201-1600 dot1q-tunnel 40
switch(config-if)# switchport vlan mapping 1601-2000 dot1q-tunnel 50
switch(config-if)# switchport vlan mapping 2001-2400 dot1q-tunnel 60
switch(config-if)# switchport vlan mapping 2401-2800 dot1q-tunnel 70
switch(config-if)# switchport vlan mapping 2801-3200 dot1q-tunnel 80
switch(config-if)# switchport vlan mapping 3201-3600 dot1q-tunnel 90
switch(config-if)# switchport vlan mapping 3601-3960 dot1q-tunnel 100
switch(config-if)# switchport trunk allowed vlan 10,20,30,40,50,60,70,80,90,100,3961-3967
```

Example

This example shows how to configure Selective Qinvni with multiple provider VLANs:

```
switch# show run vlan 121
vlan 121
vlan 121
  vn-segment 10000021

switch#
switch# sh run interf port-channel 5

interface port-channel5
  description VPC PO
  switchport
  switchport mode trunk
  switchport trunk native vlan 504
  switchport vlan mapping 11 dot1q-tunnel 111
  switchport vlan mapping 12 dot1q-tunnel 112
  switchport vlan mapping 13 dot1q-tunnel 113
  switchport vlan mapping 14 dot1q-tunnel 114
  switchport vlan mapping 15 dot1q-tunnel 115
  switchport vlan mapping 16 dot1q-tunnel 116
  switchport vlan mapping 17 dot1q-tunnel 117
  switchport vlan mapping 18 dot1q-tunnel 118
  switchport vlan mapping 19 dot1q-tunnel 119
  switchport vlan mapping 20 dot1q-tunnel 120
  switchport trunk allowed vlan 111-120,500-505
  vpc 5

switch#

switch# sh spanning-tree vlan 111

VLAN0111
  Spanning tree enabled protocol rstp
  Root ID    Priority    32879
             Address     7079.b3cf.956d
             This bridge is the root
             Hello Time  2  sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32879  (priority 32768 sys-id-ext 111)
             Address     7079.b3cf.956d
```

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

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Po1              Desg FWD 1         128.4096 (vPC peer-link) Network P2p
Po5              Desg FWD 1         128.4100 (vPC) P2p
Eth1/7/2         Desg FWD 10        128.26   P2p

switch#


switch# sh vlan internal info mapping | b Po5
  ifindex Po5(0x16000004)
  vlan mapping enabled: TRUE
  vlan translation mapping information (count=10):
    Original Vlan       Translated Vlan
    -------------       ---------------
    11                   111
    12                   112
    13                   113
    14                   114
    15                   115
    16                   116
    17                   117
    18                   118
    19                   119
    20                   120
switch#


switch# sh consistency-checker vxlan selective-qinvni interface port-channel 5
Performing port specific checks for intf port-channel5
Port specific selective QinVNI checks for interface port-channel5 : PASS
Performing port specific checks for intf port-channel5
Port specific selective QinVNI checks for interface port-channel5 : PASS

switch#
```

Configuring QinQ-QinVNI

Overview for QinQ-QinVNI

- QinQ-QinVNI is a VXLAN tunneling feature that allows you to configure a trunk port as a multi-tag port to preserve the customer VLANs that are carried across the network.
- On a port that is configured as multi-tag, packets are expected with multiple-tags or at least one tag. When multi-tag packets ingress on this port, the outer-most or first tag is treated as provider-tag or provider-vlan. The remaining tags are treated as customer-tag or customer-vlan.
- This feature is supported on both vPC and non-vPC ports.
- Ensure that the **switchport trunk allow-multi-tag** command is configured on both of the vPC-peers. It is a type 1 consistency check.
- This feature is supported with VXLAN Flood and Learn and VXLAN EVPN.

Guidelines and Limitations for QinQ-QinVNI

QinQ-QinVNI has the following guidelines and limitations:

- This feature is supported on the Cisco Nexus 9300-FX/FX2/FX3, and 9300-GX platform switches.
- This feature supports vPC Fabric Peering.
- On a multi-tag port, provider VLANs must be a part of the port. They are used to derive the VNI for that packet.
- Untagged packets are associated with the native VLAN. If the native VLAN is not configured, the packet is associated with the default VLAN (VLAN 1).
- Packets coming in with an outermost VLAN tag (provider-vlan), not present in the range of allowed VLANs on a multi-tag port, are dropped.
- Packets coming in with an outermost VLAN tag (provider-vlan) tag matching the native VLAN are routed or bridged in the native VLAN's domain.
- This feature supports VXLAN bridging but does not support VXLAN routing.
- Multicast data traffic with more than two Q-Tags is not supported when snooping is enabled on the VXLAN VLAN.
- You need at least one multi-tag trunk port allowing the provider VLANs in Up state on both vPC peers. Otherwise, traffic traversing via the peer-link for these provider VLANs will not carry all inner C-Tags.
- The **system dot1q-tunnel transit** [**vlan** *vlan-range*] command is required when running this feature on vPC VTEPs.

Configuring QinQ-QinVNI

Note You can also carry native VLAN (untagged traffic) on the same multi-tag trunk port.

The native VLAN on a multi-tag port cannot be configured as a provider VLAN on another multi-tag port or a dot1q enabled port on the same switch.

The **allow-multi-tag** command is allowed only on a trunk port. It is not available on access or dot1q ports.

The **allow-multi-tag** command is not allowed on Peer Link ports. Port channel with multi-tag enabled must not be configured as a vPC peer-link.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *slot/port*
3. **switchport**
4. **switchport mode trunk**
5. **switchport trunk native vlan** *vlan-id*
6. **switchport trunk allowed vlan** *vlan-list*
7. **switchport trunk allow-multi-tag**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**interface ethernet** *slot/port* **Example:** `switch(config)# interface ethernet1/7`	Specifies the interface that you are configuring.
Step 3	**switchport** **Example:** `switch(config-inf)# switchport`	Configures it as a Layer 2 port.
Step 4	**switchport mode trunk** **Example:** `switch(config-inf)# switchport mode trunk`	Sets the interface as a Layer 2 trunk port.
Step 5	**switchport trunk native vlan** *vlan-id* **Example:** `switch(config-inf)# switchport trunk native vlan 30`	Sets the native VLAN for the 802.1Q trunk. Valid values are from 1 to 4094. The default value is VLAN1.
Step 6	**switchport trunk allowed vlan** *vlan-list* **Example:** `switch(config-inf)# switchport trunk allowed vlan 10,20,30`	Sets the allowed VLANs for the trunk interface. The default is to allow all VLANs on the trunk interface: 1 to 3967 and 4048 to 4094. VLANs 3968 to 4047 are the default VLANs reserved for internal use by default.
Step 7	**switchport trunk allow-multi-tag** **Example:** `switch(config-inf)# switchport trunk allow-multi-tag`	Sets the allowed VLANs as the provider VLANs excluding the native VLAN. In the following example, VLANs 10 and 20 are provider VLANs and can carry multiple Inner Q-tags. Native VLAN 30 will not carry inner Q-tags.

Example

```
interface Ethernet1/7
switchport
switchport mode trunk
switchport trunk native vlan 30
switchport trunk allow-multi-tag
switchport trunk allowed vlan 10,20,30
no shutdown
```

Removing a VNI

Use this procedure to remove a VNI.

Step 1 Remove the VNI under NVE.

Step 2 Remove the VRF from BGP (applicable when decommissioning for Layer 3 VNI).

Step 3 Delete the SVI.

Step 4 Delete the VLAN and VNI.

CHAPTER 14

Configuring Port VLAN Mapping

This chapter contains the following sections:

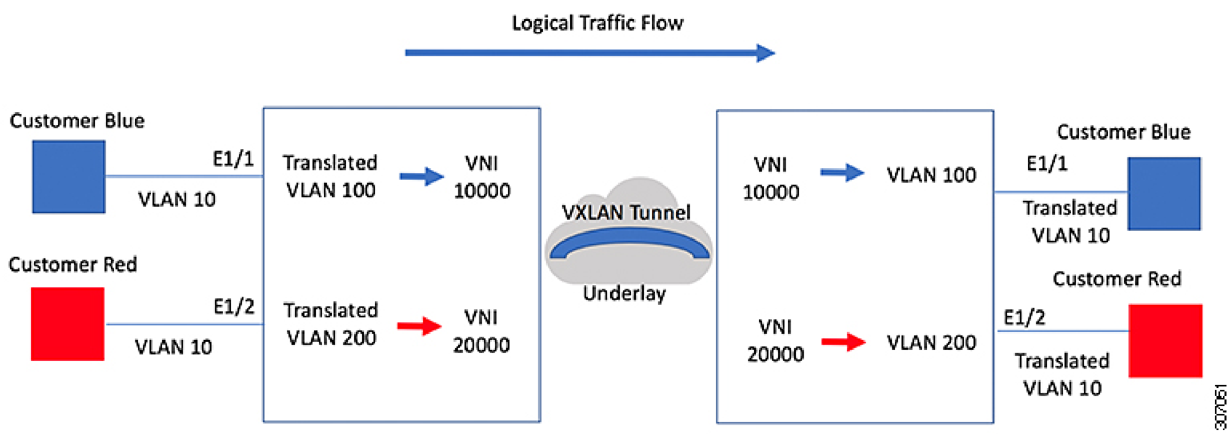
About Translating Incoming VLANs

Sometimes a VLAN translation is required or desired. One such use case is when a service provider has multiple customers connecting to the same physical switch using the same VLAN encapsulation, but they are not and should not be on the same Layer 2 segment. In such cases translating the incoming VLAN to a unique VLAN that is then mapped to a VNI is the right way to extending the segment. In the figure below two customers, Blue and Red are both connecting to the leaf using VLAN 10 as their encapsulation.

Customers Blue and Red should not be on the same VNI. In this example VLAN 10 for Customer Blue (on interface E1/1) is mapped/translated to VLAN 100, and VLAN 10 for customer Red (on interface E1/2) is mapped to VLAN 200. In turn, VLAN 100 is mapped to VNI 10000 and VLAN 200 is mapped to VNI 20000.

On the other leaf, this mapping is applied in reverse. Incoming VXLAN encapsulated traffic on VNI 10000 is mapped to VLAN 100 which in turn is mapped to VLAN 10 on Interface E1/1. VXLAN encapsulated traffic on VNI 20000 is mapped to VLAN 200 which in turn is mapped to VLAN 10 on Interface E1/2.

Figure 30: Logical Traffic Flow



You can configure VLAN translation between the ingress (incoming) VLAN and a local (translated) VLAN on a port. For the traffic arriving on the interface where VLAN translation is enabled, the incoming VLAN is mapped to a translated VLAN that is VXLAN enabled.

On the underlay, this is mapped to a VNI, the inner dot1q is deleted, and switched over to the VXLAN network. On the egress switch, the VNI is mapped to a translated VLAN. On the outgoing interface, where VLAN translation is configured, the traffic is converted to the original VLAN and egressed out. Refer to the VLAN counters on the translated VLAN for the traffic counters and not on the ingress VLAN. Port VLAN (PV) mapping is an access side feature and is supported with both multicast and ingress replication for flood and learn and MP-BGP EVPN mode for VXLAN.

Guidelines and Limitations for Port VLAN Mapping

The following are the guidelines and Limitations for Port VLAN Mapping:

- Support is added for vPC Fabric Peering.
- VLAN translation is supported only on VXLAN enabled VLANs
- The ingress (incoming) VLAN does not need to be configured on the switch as a VLAN. The translated VLAN needs to be configured and a vn-segment mapping given to it. An NVE interface with VNI mapping is essential for the same.
- All Layer 2 source address learning and Layer 2 MAC destination lookup occurs on the translated VLAN. Refer to the VLAN counters on the translated VLAN and not on the ingress (incoming) VLAN.
- Port VLAN mapping is supported on Cisco Nexus 9300, 9300-EX, and 9300-FX3 platform switches.
- Cisco Nexus 9300 and 9500 switches support switching and routing on overlapped VLAN interfaces. Only VLAN-mapping switching is applicable for Cisco Nexus 9300-EX/FX/FX2/FX3 platform switches and Cisco Nexus 9500 with -EX/FX line cards.
- Port VLAN routing is supported on the following platforms:
    - Beginning with Cisco NX-OS Release 7.x, this feature is supported on Cisco Nexus 9300-EX/FX/FX2 platform switches.

    - Beginning with Cisco NX-OS Release 9.2(x), this feature is supported on Cisco Nexus 9300-GX platform switches.
    - Beginning with Cisco NX-OS Release 9.3(x), this feature is supported on Cisco Nexus 9300-FX3 platform switches.

- Beginning with Cisco NX-OS Release 9.3(3), PV Translation is supported for Cisco Nexus 9300-GX platform switches.
- On Cisco Nexus 9300 Series switches with NFE ASIC, PV routing is not supported on 40 G ALE ports.
- PV routing supports configuring an SVI on the translated VLAN for flood and learn and BGP EVPN mode for VXLAN.
- VLAN translation (mapping) is supported on Cisco Nexus 9000 Series switches with a Network Forwarding Engine (NFE).
- When changing a property on a translated VLAN, the port that has a mapping configuration with that VLAN as the translated VLAN, must be flapped to ensure correct behavior. This is applicable only to the following platforms:
    - N9K-C9504 modules
    - N9K-C9508 modules
    - N9K-C9516 modules
    - Nexus 9400 line cards
    - Nexus 9500 line cards
    - Nexus 9600 line cards
    - Nexus 9700-X Cloud Scale line cards
    - Nexus 9600-R and R2 line cards

```
Int eth 1/1
switchport vlan mapping 101 10
.
.
.

/***Deleting vn-segment from vlan 10.***/
/***Adding vn-segment back.***/
/***Flap Eth 1/1 to ensure correct behavior.***/
```

- The following example shows incoming VLAN 10 being mapped to local VLAN 100. Local VLAN 100 will be the one mapped to a VXLAN VNI.

```
interface ethernet1/1
switchport vlan mapping 10 100
```

- The following is an example of overlapping VLAN for PV translation. In the first statement, VLAN-102 is a translated VLAN with VNI mapping. In the second statement, VLAN-102 the VLAN where it is translated to VLAN-103 with VNI mapping.

```
interface ethernet1/1
switchport vlan mapping 101 102
switchport vlan mapping 102 103/
```

- When adding a member to an existing port channel using the force command, the "mapping enable" configuration must be consistent. For example:

```
Int po 101
switchport vlan mapping enable
switchport vlan mapping 101 10
switchport trunk allowed vlan 10

int eth 1/8
/***No configuration***/
```

> **Note** The **switchport vlan mapping enable** command is supported only when the port mode is trunk.

- Port VLAN mapping is not supported on Cisco Nexus 9200 platform switches.
- VLAN mapping helps with VLAN localization to a port, scoping the VLANs per port. A typical use case is in the service provider environment where the service provider leaf switch has different customers with overlapping VLANs that come in on different ports. For example, customer A has VLAN 10 coming in on Eth 1/1 and customer B has VLAN 10 coming in on Eth 2/2.

  In this scenario, you can map the customer VLAN to a provider VLAN and map that to a Layer 2 VNI. There is an operational benefit in terminating different customer VLANs and mapping them to the fabric-managed VLANs, L2 VNIs.
- An NVE interface with VNI mapping must be configured for Port VLAN translation to work.
- You should not enable super bridging VLAN in the provider VLAN list of the **system dot1q-tunnel transit vlan** <*id*> command. If enabled it will end up in unrecoverable functional and forwarding impacts.

Configuring Port VLAN Mapping on a Trunk Port

Before you begin

- Ensure that the physical or port channel on which you want to implement VLAN translation is configured as a Layer 2 trunk port.
- Ensure that the translated VLANs are created on the switch and are also added to the Layer 2 trunk ports trunk-allowed VLAN vlan-list.

> **Note** As a best practice, do not add the ingress VLAN ID to the switchport allowed vlan-list under the interface.

- Ensure that all translated VLANs are VXLAN enabled.

SUMMARY STEPS

1. **configure terminal**

2. **interface** *type/port*
3. **[no] switchport vlan mapping enable**
4. **[no] switchport vlan mapping** *vlan-id translated-vlan-id*
5. **[no] switchport vlan mapping all**
6. **copy running-config startup-config**
7. **show interface** [*if-identifier*] **vlan mapping**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**interface** *type/port* **Example:** `switch(config)# interface Ethernet1/1`	Specifies the interface that you are configuring.
Step 3	**[no] switchport vlan mapping enable** **Example:** `switch(config-if)# [no] switchport vlan mapping enable`	Enables VLAN translation on the switch port. VLAN translation is disabled by default. **Note** Use the **no** form of this command to disable VLAN translation.
Step 4	**[no] switchport vlan mapping** *vlan-id translated-vlan-id* **Example:** `switch(config-if)# switchport vlan mapping 10 100`	Translates a VLAN to another VLAN. • The range for both the *vlan-id* and *translated-vlan-id* arguments are from 1 to 4094. • You can configure VLAN translation between the ingress (incoming) VLAN and a local (translated) VLAN on a port. For the traffic arriving on the interface where VLAN translation is enabled, the incoming VLAN is mapped to a translated VLAN that is VXLAN enabled. On the underlay, this is mapped to a VNI, the inner dot1q is deleted, and switched over to the VXLAN network. On the egress switch, the VNI is mapped to a local translated VLAN. On the outgoing interface, where VLAN translation is configured, the traffic is converted to the original VLAN and egresses out. **Note** Use the **no** form of this command to clear the mappings between a pair of VLANs.
Step 5	**[no] switchport vlan mapping all** **Example:** `switch(config-if)# switchport vlan mapping all`	Removes all VLAN mappings configured on the interface.

	Command or Action	Purpose
Step 6	**copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration. **Note** The VLAN translation configuration does not become effective until the switch port becomes an operational trunk port.
Step 7	**show interface** [*if-identifier*] **vlan mapping** **Example:** `switch# show interface ethernet1/1 vlan mapping`	Displays VLAN mapping information for a range of interfaces or for a specific interface.

Example

This example shows how to configure VLAN translation between (the ingress) VLAN 10 and (the local) VLAN 100. The show vlan counters command output shows the statistic counters as translated VLAN instead of customer VLAN.

```
switch#  configure terminal
switch(config)#  interface ethernet1/1
switch(config-if)#  switchport vlan mapping enable
switch(config-if)#  switchport vlan mapping 10 100
switch(config-if)#  switchport trunk allowed vlan 100
switch(config-if)#  show interface ethernet1/1 vlan mapping
Interface eth1/1:
Original VLAN           Translated VLAN
------------------      ---------------
10                           100

switch(config-if)#  show vlan counters
Vlan Id                              :100
Unicast Octets In                    :292442462
Unicast Packets In                   :1950525
Multicast Octets In                  :14619624
Multicast Packets In                 :91088
Broadcast Octets In                  :14619624
Broadcast Packets In                 :91088
Unicast Octets Out                   :304012656
Unicast Packets Out                  :2061976
L3 Unicast Octets In                 :0
L3 Unicast Packets In                :0
```

Configuring Inner VLAN and Outer VLAN Mapping on a Trunk Port

Configuring Inner VLAN and Outer VLAN Mapping on a Trunk Port is applicable only for Cisco Nexus 9300 platforms and not supported on Cisco Nexus 9200, 9300-EX, 9300-FX, 9300-FX2, 9300-FX3, 9300-GX, 9300-GX2, 9364C, 9332C platforms.

You can configure VLAN translation from an inner VLAN and an outer VLAN to a local (translated) VLAN on a port. For the double tag VLAN traffic arriving on the interfaces where VLAN translation is enabled, the inner VLAN and outer VLAN are mapped to a translated VLAN that is VXLAN enabled.

Notes for configuring inner VLAN and outer VLAN mapping:

- Inner and outer VLAN cannot be on the trunk allowed list on a port where inner VLAN and outer VLAN is configured.

  For example:

```
switchport vlan mapping 11 inner 12 111
switchport trunk allowed vlan 11-12,111 /***Not valid because 11 is outer VLAN and 12
is inner VLAN.***/
```

- On the same port, no two mapping (translation) configurations can have the same outer (or original) or translated VLAN. Multiple inner VLAN and outer VLAN mapping configurations can have the same inner VLAN.

  For example:

```
switchport vlan mapping 101 inner 102 1001
switchport vlan mapping 101 inner 103 1002  /***Not valid because 101 is already used
as an original VLAN.***/
switchport vlan mapping 111 inner 104 1001  /***Not valid because 1001 is already used
 as a translated VLAN.***/
switchport vlan mapping 106 inner 102 1003  /***Valid because inner vlan can be the
same.***/
```

- When a packet comes double-tagged on a port which is enabled with the inner option, only bridging is supported.
- VXLAN PV routing is not supported for double-tagged frames.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *type port*
3. [**no**] **switchport mode trunk**
4. **switchport vlan mapping enable**
5. **switchport vlan mapping** *outer-vlan-id* **inner** *inner-vlan-id translated-vlan-id*
6. (Optional) **copy running-config startup-config**
7. (Optional) **show interface** [*if-identifier*] **vlan mapping**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**interface** *type port*	Enters interface configuration mode.
Step 3	[**no**] **switchport mode trunk**	Enters trunk configuration mode.

	Command or Action	Purpose
Step 4	**switchport vlan mapping enable**	Enables VLAN translation on the switch port. VLAN translation is disabled by default. **Note** Use the**no** form of this command to disable VLAN translation.
Step 5	**switchport vlan mapping** *outer-vlan-id* **inner** *inner-vlan-id translated-vlan-id*	Translates inner VLAN and outer VLAN to another VLAN.
Step 6	(Optional) **copy running-config startup-config**	Copies the running configuration to the startup configuration. **Note** The VLAN translation configuration does not become effective until the switch port becomes an operational trunk port
Step 7	(Optional) **show interface** [*if-identifier*] **vlan mapping**	Displays VLAN mapping information for a range of interfaces or for a specific interface.

Example

This example shows how to configure translation of double tag VLAN traffic (inner VLAN 12; outer VLAN 11) to VLAN 111.

```
switch# configure terminal
switch(config)# interface ethernet1/1
switch(config-if)# switchport mode trunk
switch(config-if)# switchport vlan mapping enable
switch(config-if)# switchport vlan mapping 11 inner 12 111
switch(config-if)# switchport trunk allowed vlan 101-170
switch(config-if)# no shutdown

switch(config-if)# show mac address-table dynamic vlan 111

Legend:
        * - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
        age - seconds since last seen,+ - primary entry using vPC Peer-Link,
        (T) - True, (F) - False
   VLAN     MAC Address      Type      age     Secure NTFY Ports
---------+-----------------+--------+---------+------+----+------------------
*  111     0000.0092.0001   dynamic  0         F      F    nve1(100.100.100.254)
*  111     0000.0940.0001   dynamic  0         F      F    Eth1/1
```

About Port Multi-VLAN Mapping

With Port Multi-VLAN Mapping feature multiple VLANs are mapped on a trunk interface to a single global VLAN/VNI. Layer 2 (L2) sub-interface has to be created for the mapping and a qTag has to be provided for each L2 sub-interface.

Different Port-VLANs can serve different services on the same physical interface.

For the Port Multi-VLAN mappings per trunk port, ACLs are installed per each of the mapping using L2 sub-interface. Some ACLs are installed automatically by default and some are installed with static MAC address configuration. L2 sub-interface has a qtag, flood-domain or provider-VLAN. The provider-VLAN is configured on the switch and is used for traffic forwarding. There can be only one provider-VLAN on the switch.

This static MAC configuration is done using the **switchport mac-address static-only** command configured on L2 sub-interface parent port. This command disables the MAC learning on the parent port and enables MAC-ACL per each static MAC configured on the L2 sub-interfaces.

Guidelines and Limitations for Port Multi-VLAN Mapping

The following are the guidelines and limitations for Port Multi-VLAN Mapping:

- Beginning with Cisco NX-OS Release 10.1(2), Port Multi-VLAN Mapping is supported on Cisco Nexus 9300-EX, FX, and FX2 platform switches.
- Port Multi-VLAN Mapping is an access side feature and is supported with both multicast and ingress replication for VXLAN flood and learn mode. This feature is not supported for VXLAN MP-BGP EVPN mode in Cisco NX-OS Release 10.1(2).
- For a device that is running on Cisco Nexus Release 10.1(2) ND-ISSU is not supported if L2 sub-interfaces are configured.
- This feature is not supported with vPC fabric peering configuration.
- In order to protect against broadcast or multicast flood, all flooding traffic is dropped except ARP and NS/ND.
- Layer 2 is supported.
- STP is not supported.
- Static default route or specific route to remote VTEP is recommended to be configured on ToRs.
- Interaction with other access features like QinQ/QinVNI, Port VLAN mapping, PVLAN and Xconnect are not supported.

The following are the guidelines and limitations related to the parent interface:

- TCAM entries are only installed on the slice where the parent port exists. To check TCAM utilization, use the **show system internal access-list resource utilization** command.
- To check the port slice, use the **show interface hardware-mappings** command.
- For hosts using static ARP, add on ToR static MAC entry for remote host on interface nve 1. Example:

```
mac address-table static 0034.0100.0001 vni 10013001 interface nve 1 peer-ip 192.168.75.2
```

- Port-security/dot1x is not supported on the parent interface.
- vPC mode is not supported for parent interface or L2 sub interface.

The following are the guidelines and limitations related to the sub interface:

- Maximum of 510 sub-interfaces are supported per switch.

- ACL and storm-control per sub-interface cannot be configured under the switch port mapping.
- TCAM region must be re-configured in order to support Max 510 L2 sub interfaces. For each L2 sub interface nine TCAM ing-pacl-sb entries are allocated.
- Static MAC is configured on L2 sub interface using the **switchport mac-address static-only** command on the parent interface.
- L2 sub interfaces are not supported without VXLAN deployment. The provider VLAN must be a VXLAN VLAN.
- Dynamic MAC learning is disabled on L2 sub interface.
- Storm control is not supported for L2 sub interface.
- The **hardware profile svi-and-si flex-stats-enable** command supports only ingress L2 sub interface counters. This profile statistics command does not support egress L2 sub interface counters and VxLAN statistics.
- IGMP snooping is not supported on the provider VLAN where L2 sub interface is configured.

Configuring Port Multi-VLAN Mapping

A sample configuration of Port Multi-VLAN Mapping is provided below:

```
feature ospf
feature pim
feature bfd
feature interface-vlan
feature vn-segment-vlan-based
feature private-vlan
feature lacp
feature nv overlay

hardware access-list tcam region ing-pacl-sb 2560
hardware profile svi-and-si flex-stats-enable

ip pim rp-address 2.0.0.254 group-list 224.0.0.0/4

vlan 3001
  vn-segment 10013001

interface Ethernet1/22
  switchport
  switchport mode trunk
  switchport trunk allowed vlan 3001
  mtu 9216
  storm-control broadcast level 0.01
  storm-control action trap
  switchport isolated
  switchport mac-address static-only
  no shutdown

interface Ethernet1/22.1
  encapsulation dot1q 301 provider-vlan 3001
  no shutdown

interface Ethernet1/22.2
  encapsulation dot1q 302 provider-vlan 3001
```

```
  no shutdown

interface Ethernet1/22.3
  encapsulation dot1q 303 provider-vlan 3001
  no shutdown

interface Ethernet1/22.4
  encapsulation dot1q 304 provider-vlan 3001
  no shutdown

interface Ethernet1/22.5
  encapsulation dot1q 305 provider-vlan 3001
  no shutdown

interface port-channel1
  switchport
  switchport mode trunk
  switchport trunk allowed vlan 3001
  mtu 9216
  storm-control broadcast level 0.01
  storm-control multicast level 0.01
  storm-control unicast level 0.01
  storm-control action trap
  switchport isolated
  switchport mac-address static-only

interface port-channel1.1
  encapsulation dot1q 301 provider-vlan 3001
  no shutdown

interface port-channel1.2
  encapsulation dot1q 302 provider-vlan 3001
  no shutdown

interface port-channel1.3
  encapsulation dot1q 303 provider-vlan 3001
  no shutdown

interface port-channel1.4
  encapsulation dot1q 304 provider-vlan 3001
  no shutdown

interface port-channel1.5
  encapsulation dot1q 305 provider-vlan 3001
  no shutdown

interface Ethernet1/24
  switchport
  switchport mode trunk
  switchport trunk allowed vlan 3001
  mtu 9216
  storm-control broadcast level 0.01
  storm-control multicast level 0.01
  storm-control unicast level 0.01
  storm-control action trap
  switchport isolated
  switchport mac-address static-only
  channel-group 1 mode active
  no shutdown

interface Ethernet1/25
  switchport
  switchport mode trunk
  switchport trunk allowed vlan 3001
```

```
  mtu 9216
  storm-control broadcast level 0.01
  storm-control multicast level 0.01
  storm-control unicast level 0.01
  storm-control action trap
  switchport isolated
  switchport mac-address static-only
  channel-group 1 mode active
  no shutdown

mac address-table static 0035.0100.0001 vlan 3001 interface Ethernet1/22.1
mac address-table static 0035.0100.0002 vlan 3001 interface Ethernet1/22.2
mac address-table static 0035.0100.0003 vlan 3001 interface Ethernet1/22.3
mac address-table static 0035.0100.0004 vlan 3001 interface Ethernet1/22.4
mac address-table static 0035.0100.0005 vlan 3001 interface Ethernet1/22.5

mac address-table static 003b.0100.0001 vlan 3001 interface port-channel1.1
mac address-table static 003b.0100.0002 vlan 3001 interface port-channel1.2
mac address-table static 003b.0100.0003 vlan 3001 interface port-channel1.3
mac address-table static 003b.0100.0004 vlan 3001 interface port-channel1.4
mac address-table static 003b.0100.0005 vlan 3001 interface port-channel1.5


router ospf p1
  bfd
  router-id 192.168.210.1

interface loopback0
  ip address 192.168.210.1/32
  ip router ospf p1 area 0.0.0.0
  ip pim sparse-mode

interface loopback1
  description NVE_IP
  ip address 192.168.210.2/32
  ip router ospf p1 area 0.0.0.0
  ip pim sparse-mode

interface Ethernet1/49
  mtu 9216
  no ip redirects
  ip address 10.0.1.16/31
  ip router ospf p1 area 0.0.0.0
  ip pim sparse-mode
  no shutdown

interface Ethernet1/54
  mtu 9216
  no ip redirects
  ip address 10.0.1.18/31
  ip router ospf p1 area 0.0.0.0
  ip pim sparse-mode
  no shutdown

interface nve1
  no shutdown
  source-interface loopback1
 member vni 10013001
   mcast-group 227.1.1.1
```

The following examples provide show command outputs related to Port Multi-VLAN Mapping:

```
switch# show hardware access-list resource utilization | grep Super

Ingress PACL Super Bridge               2445    115     95.50
```

```
 Ingress PACL Super Bridge IPv4        0               0.00
 Ingress PACL Super Bridge IPv6        0               0.00
 Ingress PACL Super Bridge MAC         0               0.00
 Ingress PACL Super Bridge ALL         1956            76.40
 Ingress PACL Super Bridge OTHER       489             19.10

switch # show hardware access-list resource entries | in Super

Ingress PACL Super Bridge              : 2445 valid entries   115 free entries

switch# show interface ethernet 1/22.1-5 brief
--------------------------------------------------------------------------------
Ethernet        VLAN    Type Mode   Status  Reason                 Speed     Port
Interface                                                                    Ch #
--------------------------------------------------------------------------------
Eth1/22.1       301     eth  trunk  up      none                      10G(D) --
Eth1/22.2       302     eth  trunk  up      none                      10G(D) --
Eth1/22.3       303     eth  trunk  up      none                      10G(D) --
Eth1/22.4       304     eth  trunk  up      none                      10G(D) --
Eth1/22.5       305     eth  trunk  up      none                      10G(D) --


switch# show interface port-channel 1.1-5 brief
------------------------------------------------------------------------------------------
Port-channel VLAN    Type Mode   Status  Reason     Speed   Protocol
Interface
------------------------------------------------------------------------------------------
Po1.1        301     eth  trunk  up      none        a-10G(D)    --
Po1.2        302     eth  trunk  up      none        a-10G(D)    --
Po1.3        303     eth  trunk  up      none        a-10G(D)    --
Po1.4        304     eth  trunk  up      none        a-10G(D)    --
Po1.5        305     eth  trunk  up      none        a-10G(D)    --

switch# show interface ethernet 1/22.1 counters

----------------------------------------------------------------------------------
Port                                    InOctets                       InUcastPkts
----------------------------------------------------------------------------------
Eth1/22.1                          1145503766466                         125246421

----------------------------------------------------------------------------------
Port                                 InMcastPkts                       InBcastPkts
----------------------------------------------------------------------------------
Eth1/22.1                                      0                                 0

----------------------------------------------------------------------------------
Port                                   OutOctets                      OutUcastPkts
----------------------------------------------------------------------------------
Eth1/22.1                                      0                                 0

----------------------------------------------------------------------------------
Port                                OutMcastPkts                      OutBcastPkts
----------------------------------------------------------------------------------
Eth1/22.1                                      0                                 0

switch# show consistency-checker l2 sub-interface port-channel 1.1

Getting details for port-channel1.1 (0x16001000)
=========================================
Running CC for port-channel1.1
=========================================
CC for Permit Static: PASSED
CC for Deny ACL: PASSED
CC for Permit ARP ACL: PASSED
```

```
CC for Permit Multi-Dest ACL: PASSED
CC for info_src_idx: PASSED
CC for info_bd_xlate_idx: PASSED
CC for info_vlan_mbr_chk_bypasss: PASSED
CC for info_set_dont_learn: PASSED
CC for VlanXlate Table: PASSED
CC for BD State Table: PASSED
CC for QSMT BD State Table: PASSED
CC for Local Multipath Table: PASSED
CC for Rw VifTable: PASSED
CC for Rwx VlanXlate Table: PASSED


switch# show system internal access-list interface eth 1/22.1

slot 1
=======

Policies in ingress direction:
Policy type Policy Id Policy name
------------------------------------------------------------
PACL Super Bridge 341 l2fm-acl-mac-Eth1/22.1
PACL Super Bridge 342 l2fm-acl-ipv6-Eth1/22.1

No Netflow profiles in ingress direction

INSTANCE 0x0
---------------

Tcam 20 resource usage:
----------------------
LBL AB = 0x11
Bank 0
------
IPv6 Class
Policies: PACL Super Bridge(l2fm-acl-ipv6-Eth1/22.1)
Netflow profile: 0
Netflow deny profile: 0
2 tcam entries
MAC Class
Policies: PACL Super Bridge(l2fm-acl-mac-Eth1/22.1)
Netflow profile: 0
Netflow deny profile: 0
3 tcam entries

0 l4 protocol cam entries
0 mac etype/proto cam entries
0 lous
0 tcp flags table entries
0 adjacency entries

No egress policies
No Netflow profiles in egress direction



switch# show system internal access-list interface eth 1/22.1 input statistics

slot 1
=======
INSTANCE 0x0
---------------

Tcam 20 resource usage:
```

```
----------------------
LBL AB = 0xb
Bank 0
------
IPv6 Class
Policies: PACL Super Bridge(l2fm-acl-ipv6-Eth1/22.1)
Netflow profile: 0
Netflow deny profile: 0
Entries:
[Index] Entry [Stats]
---------------------
[0x0038:0x0038:0x0038] permit lbl(0x0) 0000.0000.0000 ffff.ffff.ffff 0000.0000.0000
ffff.ffff.ffff vlan 502 [9]
[0x003a:0x003a:0x003a] permit lbl(0x0) 0000.0000.0000 ffff.ffff.ffff 0000.0000.0000
ffff.ffff.ffff vlan 502 [0]
MAC Class
Policies: PACL Super Bridge(l2fm-acl-mac-Eth1/22.1)
Netflow profile: 0
Netflow deny profile: 0
Entries:
[Index] Entry [Stats]
---------------------
[0x003c:0x003c:0x003c] permit lbl(0x0) arp [7]
[0x003d:0x08de:0x08de] permit lbl(0x0) 0035.0100.0001 ffff.ffff.ffff 0000.0000.0000
ffff.ffff.ffff vlan 502 [6279856]
[0x08dd:0x08e0:0x08e0] deny lbl(0x0) 0000.0000.0000 ffff.ffff.ffff 0000.0000.0000
ffff.ffff.ffff vlan 502 [279]
```

CHAPTER 15

Configuring IGMP Snooping

This chapter contains the following sections:

Configuring IGMP Snooping Over VXLAN

Overview of IGMP Snooping Over VXLAN

By default, multicast traffic over VXLAN is flooded in the VNI/VLAN like any broadcast and unknown unicast traffic. With IGMP snooping enabled, each VTEP can snoop IGMP reports and only forward multicast traffic towards interested receivers.

The configuration of IGMP snooping is the same in VXLAN as in the configuration of IGMP snooping in a regular VLAN domain. For more information on IGMP snooping, see the *Configuring IGMP Snooping* section in the Cisco Nexus 9000 Series NX-OS Multicast Routing Configuration Guide, Release 7.x.

Guidelines and Limitations for IGMP Snooping Over VXLAN

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

- IGMP snooping over VXLAN is not supported on VLANs with FEX member ports.
- IGMP snooping over VXLAN is supported with both IR and multicast underlay.
- IGMP snooping over VXLAN is supported in BGP EVPN topologies, not flood and learn topologies.

Configuring IGMP Snooping Over VXLAN

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)#**ip igmp snooping vxlan**
3. switch(config)#**ip igmp snooping disable-nve-static-router-port**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	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 3	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.

CHAPTER 16

Configuring VLANs

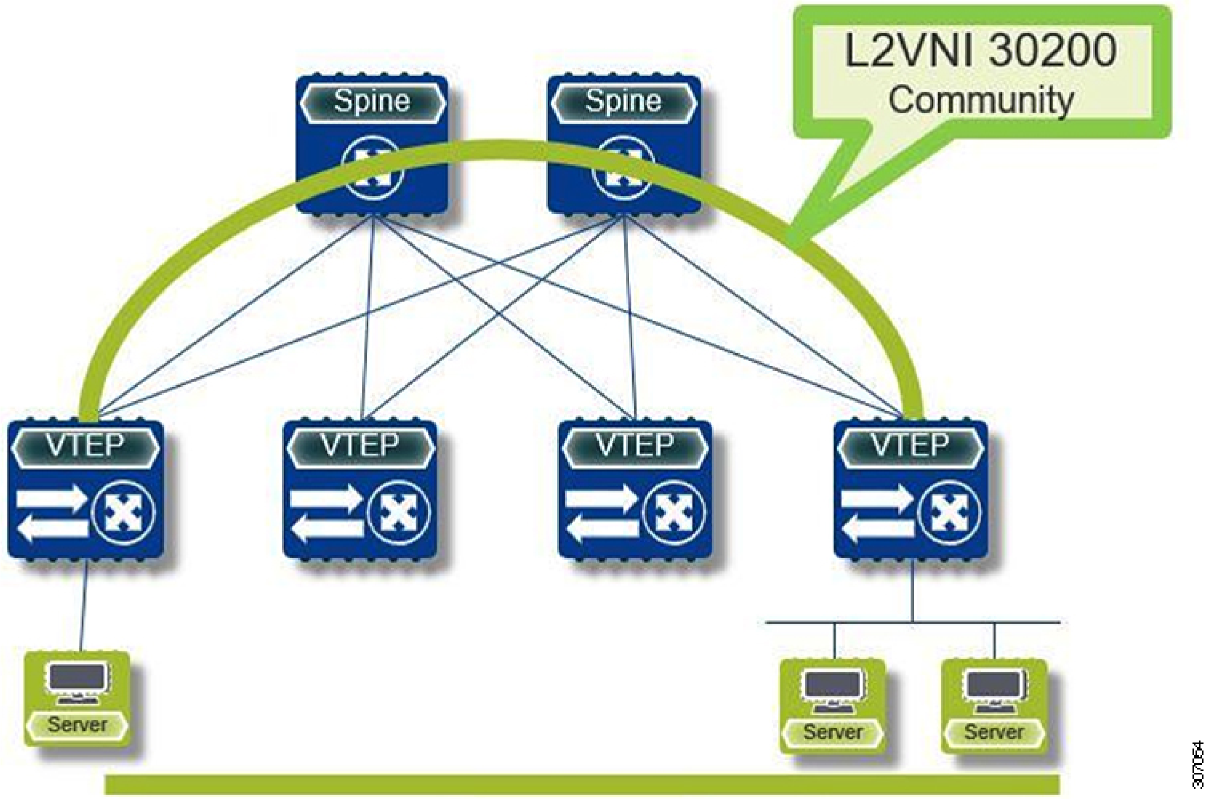
This chapter contains the following sections:

About Private VLANs over VXLAN

The private VLAN feature allows segmenting the Layer 2 broadcast domain of a VLAN into subdomains. A subdomain is represented by a pair of private VLANs: a primary VLAN and a secondary VLAN. A private VLAN domain can have multiple private VLAN pairs, one pair for each subdomain. All VLAN pairs in a private VLAN domain share the same primary VLAN. The secondary VLAN ID differentiates one subdomain from another.

Private VLANs over VXLAN extends private VLAN across VXLAN. The secondary VLAN can exist on multiple VTEPs across VXLAN. MAC address learning happens over the primary VLAN and advertises via BGP EVPN. When traffic is encapsulated, the VNI used is that of the secondary VLAN. The feature also supports Anycast Gateway. Anycast Gateway must be defined using the primary VLAN.

Figure 31: L2VNI 30200 Community



Guidelines and Limitations for Private VLANs over VXLAN

Private VLANs over VXLAN has the following configuration guidelines and limitations:

- The following platforms support private VLANs over VXLAN:
    - Cisco Nexus 9300-EX platform switches
    - Cisco Nexus 9300-FX/FX2 platform switches
    - Cisco Nexus 9300-GX platform switches
- Beginning with Cisco NX-OS Release 9.3(9), PVLAN configuration is not allowed on vPC Peer-link interfaces.
- Flood and learn underlay is not supported.
- Fabric Extenders (FEX) VLAN cannot be mapped to a private VLAN.
- vPC Fabric Peering supports private VLANs.

Configuration Example for Private VLANs

The following is a private VLAN configuration example:

```
vlan 500
  private-vlan primary
  private-vlan association 501-503
  vn-segment 5000
vlan 501
  private-vlan isolated
  vn-segment 5001
vlan 502
  private-vlan community
  vn-segment 5002
vlan 503
  private-vlan community
  vn-segment 5003

vlan 1001
  !L3 VNI for tenant VRF
  vn-segment 900001

interface Vlan500
  no shutdown
  private-vlan mapping 501-503
  vrf member vxlan-900001
  no ip redirects
  ip address 50.1.1.1/8
  ipv6 address 50::1:1:1/64
  no ipv6 redirects
  fabric forwarding mode anycast-gateway

interface Vlan1001
  no shutdown
  vrf member vxlan-900001
  no ip redirects
  ip forward
  ipv6 forward
  ipv6 address use-link-local-only
  no ipv6 redirects

interface nve 1
  no shutdown
  host-reachability protocol bgp
  source-interface loopback0
  member vni 5000
    mcast-group 225.5.0.1
  member vni 5001
    mcast-group 225.5.0.2
  member vni 5002
    ingress-replication protocol bgp
  member vni 5003
    mcast-group 225.5.0.4
member vni 900001 associate-vrf
```

Note If you use an external gateway, the interface towards the external router must be configured as a PVLAN promiscuous port

```
interface ethernet 2/1
switchport
switchport mode private-vlan trunk promiscuous
switchport private-vlan mapping trunk 500 199,200,201
exit
```

CHAPTER 17

Service Redirection in VXLAN Fabrics

This chapter contains the following sections:

Service Redirection in VXLAN EVPN Fabrics

Today, insertion of service appliances (also referred to as service nodes or service endpoints) such as firewalls, load-balancers, etc are needed to secure and optimize applications within a data center. This section describes the Layer 4-Layer 7 service insertion and redirection features offered on VXLAN EVPN fabrics that provides sophisticated mechanisms to onboard and selectively redirect traffic to these services.

Guidelines and Limitations for Policy-Based Redirect

The following guidelines and limitations apply to PBR over VXLAN.

- The following platforms support PBR over VXLAN:
  - Cisco Nexus 9332C and 9364C switches
  - Cisco Nexus 9300-EX switches
  - Cisco Nexus 9300-FX/FX2/FX3 switches
  - Cisco Nexus 9300-GX switches
  - Cisco Nexus 9504 and 9508 switches with -EX/FX line cards
- PBR over VXLAN doesn't support the following features:VTEP ECMP, and the **load-share** keyword in the **set {ip | ipv6} next-hop** *ip-address* command.
- When you configure **bestpath as-path multipath-relax**, BGP installs all the multi-paths for IPv4 and IPv6 as best-path in URIB with metric 0.

Enabling the Policy-Based Redirect Feature

To configure basic PBR, in cases where the advanced (and recommended) ePBR functions are not deployed, see the following sections:

- Enabling the Policy-Based Redirect Feature, on page 334
- Configuring a Route Policy, on page 335
- Verifying the Policy-Based Redirect Configuration, on page 336
- Configuration Example for Policy-Based Redirect, on page 336

Before you begin

Enable the policy-based redirect feature before you can configure a route policy.

SUMMARY STEPS

1. **configure terminal**
2. **[no] feature pbr**
3. (Optional) **show feature**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**[no] feature pbr** **Example:** `switch(config)# feature pbr`	Enables the policy-based routing feature.
Step 3	(Optional) **show feature** **Example:** `switch(config)# show feature`	Displays enabled and disabled features.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Saves this configuration change.

Configuring a Route Policy

You can use route maps in policy-based routing to assign routing policies to the inbound interface. Cisco NX-OS routes the packets when it finds a next hop and an interface.

Note The switch has a RACL TCAM region by default for IPv4 traffic.

Before you begin

Configure the RACL TCAM region (using TCAM carving) before you apply the policy-based routing policy. For instructions, see the “Configuring ACL TCAM Region Sizes” section in the Cisco Nexus 9000 Series NX-OS Security Configuration Guide, Release 9.2(x).

SUMMARY STEPS

1. **configure terminal**
2. **interface** *type slot/port*
3. **{ip | ipv6} policy route-map** *map-name*
4. **route-map** *map-name* **[permit | deny]** [*seq*]
5. **match {ip | ipv6} address access-list-name** *name* [*name*...]
6. **set ip next-hop** *address1*
7. **set ipv6 next-hop** *address1*
8. (Optional) **set interface null0**
9. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**interface** *type slot/port* **Example:** `switch(config)# interface ethernet 1/2`	Enters interface configuration mode.
Step 3	**{ip | ipv6} policy route-map** *map-name* **Example:** `switch(config-inf)# ip policy route-map Testmap`	Assigns a route map for IPv4 or IPv6 policy-based routing to the interface.
Step 4	**route-map** *map-name* **[permit | deny]** [*seq*] **Example:** `switch(config-inf)# route-map Testmap`	Creates a route map or enters route-map configuration mode for an existing route map. Use *seq* to order the entries in a route map.

	Command or Action	Purpose
Step 5	**match {ip | ipv6} address access-list-name** *name* [*name...*] **Example:** `switch(config-route-map)# match ip address access-list-name ACL1`	Matches an IPv4 or IPv6 address against one or more IPv4 or IPv6 access control lists (ACLs). This command is used for policy-based routing and is ignored by route filtering or redistribution.
Step 6	**set ip next-hop** *address1* **Example:** `switch(config-route-map)# set ip next-hop 192.0.2.1`	Sets the IPv4 next-hop address for policy-based routing.
Step 7	**set ipv6 next-hop** *address1* **Example:** `switch(config-route-map)# set ipv6 next-hop 2001:0DB8::1`	Sets the IPv6 next-hop address for policy-based routing.
Step 8	(Optional) **set interface null0** **Example:** `switch(config-route-map)# set interface null0`	Sets the interface that is used for routing. Use the **null0** interface to drop packets.
Step 9	(Optional) **copy running-config startup-config** **Example:** `switch(config-route-map)# copy running-config startup-config`	Saves this configuration change.

Verifying the Policy-Based Redirect Configuration

To display the policy-based redirect configuration information, perform one of the following tasks:

Command	Purpose
show [ip | ipv6] policy [*name*]	Displays information about an IPv4 or IPv6 policy.
show route-map [*name*] **pbr-statistics**	Displays policy statistics.

Use the **route-map** *map-name* **pbr-statistics** command to enable policy statistics. Use the **clear route-map** *map-name* **pbr-statistics** command to clear these policy statistics.

Configuration Example for Policy-Based Redirect

Perform the following configuration on all tenant VTEPs, excluding the service VTEP.

```
feature pbr

ipv6 access-list IPV6_App_group_1
10 permit ipv6 any 2001:10:1:1::0/64

ip access-list IPV4_App_group_1
10 permit ip any 10.1.1.0/24
```

```
ipv6 access-list IPV6_App_group_2
10 permit ipv6 any 2001:20:1:1::0/64

ip access-list IPV4_App_group_2
10 permit ip any 20.1.1.0/24

route-map IPV6_PBR_Appgroup1 permit 10
  match ipv6 address IPV6_App_group_2
  set ipv6 next-hop 2001:100:1:1::20  (next hop is that of the firewall)

route-map IPV4_ PBR_Appgroup1 permit 10
  match ip address IPV4_App_group_2
  set ip next-hop 10.100.1.20 (next hop is that of the firewall)

route-map IPV6_PBR_Appgroup2 permit 10
  match ipv6 address IPV6_App_group1
  set ipv6 next-hop 2001:100:1:1::20  (next hop is that of the firewall)

route-map IPV4_ PBR_Appgroup2 permit 10
  match ip address IPV4_App_group_1
  set ip next-hop 10.100.1.20 (next hop is that of the firewall)


interface Vlan10
! tenant SVI appgroup 1
vrf member appgroup
 ip address 10.1.1.1/24
 no ip redirect
 ipv6 address 2001:10:1:1::1/64
 no ipv6 redirects
 fabric forwarding mode anycast-gateway
ip policy route-map IPV4_ PBR_Appgroup1
ipv6 policy route-map IPV6_PBR_Appgroup1
interface Vlan20
! tenant SVI appgroup 2
vrf member appgroup
 ip address 20.1.1.1/24
 no ip redirect
 ipv6 address 2001:20:1:1::1/64
 no ipv6 redirects
 fabric forwarding mode anycast-gateway
ip policy route-map IPV4_ PBR_Appgroup2
ipv6 policy route-map IPV6_PBR_Appgroup2

On the service VTEP, the PBR policy is applied on the tenant VRF SVI. This ensures the
traffic post decapsulation will be redirected to firewall.
feature pbr

ipv6 access-list IPV6_App_group_1
10 permit ipv6 any 2001:10:1:1::0/64

ip access-list IPV4_App_group_1
10 permit ip any 10.1.1.0/24

ipv6 access-list IPV6_App_group_2
10 permit ipv6 any 2001:20:1:1::0/64

ip access-list IPV4_App_group_2
10 permit ip any 20.1.1.0/24

route-map IPV6_PBR_Appgroup1 permit 10
  match ipv6 address IPV6_App_group_2
  set ipv6 next-hop 2001:100:1:1::20  (next hop is that of the firewall)
```

```
route-map IPV6_PBR_Appgroup permit 20
  match ipv6 address IPV6_App_group1
  set ipv6 next-hop 2001:100:1:1::20  (next hop is that of the firewall)

route-map IPV4_ PBR_Appgroup permit 10
  match ip address IPV4_App_group_2
  set ip next-hop 10.100.1.20 (next hop is that of the firewall)

route-map IPV4_ PBR_Appgroup permit 20
  match ip address IPV4_App_group_1
  set ip next-hop 10.100.1.20 (next hop is that of the firewall)


interface vlan1000
!L3VNI SVI for Tenant VRF
vrf member appgroup
ip forward
ipv6 forward
ipv6 ipv6 address use-link-local-only
ip policy route-map IPV4_ PBR_Appgroup
ipv6 policy route-map IPV6_PBR_Appgroup
```

CHAPTER 18

Configuring ACL

This chapter contains the following sections:

About Access Control Lists

Table 7: ACL Options That Can Be Used for VXLAN Traffic on Cisco Nexus 92300YC, 92160YC-X, 93120TX, 9332PQ, and 9348GC-FXP Switches

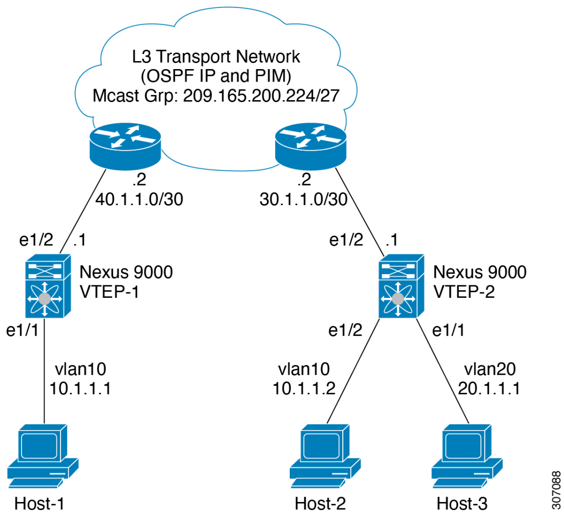
Scenario	ACL Direction	ACL Type	VTEP Type	Port Type	Flow Direction	Traffic Type	Supported
1	Ingress	PACL	Ingress VTEP	L2 port	Access to Network [GROUP:encap direction]	Native L2 traffic [GROUP:inner]	YES
2		VACL	Ingress VTEP	VLAN	Access to Network [GROUP:encap direction]	Native L2 traffic [GROUP:inner]	YES
3	Ingress	RACL	Ingress VTEP	Tenant L3 SVI	Access to Network [GROUP:encap direction]	Native L3 traffic [GROUP:inner]	YES
4	Egress	RACL	Ingress VTEP	uplink L3/L3-PO/SVI	Access to Network [GROUP:encap direction]	VXLAN encap [GROUP:outer]	NO

Scenario	ACL Direction	ACL Type	VTEP Type	Port Type	Flow Direction	Traffic Type	Supported
5	Ingress	RACL	Egress VTEP	Uplink L3/L3-PO/SVI	Network to Access [GROUP:decap direction]	VXLAN encap [GROUP:outer]	NO
6	Egress	PACL	Egress VTEP	L2 port	Network to Access [GROUP:decap direction]	Native L2 traffic [GROUP:inner]	NO
7a		VACL	Egress VTEP	VLAN	Network to Access [GROUP:decap direction]	Native L2 traffic [GROUP:inner]	YES
7b		VACL	Egress VTEP	Destination VLAN	Network to Access [GROUP:decap direction]	Native L3 traffic [GROUP:inner]	YES
8	Egress	RACL	Egress VTEP	Tenant L3 SVI	Network to Access [GROUP:decap direction]	Post-decap L3 traffic [GROUP:inner]	YES

ACL implementation for VXLAN is the same as regular IP traffic. The host traffic is not encapsulated in the ingress direction at the encapsulation switch. The implementation is a bit different for the VXLAN encapsulated traffic at the decapsulation switch as the ACL classification is based on the inner payload. The supported ACL scenarios for VXLAN are explained in the following topics and the unsupported cases are also covered for both encapsulation and decapsulation switches.

All scenarios that are mentioned in the previous table are explained with the following host details:

Figure 32: Port ACL on VXLAN Encap Switch



- Host-1: 10.1.1.1/24 VLAN-10
- Host-2: 10.1.1.2/24 VLAN-10
- Host-3: 20.1.1.1/24 VLAN-20
- Case 1: Layer 2 traffic/L2 VNI that flows between Host-1 and Host-2 on VLAN-10.
- Case 2: Layer 3 traffic/L3 VNI that flows between Host-1 and Host-3 on VLAN-10 and VLAN-20.

Guidelines and Limitations for VXLAN ACLs

VXLAN ACLs have the following guidelines and limitations:

- A router ACL (RACL) on an SVI of the incoming VLAN-10 and the uplink port (eth1/2) does not support filtering the encapsulated VXLAN traffic with outer or inner headers in an egress direction. The limitation also applies to the Layer 3 port-channel uplink interfaces.
- A router ACL (RACL) on an SVI and the Layer 3 uplink ports is not supported to filter the encapsulated VXLAN traffic with outer or inner headers in an ingress direction. This limitation also applies to the Layer 3 port-channel uplink interfaces.
- A port ACL (PACL) cannot be applied on the Layer 2 port to which a host is connected. Cisco NX-OS does not support a PACL in the egress direction.

VXLAN Tunnel Encapsulation Switch

Port ACL on the Access Port on Ingress

You can apply a port ACL (PACL) on the Layer 2 trunk or access port that a host is connected on the encapsulating switch. As the incoming traffic from access to the network is normal IP traffic. The ACL that is being applied on the Layer 2 port can filter it as it does for any IP traffic in the non-VXLAN environment.

The **ing-ifacl** TCAM region must be carved as follows:

SUMMARY STEPS

1. **configure terminal**
2. **hardware access-list tcam region ing-ifacl 256**
3. **ip access-list** *name*
4. *sequence-number* **permit ip** *source-address destination-address*
5. **exit**
6. **interface ethernet** *slot/port*
7. **ip port access-group** *pacl-name***in**
8. **switchport**
9. **switchport mode trunk**
10. **switchport trunk allowed vlan** *vlan-list*
11. **no shutdown**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**hardware access-list tcam region ing-ifacl 256** **Example:** `switch(config)# hardware access-list tcam region ing-ifacl 256`	Attaches the UDFs to the **ing-ifacl** TCAM region, which applies to IPv4 or IPv6 port ACLs.
Step 3	**ip access-list** *name* **Example:** `switch(config)# ip access list PACL_On_Host_Port`	Creates an IPv4 ACL and enters IP ACL configuration mode. The name arguments can be up to 64 characters.
Step 4	*sequence-number* **permit ip** *source-address destination-address* **Example:** `switch(config-acl)# 10 permit ip 10.1.1.1/32 10.1.1.2/32`	Creates an ACL rule that permits or denies IPv4 traffic matching its condition. The *source-address destination-address* arguments can be the IP address with a network wildcard, the IP address

	Command or Action	Purpose
		and variable-length subnet mask, the host address, and **any** to designate any address.
Step 5	**exit** **Example:** `switch(config-acl)# exit`	Exits IP ACL configuration mode.
Step 6	**interface ethernet** *slot/port* **Example:** `switch(config)# interface ethernet1/1`	Enters interface configuration mode.
Step 7	**ip port access-group** *pacl-name***in** **Example:** `switch(config-if)# ip port access-group PACL_On_Host_Port in`	Applies a Layer 2 PACL to the interface. Only inbound filtering is supported with port ACLs. You can apply one port ACL to an interface.
Step 8	**switchport** **Example:** `switch(config-if)# switchport`	Configures the interface as a Layer 2 interface.
Step 9	**switchport mode trunk** **Example:** `switch(config-if)# switchport mode trunk`	Configures the interface as a Layer 2 trunk port.
Step 10	**switchport trunk allowed vlan** *vlan-list* **Example:** `switch(config-if)# switchport trunk allowed vlan 10,20`	Sets the allowed VLANs for the trunk interface. The default is to allow all VLANs on the trunk interface, 1 through 3967 and 4048 through 4094. VLANs 3968 through 4047 are the default VLANs reserved for internal use.
Step 11	**no shutdown** **Example:** `switch(config-if)# no shutdown`	Negates the **shutdown** command.

VLAN ACL on the Server VLAN

A VLAN ACL (VACL) can be applied on the incoming VLAN-10 that the host is connected to on the encap switch. As the incoming traffic from access to network is normal IP traffic, the ACL that is being applied to VLAN-10 can filter it as it does for any IP traffic in the non-VXLAN environment. For more information on VACL, see About Access Control Lists, on page 339.

SUMMARY STEPS

1. **configure terminal**
2. **ip access-list** *name*
3. *sequence-number* **permit ip** *source-address destination-address*

4. **vlan access-map** *map-name* **[***sequence-number***]**
5. **match ip address** *ip-access-list*
6. **action forward**
7. **vlan access-map** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**ip access-list** *name* **Example:** `switch(config)# ip access list Vacl_On_Source_VLAN`	Creates an IPv4 ACL and enters IP ACL configuration mode. The name arguments can be up to 64 characters.
Step 3	*sequence-number* **permit ip** *source-address destination-address* **Example:** `switch(config-acl)# 10 permit ip 10.1.1.1 10.1.1.2`	Creates an ACL rule that permits or denies IPv4 traffic matching its condition. The *source-address destination-address* arguments can be the IP address with a network wildcard, the IP address and variable-length subnet mask, the host address, and **any** to designate any address.
Step 4	**vlan access-map** *map-name* **[***sequence-number***]** **Example:** `switch(config-acl)# vlan access-map Vacl_on_Source_Vlan 10`	Enters VLAN access-map configuration mode for the VLAN access map specified. If the VLAN access map does not exist, the device creates it. If you do no specify a sequence number, the device creates a new entry whose sequence number is 10 greater than the last sequence number in the access map.
Step 5	**match ip address** *ip-access-list* **Example:** `switch(config-acl)# match ip address Vacl_on_Source_Vlan`	Specifies an ACL for the access-map entry.
Step 6	**action forward** **Example:** `switch(config-acl)# action forward`	Specifies the action that the device applies to traffic that matches the ACL.
Step 7	**vlan access-map** *name* **Example:** `switch(config-acl)# vlan access map Vacl_on_Source_Vlan`	Enters VLAN access-map configuration mode for the VLAN access map specified.

Routed ACL on an SVI on Ingress

A router ACL (RACL) in the ingress direction can be applied on an SVI of the incoming VLAN-10 that the host that connects to the encapsulating switch. As the incoming traffic from access to network is normal IP traffic, the ACL that is being applied on SVI 10 can filter it as it does for any IP traffic in the non-VXLAN environment.

The **ing-racl** TCAM region must be carved as follows:

SUMMARY STEPS

1. **configure terminal**
2. **hardware access-list tcam region ing-ifacl 256**
3. **ip access-list** *name*
4. *sequence-number* **permit ip** *source-address destination-address*
5. **exit**
6. **interface ethernet** *slot/port*
7. **no shutdown**
8. **ip access-group** *pacl-name***in**
9. **vrf member** *vxlan-number*
10. **no ip redirects**
11. **ip address** *ip-address*
12. **no ipv6 redirects**
13. **fabric forwarding mode anycast-gateway**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**hardware access-list tcam region ing-ifacl 256** **Example:** `switch(config)# hardware access-list tcam region ing-ifacl 256`	Attaches the UDFs to the **ing-racl** TCAM region, which applies to IPv4 or IPv6 port ACLs.
Step 3	**ip access-list** *name* **Example:** `switch(config)# ip access list PACL_On_Host_Port`	Creates an IPv4 ACL and enters IP ACL configuration mode. The name arguments can be up to 64 characters.
Step 4	*sequence-number* **permit ip** *source-address destination-address* **Example:** `switch(config-acl)# 10 permit ip 10.1.1.1/32 10.1.1.2/32`	Creates an ACL rule that permits or denies IPv4 traffic matching its condition. The *source-address destination-address* arguments can be the IP address with a network wildcard, the IP address and variable-length subnet mask, the host address, and **any** to designate any address.

	Command or Action	Purpose
Step 5	**exit** **Example:** `switch(config-acl)# exit`	Exits IP ACL configuration mode.
Step 6	**interface ethernet** *slot/port* **Example:** `switch(config)# interface ethernet1/1`	Enters interface configuration mode.
Step 7	**no shutdown** **Example:** `switch(config-if)# no shutdown`	Negates **shutdown** command.
Step 8	**ip access-group** *pacl-name***in** **Example:** `switch(config-if)# ip port access-group Racl_On_Source_Vlan_SVI in`	Applies a Layer 2 PACL to the interface. Only inbound filtering is supported with port ACLs. You can apply one port ACL to an interface.
Step 9	**vrf member** *vxlan-number* **Example:** `switch(config-if)# vrf member Cust-A`	Configure SVI for host.
Step 10	**no ip redirects** **Example:** `switch(config-if)# no ip redirects`	Prevents the device from sending redirects.
Step 11	**ip address** *ip-address* **Example:** `switch(config-if)# ip address 10.1.1.10`	Configures an IP address for this interface.
Step 12	**no ipv6 redirects** **Example:** `switch(config-if)# no ipv6 redirects`	Disables the ICMP redirect messages on BFD-enabled interfaces.
Step 13	**fabric forwarding mode anycast-gateway** **Example:** `switch(config-if)# fabric forwarding mode anycast-gateway`	Configure Anycast gateway forwarding mode.

Routed ACL on the Uplink on Egress

A RACL on an SVI of the incoming VLAN-10 and the uplink port (eth1/2) is not supported to filter the encapsulated VXLAN traffic with an outer or inner header in an egress direction. This limitation also applies to the Layer 3 port-channel uplink interfaces.

VXLAN Tunnel Decapsulation Switch

Routed ACL on the Uplink on Ingress

A RACL on a SVI and the Layer 3 uplink ports is not supported to filter the encapsulated VXLAN traffic with outer or inner header in an ingress direction. This limitation also applies to the Layer 3 port-channel uplink interfaces.

Port ACL on the Access Port on Egress

Do not apply a PACL on the Layer 2 port to which a host is connected. Cisco Nexus 9000 Series switches do not support a PACL in the egress direction.

VLAN ACL for the Layer 2 VNI Traffic

A VLAN ACL (VACL) can be applied on VLAN-10 to filter with the inner header when the Layer 2 VNI traffic is flowing from Host-1 to Host-2. For more information on VACL, see About Access Control Lists, on page 339.

The VACL TCAM region must be carved as follows:

SUMMARY STEPS

1. **configure terminal**
2. **hardware access-list tcam region vacl 256**
3. **ip access-list** *name*
4. **statistics per-entry**
5. *sequence-number* **permit ip** *source-address destination-address*
6. *sequence-number* **permit** *protocol source-address destination-address*
7. **exit**
8. **vlan access-map** *map-name* **[***sequence-number***]**
9. **match ip address** *list-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**hardware access-list tcam region vacl 256** **Example:** `switch(config)# hardware access-list tcam region vacl 256`	Changes the ACL TCAM region size.

	Command or Action	Purpose
Step 3	**ip access-list** *name* **Example:** `switch(config)# ip access list VXLAN-L2-VNI`	Creates an IPv4 ACL and enters IP ACL configuration mode. The name arguments can be up to 64 characters.
Step 4	**statistics per-entry** **Example:** `switch(config-acl)# statistics per-entry`	Specifies that the device maintains global statistics for packets that match the rules in the VACL.
Step 5	*sequence-number* **permit ip** *source-address destination-address* **Example:** `switch(config-acl)# 10 permit ip 10.1.1.1/32 10.1.1.2/32`	Creates an ACL rule that permits or denies IPv4 traffic matching its condition. The *source-address destination-address* arguments can be the IP address with a network wildcard, the IP address and variable-length subnet mask, the host address, and **any** to designate any address.
Step 6	*sequence-number* **permit** *protocol source-address destination-address* **Example:** `switch(config-acl)# 20 permit tcp 10.1.1.2/32 10.1.1.1/32`	Creates an ACL rule that permits or denies IPv4 traffic matching its condition. The *source-address destination-address* arguments can be the IP address with a network wildcard, the IP address and variable-length subnet mask, the host address, and **any** to designate any address.
Step 7	**exit** **Example:** `switch(config-acl)# exit`	Exit ACL configuration mode.
Step 8	**vlan access-map** *map-name* **[***sequence-number***]** **Example:** `switch(config)# vlan access-map VXLAN-L2-VNI 10`	Enters VLAN access-map configuration mode for the VLAN access map specified. If the VLAN access map does not exist, the device creates it. If you do no specify a sequence number, the device creates a new entry whose sequence number is 10 greater than the last sequence number in the access map.
Step 9	**match ip address** *list-name* **Example:** `switch(config-access-map)# match ip VXLAN-L2-VNI`	Configure the IP list name.

VLAN ACL for the Layer 3 VNI Traffic

A VLAN ACL (VACL) can be applied on the destination VLAN-20 to filter with the inner header when the Layer 3 VNI traffic is flowing from Host-1 to Host-3. It slightly differs from the previous case as the VACL for the Layer 3 traffic is accounted on the egress on the system. The keyword **output** must be used while dumping the VACL entries for the Layer 3 VNI traffic. For more information on VACL, see About Access Control Lists, on page 339.

The VACL TCAM region must be carved as follows.

SUMMARY STEPS

1. **configure terminal**
2. **hardware access-list tcam region vacl 256**
3. **ip access-list** *name*
4. **statistics per-entry**
5. *sequence-number* **permit ip** *source-address destination-address*
6. *sequence-number* **permit** *protocol source-address destination-address*
7. **vlan access-map** *map-name* **[***sequence-number***]**
8. **action forward**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**hardware access-list tcam region vacl 256** **Example:** `switch(config)# hardware access-list tcam region vacl 256`	Changes the ACL TCAM region size.
Step 3	**ip access-list** *name* **Example:** `switch(config)# ip access list VXLAN-L3-VNI`	Creates an IPv4 ACL and enters IP ACL configuration mode. The name arguments can be up to 64 characters.
Step 4	**statistics per-entry** **Example:** `switch(config)# statistics per-entry`	Specifies that the device maintains global statistics for packets that match the rules in the VACL.
Step 5	*sequence-number* **permit ip** *source-address destination-address* **Example:** `switch(config-acl)# 10 permit ip 10.1.1.1/32 20.1.1.1/32`	Creates an ACL rule that permits or denies IPv4 traffic matching its condition. The *source-address destination-address* arguments can be the IP address with a network wildcard, the IP address and variable-length subnet mask, the host address, and **any** to designate any address.
Step 6	*sequence-number* **permit** *protocol source-address destination-address* **Example:** `switch(config-acl)# 20 permit tcp 20.1.1.1/32 10.1.1.1/32`	Configures the ACL to redirect-specific HTTP methods to a server.

	Command or Action	Purpose
Step 7	**vlan access-map** *map-name* **[***sequence-number***]** **Example:** `switch(config-acl)# vlan access-map VXLAN-L3-VNI 10`	Enters VLAN access-map configuration mode for the VLAN access map specified. If the VLAN access map does not exist, the device creates it. If you do no specify a sequence number, the device creates a new entry whose sequence number is 10 greater than the last sequence number in the access map.
Step 8	**action forward** **Example:** `switch(config-acl)# action forward`	Specifies the action that the device applies to traffic that matches the ACL.

Routed ACL on an SVI on Egress

A router ACL (RACL) on the egress direction can be applied on an SVI of the destination VLAN-20 that Host-3 is connected to on the decap switch to filter with the inner header for traffic flows from the network to access which is normal post-decapsulated IP traffic post. The ACL that is being applied on SVI 20 can filter it as it does for any IP traffic in the non-VXLAN environment. For more information on ACL, see About Access Control Lists, on page 339.

The egr-racl TCAM region must be carved as follows:

SUMMARY STEPS

1. **configure terminal**
2. **hardware access-list tcam region egr-racl 256**
3. **ip access-list** *name*
4. *sequence-number* **permit ip** *source-address destination-address*
5. **interface vlan** *vlan-id*
6. **no shutdown**
7. **ip access-group** *access-list* **out**
8. **vrf member** *vxlan-number*
9. **no ip redirects**
10. **ip address** *ip-address/length*
11. **no ipv6 redirects**
12. **fabric forwarding mode anycast-gateway**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**hardware access-list tcam region egr-racl 256** **Example:**	Changes the ACL TCAM region size.

	Command or Action	Purpose
	`switch(config)# hardware access-list tcam region egr-racl 256`	
Step 3	**ip access-list** *name* **Example:** `switch(config)# ip access-list Racl_on_Source_Vlan_SVI`	Creates an IPv4 ACL and enters IP ACL configuration mode. The name arguments can be up to 64 characters.
Step 4	*sequence-number* **permit ip** *source-address destination-address* **Example:** `switch(config-acl)# 10 permit ip 10.1.1.1/32 20.1.1.1/32`	Creates an ACL rule that permits or denies IPv4 traffic matching its condition. The *source-address destination-address* arguments can be the IP address with a network wildcard, the IP address and variable-length subnet mask, the host address, and **any** to designate any address.
Step 5	**interface vlan** *vlan-id* **Example:** `switch(config-acl)# interface vlan vlan20`	Enters interface configuration mode, where *vlan-id* is the ID of the VLAN that you want to configure with a DHCP server IP address.
Step 6	**no shutdown** **Example:** `switch(config-if)# no shutdown`	Negate the shutdown command.
Step 7	**ip access-group** *access-list* **out** **Example:** `switch(config-if)# ip access-group Racl_On_Detination_Vlan_SVI out`	Applies an IPv4 or IPv6 ACL to the Layer 3 interfaces for traffic flowing in the direction specified. You can apply one router ACL per direction.
Step 8	**vrf member** *vxlan-number* **Example:** `switch(config-if)# vrf member Cust-A`	Configure SVI for host.
Step 9	**no ip redirects** **Example:** `switch(config-if)# no ip redirects`	Prevents the device from sending redirects.
Step 10	**ip address** *ip-address/length* **Example:** `switch(config-if)# ip address 20.1.1.10/24`	Configures an IP address for this interface.
Step 11	**no ipv6 redirects** **Example:** `switch(config-if)# no ipv6 redirects`	Disables the ICMP redirect messages on BFD-enabled interfaces.
Step 12	**fabric forwarding mode anycast-gateway** **Example:**	Configure Anycast gateway forwarding mode.

	Command or Action	Purpose
	switch(config-if)# **fabric forwarding mode anycast-gateway**	

CHAPTER 19

Configuring Secure VXLAN EVPN Multi-Site Using CloudSec

This chapter contains the following sections:

About Secure VXLAN EVPN Multi-Site Using CloudSec

Secure VXLAN EVPN Multi-Site using CloudSec ensures data security and data integrity for VXLAN-based Multi-Site fabrics. Using the cryptographic machinery of IEEE MACsec for UDP packets, this feature provides a secure tunnel between authorized VXLAN EVPN endpoints.

The CloudSec session is point to point over DCI between border gateways (BGWs) on two different sites. All communication between sites uses Multi-Site PIP instead of VIP. For migration information, see Migrating from Multi-Site with VIP to Multi-Site with PIP, on page 370.

Secure VXLAN EVPN Multi-Site using CloudSec is enabled on a per-peer basis. Peers that do not support CloudSec can operate with peers that do support CloudSec, but the traffic is unencrypted. We recommend allowing unencrypted traffic only during migration from non-CloudSec-enabled sites to CloudSec-enabled sites.

CloudSec key exchange uses BGP while MACsec uses the MACsec Key Agreement (MKA). The CloudSec control plane uses the BGP IPv4 address family to exchange the key information. CloudSec keys are carried as part of Tunnel Encapsulation (tunnel type 18) attribute with BGP IPv4 routes using underlay BGP session.

Key Lifetime and Hitless Key Rollover

A CloudSec keychain can have multiple pre-shared keys (PSKs), each configured with a key ID and an optional lifetime. Pre-shared keys are seed keys used to derive further keys for traffic encryption and integrity validation. A list of pre-shared keys can be configured in a keychain with different lifetimes.

A key lifetime specifies when the key expires. CloudSec rolls over to the next configured pre-shared key in the keychain after the lifetime expires. The time zone of the key can be local or UTC. The default time zone is UTC. In the absence of a lifetime configuration, the default lifetime is unlimited.

To configure the CloudSec keychain, see Configuring a CloudSec Keychain and Keys, on page 358.

When the lifetime of the first key expires, it automatically rolls over to the next key in the list. If the same key is configured on both sides of the link at the same time, the key rollover is hitless. That is, the key rolls over without traffic interruption. The lifetime of the keys must be overlapped in order to achieve hitless key rollover.

Guidelines and Limitations for Secure VXLAN EVPN Multi-Site Using CloudSec

Secure VXLAN EVPN Multi-Site using CloudSec has the following guidelines and limitations:

- Secure VXLAN EVPN Multi-Site using CloudSec is supported on Cisco Nexus 9300-FX2 platform switches beginning with Cisco NX-OS Release 9.3(5).
- Secure VXLAN EVPN Multi-Site using CloudSec is supported on Cisco Nexus 9300-FX3 platform switches from Cisco NX-OS Release 10.1(1) onwards.
- L3 interfaces and L3 port channels are supported as DCI links.
- CloudSec traffic that is destined for the switch must enter the switch through the DCI uplinks.
- Secure VXLAN EVPN Multi-Site using CloudSec is supported for sites that are connected through a route server or sites that are connected using full mesh (without a route server). For sites that are connected through a route server, upgrade the server to Cisco NX-OS Release 9.3(5) or a later release and follow the instructions in Enabling CloudSec VXLAN EVPN Tunnel Encryption, on page 355.
- Beginning with Cisco NX-OS Release 10.1(1), VXLAN Tunnel Encryption feature is supported on Cisco Nexus 9300-FX3 platform switches.
- ICV is disabled by default in Cisco NX-OS Release 9.3(7). ICV should be disabled on the node when forming cloudsec tunnel sessions with node from the previous release (Cisco NX-OS Release 9.3(6)).
- All of the BGWs on the same site should be configured for Secure VXLAN EVPN Multi-Site using CloudSec.
- Secure VXLAN EVPN Multi-Site using CloudSec on DCI links and MACsec on the internal fabric can coexist. However, they can't be enabled simultaneously on the same port or port group (MAC ID).
- Secure VXLAN EVPN Multi-Site using CloudSec peers must have the same keychain configuration in order to decrypt the secure traffic between them.
- A maximum of 60 peers are supported in the BGP IPv4 update of security key distribution in the Cisco Nexus 9300-FX2 family switches.

- In order to keep a session alive when all keys with an active timer expire, configure no more than one key per keychain without a lifetime. As a best practice, we recommend configuring a lifetime for each key.
- CloudSec keys are exchanged between BGWs using Tunnel Encapsulation attribute with BGP IPv4 routes using underlay BGP session.

  If this attribute do not get propagated by intermediate nodes, you have to configure direct BGP IPv4 unicast session between the CloudSec end point nodes i.e., BGWs.
- Direct eBGP peering must be established between BGWs in each site if:
  - BGP is used as the IPv4 unicast routing protocol, but the Tunnel Encryption attribute is not propagated through DCI.
  - A routing protocol other than BGP is used for IPv4 unicast routing in the DCI (e.g., OSPF).
- eBGP peering is to be established over a Loopback interface that is different from the following interface:
  - The tunnel-encryption source-interface
  - The nve source-interface
- eBGP peering must filter the loopback IP used as the source of the adjacency. For example, if Loopback10 is used to establish eBGP peering for CloudSec, the IP of Lo10 should not be advertised over this adjacency.
- Secure VXLAN EVPN Multi-Site using CloudSec doesn't support the following:
  - Directly connected L2 hosts on border gateways
  - IP unnumbered configurations on the DCI interface
  - Multicast underlay
  - OAM pathtrace
  - TRM
  - VIP-only model on border gateways
  - VXLAN EVPN with downstream VNI
- Beginning with Cisco NX-OS Release 10.3(1), vPC cloudsec with DSVNI is not supported on Cisco Nexus 9000 Series switches.
- If CloudSec is enabled, non-disruptive ISSU is not supported.

Configuring Secure VXLAN EVPN Multi-Site Using CloudSec

Follow these procedures to configure Secure VXLAN EVPN Multi-Site using CloudSec:

Enabling CloudSec VXLAN EVPN Tunnel Encryption

Follow these steps to enable CloudSec VXLAN EVPN Tunnel Encryption.

Before you begin

Configure BGP peers in the IPv4 unicast address family. Make sure that the IPv4 prefix is propagated with the tunnel community attribute that carries CloudSec keys.

Configure VXLAN EVPN Multi-Site and use the following commands to ensure that peer IP addresses are advertised for CloudSec VXLAN EVPN Tunnel Encryption:

```
evpn multisite border-gateway ms-id
dci-advertise-pip
```

> **Caution** Configuring VXLAN EVPN Multi-Site without **dci-advertise-pip** reverts border gateways to VIP-only mode, which is not supported for CloudSec VXLAN EVPN Tunnel Encryption.

You have two options for sites that are connected through a route server:

- Keep dual RDs enabled – This default behavior ensures that the memory scale remains the same from previous releases in order to handle leaf devices with limited memory. All same-site BGWs use the same RD value for reoriginated routes while advertising EVPN routes to the remote BGW.
- Disable dual RDs – If you don't have memory limitations on leaf devices, you can configure the **no dual rd** command on the BGW. Different RD values are used for reoriginated routes on the same BGWs while advertising EVPN routes to the remote BGW.

Perform one of the following actions, depending on whether dual RDs are enabled on the BGW:

- If dual RDs are configured on the BGWs, follow these steps:

1. Apply BGP additional paths on the BGW.

```
router bgp as-num
    address-family l2vpn evpn
      maximum-paths number
    additional-paths send
    additional-paths receive
```

2. Configure multipath for each L3VNI VRF on the BGW.

```
vrf evpn-tenant-00001
  address-family ipv4 unicast
    maximum-paths 64
  address-family ipv6 unicast
    maximum-paths 64
```

3. Apply BGP additional paths on the route server.

```
router bgp as-num
  address-family l2vpn evpn
    retain route-target all
    additional-paths send
    additional-paths receive
    additional-paths selection route-map name

route-map name permit 10
  set path-selection all advertise
```

- If **no dual rd** is configured on the BGWs or full mesh is configured, follow these steps:

  1. Configure the address family and maximum paths on the BGW.

     ```
     router bgp as-num
       address-family l2vpn evpn
         maximum-paths number
     ```

  2. Configure multipath for each L3VNI VRF on the BGW.

     ```
     vrf evpn-tenant-00001
         address-family ipv4 unicast
           maximum-paths 64
         address-family ipv6 unicast
           maximum-paths 64
     ```

Note BGP additional paths are not required on the route server.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **feature tunnel-encryption**
3. [**no**] **tunnel-encryption source-interface loopback** *number*
4. **tunnel-encryption icv**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	[**no**] **feature tunnel-encryption** **Example:** `switch(config)# feature tunnel-encryption`	Enables CloudSec VXLAN EVPN Tunnel Encryption.
Step 3	[**no**] **tunnel-encryption source-interface loopback** *number* **Example:** `switch(config)# tunnel-encryption source-interface loopback 2`	Specifies the BGP loopback as the tunnel-encryption source interface. The IP address of the configured source interface is used as the prefix to announce CloudSec VXLAN EVPN Tunnel Encryption key routes. **Note** Enter the BGP loopback interface and not the NVE source interface. **Note** Any changes in the MTU should be done before the tunnel-encryption configuration on the interface. This will avoid the CRC drop errors.

	Command or Action	Purpose
Step 4	**tunnel-encryption icv** **Example:** `switch(config)# tunnel-encryption icv`	Enables the Integrity Check Value (ICV). ICV provides integrity check for the frame arriving on the port. If the generated ICV is the same as the ICV in the frame, then the frame is accepted; otherwise it is dropped.This is supported from Cisco NX-OS Release 9.3(7) onwards.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

What to do next

After enabling CloudSec VXLAN EVPN tunnel encryption, you can follow any of the following procedure for authentication.

Configuring a CloudSec Keychain and Keys.

Configuring a CloudSec Keychain and Keys

You can create a CloudSec keychain and keys on the device.

Before you begin

Make sure that Secure VXLAN EVPN Multi-Site using CloudSec is enabled.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **key chain** *name* **tunnel-encryption**
3. [**no**] **key** *key-id*
4. [**no**] **key-octet-string** *octet-string* **cryptographic-algorithm** {**AES_128_CMAC** | **AES_256_CMAC**}
5. [**no**] **send-lifetime** *start-time* **duration** *duration*
6. (Optional) **show key chain** *name*
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	[**no**] **key chain** *name* **tunnel-encryption**
	Example:
	`switch(config)# key chain kc1 tunnel-encryption`
	`switch(config-tunnelencryptkeychain)#`
		Creates a CloudSec keychain to hold a set of CloudSec keys and enters tunnel-encryption keychain configuration mode.

	Command or Action	Purpose

Step 3	[**no**] **key** *key-id*
	Example:
	`switch(config-tunnelencryptkeychain)# key 2000`
	`switch(config-tunnelencryptkeychain-tunnelencryptkey)#`
		Creates a CloudSec key and enters tunnel-encryption key configuration mode. The range is from 1 to 32 octets, and the maximum size is 64.
		Note The key must consist of an even number of characters.

Step 4	[**no**] **key-octet-string** *octet-string* **cryptographic-algorithm** {**AES_128_CMAC** | **AES_256_CMAC**} **Example:** `switch(config-tunnelencryptkeychain-tunnelencryptkey)# key-octet-string abcdef0123456789abcdef0123456789 abcdef0123456789abcdef0123456789 cryptographic-algorithm AES_256_CMAC`	Configures the octet string for the key. The *octet-string* argument can contain up to 64 hexadecimal characters. The octet key is encoded internally, so the key in clear text does not appear in the output of the **show running-config tunnel-encryption** command.
Step 5	[**no**] **send-lifetime** *start-time* **duration** *duration* **Example:** `switch(config-tunnelencryptkeychain-tunnelencryptkey)# send-lifetime 00:00:00 May 06 2020 duration 100000`	Configures a send lifetime for the key. By default, the device treats the start time as UTC. The *start-time* argument is the time of day and date that the key becomes active. The *duration* argument is the length of the lifetime in seconds. The range is from 1800 seconds to 2147483646 seconds (approximately 68 years).
Step 6	(Optional) **show key chain** *name* **Example:** `switch(config-tunnelencryptkeychain-tunnelencryptkey)# show key chain kc1`	Displays the keychain configuration.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch(config-tunnelencryptkeychain-tunnelencryptkey)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

What to do next

Configuring a CloudSec Policy.

Configuring a CloudSec Policy

You can create multiple CloudSec policies with different parameters. However, only one policy can be active on an interface.

Before you begin

Make sure that Secure VXLAN EVPN Multi-Site using CloudSec is enabled.

SUMMARY STEPS

1. **configure terminal**
2. (Optional) [**no**] **tunnel-encryption must-secure-policy**
3. [**no**] **tunnel-encryption policy** *name*
4. (Optional) [**no**] **cipher-suite** *name*
5. (Optional) [**no**] **window-size** *number*
6. (Optional) [**no**] **sak-rekey-time** *time*
7. (Optional) **show tunnel-encryption policy**
8. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	(Optional) [**no**] **tunnel-encryption must-secure-policy** **Example:** `switch(config)# tunnel-encryption must-secure-policy`	Ensures that no unencrypted packets are sent over the wire for the session. Packets that are not carrying CloudSec headers are dropped. The **no** form of this command allows unencrypted traffic. We recommend allowing unencrypted traffic only during migration from non-CloudSec-enabled sites to CloudSec-enabled sites. By default, Secure VXLAN EVPN Multi-Site using CloudSec operates in "should secure" mode.

Step 3	[**no**] **tunnel-encryption policy** *name*
	Example:
	`switch(config)# tunnel-encryption policy p1`
	`switch(config-tunenc-policy)#`
		Creates a CloudSec policy.

Step 4	(Optional) [**no**] **cipher-suite** *name* **Example:** `switch(config-tunenc-policy)# cipher-suite GCM-AES-XPN-256`	Configures one of the following ciphers: GCM-AES-XPN-128 or GCM-AES-XPN-256. The default value is GCM-AES-XPN-256.
Step 5	(Optional) [**no**] **window-size** *number* **Example:** `switch(config-tunenc-policy)# window-size 134217728`	Configures the replay protection window such that the interface will not accept any packet that is less than the configured window size. The range is from 134217728 to 1073741823 IP packets. The default value is 268435456.
Step 6	(Optional) [**no**] **sak-rekey-time** *time* **Example:** `switch(config-tunenc-policy)# sak-rekey-time 1800`	Configures the time in seconds to force an SAK rekey. This command can be used to change the session key to a predictable time interval. The range is from 1800 to 2592000 seconds. There is not a default value. We recommend using the same rekey value for all the peers.

	Command or Action	Purpose
Step 7	(Optional) **show tunnel-encryption policy** **Example:** `switch(config-tunenc-policy)# show tunnel-encryption policy`	Displays the CloudSec policy configuration.
Step 8	(Optional) **copy running-config startup-config** **Example:** `switch(config-tunenc-policy)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

What to do next

Configuring CloudSec Peers.

Configuring CloudSec Peers

This chapter contains the following sections.

Configuring CloudSec Peers

You can configure the CloudSec peers.

Before you begin

Enable Secure VXLAN EVPN Multi-Site using CloudSec.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **tunnel-encryption peer-ip** *peer-ip-address*
3. [**no**] **keychain** *name* **policy** *name*
4. **pki policy** *policy name*

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	[**no**] **tunnel-encryption peer-ip** *peer-ip-address* **Example:** `switch(config)# tunnel-encryption peer-ip 33.1.33.33`	Specifies the IP address of the NVE source interface on the peer.

	Command or Action	Purpose
Step 3	[**no**] **keychain** *name* **policy** *name* **Example:** `switch(config)# keychain kc1 policy p1`	Attaches a policy to a CloudSec peer. Step 4 is an alternative to this step.
Step 4	**pki policy** *policy name* **Example:** `switch(config)# pki policy p1`	Attaching cloudsec policy to peer with PKI.

What to do next

Enabling Secure VXLAN EVPN Multi-Site Using CloudSec on DCI Uplinks.

Enabling Secure VXLAN EVPN Multi-Site Using CloudSec on DCI Uplinks

Follow these steps to enable Secure VXLAN EVPN Multi-Site using CloudSec on all DCI uplinks.

Note This configuration cannot be applied on Layer 2 ports.

Note When CloudSec is applied or removed from an operational DCI uplink, the link will flap. The flap may not be instantaneous as the link may remain down for several seconds.

Before you begin

Make sure that Secure VXLAN EVPN Multi-Site using CloudSec is enabled.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **interface ethernet** *port/slot*
3. [**no**] **tunnel-encryption**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	[**no**] **interface ethernet** *port/slot* **Example:**	Enters interface configuration mode.

	Command or Action	Purpose
	switch(config)# **interface ethernet 1/1** switch(config-if)#	
Step 3	[**no**] **tunnel-encryption** **Example:** switch(config-if)# **tunnel-encryption**	Enables Secure VXLAN EVPN Multi-Site using CloudSec on the specified interface.

Verifying the Secure VXLAN EVPN Multi-Site Using CloudSec

To display Secure VXLAN EVPN Multi-Site using CloudSec configuration information, perform one of the following tasks:

Command	Purpose
show tunnel-encryption info global	Displays configuration information for Secure VXLAN EVPN Multi-Site using CloudSec.
show tunnel-encryption policy [*policy-name*]	Displays the configuration for a specific CloudSec policy or for all CloudSec policies.
show tunnel-encryption session [**peer-ip** *peer-ip-address*] [**detail**]	Displays information about CloudSec sessions, including whether sessions are secure between endpoints.
show running-config tunnel-encryption	Displays the running configuration information for Secure VXLAN EVPN Multi-Site using CloudSec.
show bgp ipv4 unicast *ip-address*	Displays the tunnel encryption information for BGP routes.
show bgp l2vpn evpn	Displays the Layer 2 VPN EVPN address family and routing table information.
show ip route *ip-address* **vrf** *vrf*	Displays the VRF routes.
show l2route evpn mac evi *evi*	Displays Layer 2 route information.
show nve interface *interface* **detail**	Displays the NVE interface detail.
show running-config rpm	Displays the key text in the running configuration. **Note** If you enter the **key-chain tunnelencrypt-psk no-show** command prior to running this command, the key text is hidden (with asterisks) in the running configuration. If you enter the **reload ascii** command, the key text is omitted from the running configuration.
show running-config cert-enroll	Shows the trustpoint and keypair configuration.

Command	Purpose
show crypto ca certificates *<trustpoint_label>*	Shows the certificate contents under a trustpoint.

The following example displays configuration information for Secure VXLAN EVPN Multi-Site using CloudSec:

```
switch# show tunnel-encryption info global
Global Policy Mode: Must-Secure
 SCI list: 0000.0000.0001.0002 0000.0000.0001.0004
 No. of Active Peers        : 1
```

The following example displays all configured CloudSec policies. The output shows the cipher, window size, and SAK retry time for each policy.

```
switch# show tunnel-encryption policy
Tunnel-Encryption Policy      Cipher            Window     SAK Rekey time
----------------------------  ----------------  ---------  --------------
cloudsec                      GCM-AES-XPN-256   134217728  1800
p1                            GCM-AES-XPN-256   1073741823
system-default-tunenc-policy  GCM-AES-XPN-256   268435456
```

The following example displays information about CloudSec sessions. The output shows the peer IP address and policy, the keychain available, and whether the sessions are secure.

```
switch# show tunnel-encryption session
Tunnel-Encryption  Peer Policy  Keychain  RxStatus        TxStatus
-----------------  -----------  --------  -----------     --------------
33.1.33.33         p1           kc1       Secure (AN: 0)  Secure (AN: 2)
33.2.33.33         p1           kc1       Secure (AN: 0)  Secure (AN: 2)
33.3.33.33         p1           kc1       Secure (AN: 0)  Secure (AN: 2)
44.1.44.44         p1           kc1       Secure (AN: 0)  Secure (AN: 0)
44.2.44.44         p1           kc1       Secure (AN: 0)  Secure (AN: 0)
```

The following example displays information about Cloudsec sessions based on PKI Certificate Trustpoint.

```
switch# sh tunnel-encryption session
Tunnel-Encryption Peer   Policy                                   Keychain
RxStatus         TxStatus
------------------------ ----------------------------------------
---------------------------------------- ----------------- -----------------
20.20.20.2               p1                                      PKI: myCA (RSA)
Secure (AN: 0)    Secure (AN: 0)
32.11.11.4               p1                                      PKI: myCA (RSA)
Secure (AN: 0)    Secure (AN: 0)
```

The following example shows the tunnel encryption information for BGP routes:

```
switch# show bgp ipv4 unicast 199.199.199.199  Source-loopback configured on peer BGW for
 CloudSec
BGP routing table information for VRF default, address family IPv4 Unicast
BGP routing table entry for 199.199.199.199/32, version 109
Paths: (1 available, best #1)
Flags: (0x8008001a) (high32 0x000200) on xmit-list, is in urib, is best urib route, is in
HW
Multipath: eBGP

  Advertised path-id 1
  Path type: external, path is valid, is best path, no labeled nexthop, in rib
  AS-Path: 1000 200 , path sourced external to AS
    89.89.89.89 (metric 0) from 89.89.89.89 (89.89.89.89)
```

```
      Origin IGP, MED not set, localpref 100, weight 0
      Tunnel Encapsulation attribute: Length 120

  Path-id 1 advertised to peers:
    2.2.2.2
```

The following example shows if the MAC is attached with the virtual ESI:

```
switch(config)# show bgp l2vpn evpn 0012.0100.000a
BGP routing table information for VRF default, address family L2VPN EVPN
Route Distinguisher: 110.110.110.110:32876
BGP routing table entry for [2]:[0]:[0]:[48]:[0012.0100.000a]:[0]:[0.0.0.0]/216, version
13198
Paths: (1 available, best #1)
Flags: (0x000202) (high32 00000000) on xmit-list, is not in l2rib/evpn, is not in HW
Multipath: eBGP

  Advertised path-id 1
  Path type: external, path is valid, is best path, no labeled nexthop
             Imported to 1 destination(s)
             Imported paths list: l2-10109
  AS-Path: 1000 200 , path sourced external to AS
    10.10.10.10 (metric 0) from 89.89.89.89 (89.89.89.89)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 10109
      Extcommunity: RT:100:10109 ENCAP:8
      ESI: 0300.0000.0000.0200.0309

  Path-id 1 not advertised to any peer

Route Distinguisher: 199.199.199.199:32876
BGP routing table entry for [2]:[0]:[0]:[48]:[0012.0100.000a]:[0]:[0.0.0.0]/216, version
24823
Paths: (1 available, best #1)
Flags: (0x000202) (high32 00000000) on xmit-list, is not in l2rib/evpn, is not in HW
Multipath: eBGP

  Advertised path-id 1
  Path type: external, path is valid, is best path, no labeled nexthop
             Imported to 1 destination(s)
             Imported paths list: l2-10109
  AS-Path: 1000 200 , path sourced external to AS
    9.9.9.9 (metric 0) from 89.89.89.89 (89.89.89.89)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 10109
      Extcommunity: RT:100:10109 ENCAP:8
      ESI: 0300.0000.0000.0200.0309

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

The following example shows the ECMP created for EVPN type-5 routes received from the remote site:

```
switch(config)# show ip route 205.205.205.9 vrf vrf903
IP Route Table for VRF "vrf903"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

205.205.205.9/32, ubest/mbest: 2/0
    *via 9.9.9.9%default, [20/0], 11:06:32, bgp-100, external, tag 1000, segid: 900003
tunnelid: 0x9090909 encap: VXLAN
```

```
    *via 10.10.10.10%default, [20/0], 3d05h, bgp-100, external, tag 1000, segid: 900003
tunnelid: 0xa0a0a0a encap: VXLAN
```

The following example shows if ESI-based MAC multipath is configured for MACs received from the remote site:

```
switch(config)# show l2route evpn mac evi 109 mac 0012.0100.000a detail

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, (Orp): Orphan

Topology Mac Address    Prod   Flags  Seq No Next-Hops
-------- -------------- ------ ------ ------ ----------------------
109      0012.0100.000a BGP    SplRcv 0          9.9.9.9 (Label: 10109)
                                             10.10.10.10 (Label: 10109)
         Route Resolution Type: ESI
         Forwarding State: Resolved (PL)
         Resultant PL: 9.9.9.9, 10.10.10.10
         Sent To: L2FM
         ESI : 0300.0000.0000.0200.0309
         Encap: 1
```

The following example shows that VXLAN EVPN Multi-Site with PIP is configured:

```
switch(config)# show nve interface nve1 detail
Interface: nve1, State: Up, encapsulation: VXLAN
 VPC Capability: VPC-VIP-Only [not-notified]
 Local Router MAC: 700f.6a15.c791
 Host Learning Mode: Control-Plane
 Source-Interface: loopback0 (primary: 14.14.14.14, secondary: 0.0.0.0)
 Source Interface State: Up
 Virtual RMAC Advertisement: No
 NVE Flags:
 Interface Handle: 0x49000001
 Source Interface hold-down-time: 180
 Source Interface hold-up-time: 30
 Remaining hold-down time: 0 seconds
 Virtual Router MAC: N/A
 Virtual Router MAC Re-origination: 0200.2e2e.2e2e
 Interface state: nve-intf-add-complete
 Multisite delay-restore time: 180 seconds
 Multisite delay-restore time left: 0 seconds
 Multisite dci-advertise-pip configured: True
 Multisite bgw-if: loopback1 (ip: 46.46.46.46, admin: Up, oper: Up)
 Multisite bgw-if oper down reason:
```

The following example shows the key text in the running configuration. If you enter the **key-chain tunnelencrypt-psk no-show** command, the key text is hidden.

```
switch# show running-config rpm
!Command: show running-config rpm
!Running configuration last done at: Mon Jun 15 14:41:40 2020
!Time: Mon Jun 15 15:10:27 2020

version 9.3(5) Bios:version 05.40
key chain inter tunnel-encryption
  key 3301
    key-octet-string 7 075f79696a58405441412e2a577f0f077d6461003652302552040a0b76015a504e370c
7972700604755f0e22230c03254323277d2f5359741a6b5d3a5744315f2f cryptographic-algorithm
AES_256_CMAC
```

```
key chain kc1 tunnel-encryption
  key 3537
    key-octet-string 7
072c746f172c3d274e33592e22727e7409106d0037253257580378007775562l3d4e0c7c00770576772
d08515e0804553124577f5a522e046d6a5f485c35425f59 cryptographic-algorithm AES_256_CMAC
    send-lifetime local 09:09:40 Apr 15 2020 duration 1800
  key 2001
    key-octet-string 7
075f79696a58405441412e2a577f0f077d6461003652302552040a0b76015a504e370c7972700604755
f0e22230c03254323277d2f5359741a6b5d3a5744315f2f cryptographic-algorithm AES_256_CMAC
  key 2065
    key-octet-string 7
0729791f6f5e3d213347292d517308730c156c7737223554270f787c07722a513e450a0a0703070c062
e0256210d0e204120510d29222a051f1e594c2135375359 cryptographic-algorithm AES_256_CMAC
  key 2129
    key-octet-string 7
075c796f6f2a4c2642302f5c56790e767063657a4b564f2156777c0a020228564a32780e0472007005530
c5e560f04204056577f2a222d056d1f5c4c533241525d cryptographic-algorithm AES_256_CMAC
  key 2193
    key-octet-string 7
07577014195b402336345a5f260f797d7d6264044b50415755047a7976755a574d350b7e720a0202715d7
a50530d715346205d0c2d525c001f6b5b385046365a29 cryptographic-algorithm AES_256_CMAC

switch# configure terminal
switch(config)# key-chain tunnelencrypt-psk no-show
switch(config)# show running-config rpm

!Command: show running-config rpm
!Running configuration last done at: Mon Jun 15 15:10:44 2020
!Time: Mon Jun 15 15:10:47 2020

version 9.3(5) Bios:version 05.40
key-chain tunnelencrypt-psk no-show
key chain inter tunnel-encryption
  key 3301
    key-octet-string 7 ****** cryptographic-algorithm AES_256_CMAC
key chain kc1 tunnel-encryption
  key 3537
    key-octet-string 7 ****** cryptographic-algorithm AES_256_CMAC
    send-lifetime local 09:09:40 Apr 15 2020 duration 1800
  key 2001
    key-octet-string 7 ****** cryptographic-algorithm AES_256_CMAC
  key 2065
    key-octet-string 7 ****** cryptographic-algorithm AES_256_CMAC
  key 2129
    key-octet-string 7 ****** cryptographic-algorithm AES_256_CMAC
  key 2193
    key-octet-string 7 ****** cryptographic-algorithm AES_256_CMAC
```

The following example shows the trustpoint and keypair configuration.

```
switch# show running-config cert-enroll
!Command: show running-config cert-enroll
!Running configuration last done at: Fri Apr 21 10:53:30 2023
!Time: Fri Apr 21 12:07:31 2023

version 10.3(3) Bios:version 05.47
crypto key generate rsa label myRSA exportable modulus 1024
crypto key generate rsa label myKey exportable modulus 1024
crypto key generate rsa label tmpCA exportable modulus 2048
crypto key generate ecc label src15_ECC_key exportable modulus 224
crypto ca trustpoint src15_ECC_CA
     ecckeypair switch_ECC_key and so on
     revocation-check  crl
crypto ca trustpoint myRSA
```

```
     rsakeypair myRSA
     revocation-check  crl
crypto ca trustpoint tmpCA
     rsakeypair tmpCA
     revocation-check  crl
crypto ca trustpoint myCA
     rsakeypair myKey
     revocation-check  crl
```

The following example shows the certificate contents under a trustpoint.

```
switch(config)# show crypto ca certificates myCA
Trustpoint: myCA
certificate:
subject=CN = switch, serialNumber = FBO22411ABC
issuer=C = US, ST = CA, L = San Jose, O = Org, OU = EN, CN = PKI, emailAddress = abc@xyz.com
serial=2F24FCE6823FCBE5A8AC72C82D0E8E24EB327B0C
notBefore=Apr 19 19:43:48 2023 GMT
notAfter=Aug 31 19:43:48 2024 GMT
SHA1 Fingerprint=D0:F8:1E:32:6E:6D:44:21:6B:AE:92:69:69:AD:88:73:69:76:B9:18
purposes: sslserver sslclient

CA certificate 0:
subject=C = US, ST = CA, L = San Jose, O = Org, OU = EN, CN = PKI, emailAddress = abc@xyz.com
issuer=C = US, ST = CA, L = San Jose, O = Cisco, OU = EN, CN = PKI, emailAddress = ca@ca.com
serial=1142A22DDDE63A047DE0829413359362042CCC31
notBefore=Jul 12 13:25:59 2022 GMT
notAfter=Jul 12 13:25:59 2023 GMT
SHA1 Fingerprint=33:37:C6:D5:F1:B3:E1:79:D9:5A:71:30:FD:50:E4:28:7D:E1:2D:A3
purposes: sslserver sslclient
```

Displaying Statistics for Secure VXLAN EVPN Multi-Site Using CloudSec

You can display or clear Secure VXLAN EVPN Multi-Site using CloudSec statistics using the following commands:

Command	Purpose
show tunnel-encryption statistics [**peer-ip** *peer-ip-address*]	Displays statistics for Secure VXLAN EVPN Multi-Site using CloudSec.
clear tunnel-encryption statistics [**peer-ip** *peer-ip-address*]	Clears statistics for Secure VXLAN EVPN Multi-Site using CloudSec.

The following example shows sample statistics for Secure VXLAN EVPN Multi-Site using CloudSec:

```
switch# show tunnel-encryption statistics
Peer 16.16.16.16 SecY Statistics:

SAK Rx Statistics for AN [0]:
Unchecked Pkts: 0
Delayed Pkts: 0
Late Pkts: 0
OK Pkts: 8170598
Invalid Pkts: 0
Not Valid Pkts: 0
Not-Using-SA Pkts: 0
```

```
Unused-SA Pkts: 0
Decrypted In-Pkts: 8170598
Decrypted In-Octets: 4137958460 bytes
Validated In-Octets: 0 bytes

SAK Rx Statistics for AN [3]:
Unchecked Pkts: 0
Delayed Pkts: 0
Late Pkts: 0
OK Pkts: 0
Invalid Pkts: 0
Not Valid Pkts: 0
Not-Using-SA Pkts: 0
Unused-SA Pkts: 0
Decrypted In-Pkts: 0
Decrypted In-Octets: 0 bytes
Validated In-Octets: 0 bytes

SAK Tx Statistics for AN [0]:
Encrypted Protected Pkts: 30868929
Too Long Pkts: 0
Untagged Pkts: 0
Encrypted Protected Out-Octets: 15758962530 bytes
```

Note In tunnel encryption statistics, if you observe a traffic drop coinciding with an increase in late packets, it could be due to any of the following reasons:

- The packets are being discarded because they are received outside the replay window.
- The tunnel encryption peers are out of sync.
- There is a valid security risk.

In these situations, you should reset the peer session by removing and then reconfiguring the tunnel-encryption peer on the corresponding remote peer, in order to synchronize them again.

Configuration Examples for Secure VXLAN EVPN Multi-Site Using CloudSec

The following example shows how to configure Secure VXLAN EVPN Multi-Site using keychain:

```
key chain kc1 tunnel-encryption
key 2006
key-octet-string 7 075f79696a58405441412e2a577f0f077d6461003652302552040
a0b76015a504e370c7972700604755f0e22230c03254323277d2f5359741a6b5d3a5744315f2f
cryptographic-algorithm AES_256_CMAC

feature tunnel-encryption
tunnel-encryption source-interface loopback4
tunnel-encryption must-secure-policy

tunnel-encryption policy p1
  window-size 1073741823

tunnel-encryption peer-ip 11.1.11.11
```

```
  keychain kc1 policy p1
tunnel-encryption peer-ip 11.2.11.11
  keychain kc1 policy p1
tunnel-encryption peer-ip 44.1.44.44
  keychain kc1 policy p1
tunnel-encryption peer-ip 44.2.44.44
  keychain kc1 policy p1

interface Ethernet1/1
  tunnel-encryption

interface Ethernet1/7
  tunnel-encryption

interface Ethernet1/55
  tunnel-encryption

interface Ethernet1/59
  tunnel-encryption

evpn multisite border-gateway 111
dci-advertise-pip

router bgp 1000
router-id 12.12.12.12
no rd dual
address-family ipv4 unicast
  maximum-paths 10
address-family l2vpn evpn
  maximum-paths 10
vrf vxlan-900101
address-family ipv4 unicast
  maximum-paths 10
address-family ipv6 unicast
  maximum-paths 10

show tunnel-encryption session
Tunnel-Encryption Peer  Policy  Keychain  RxStatus        TxStatus
----------------------  ------  --------  --------------  --------------
11.1.11.11               p1     kc1       Secure (AN: 0)  Secure (AN: 2)
11.2.11.11               p1     kc1       Secure (AN: 0)  Secure (AN: 2)
44.1.44.44               p1     kc1       Secure (AN: 0)  Secure (AN: 2)
44.2.44.44               p1     kc1       Secure (AN: 0)  Secure (AN: 2)
```

Migrating from Multi-Site with VIP to Multi-Site with PIP

Follow these steps for a smooth migration from Multi-Site with VIP to Multi-Site with PIP. The migration needs to be done one site at a time. You can expect minimal traffic loss during the migration.

1. Upgrade all BGWs on all sites to Cisco NX-OS Release 9.3(5) or a later release.
2. Configure BGP maximum paths on all BGWs. Doing so is required for ESI-based MAC multipath and BGP to download all of the next-hops for EVPN Type-2 and Type-5 routes.
3. Pick one site at a time for the migration.
4. Shut down the same-site BGWs except for one BGW. You can use the NVE **shutdown** command to shut down the BGWs.

5. To avoid traffic loss, wait a few minutes before enabling Multi-Site with PIP on the active BGW. Doing so allows the same-site shutdown BGWs to withdraw EVPN routes so remote BGWs send traffic to only the active BGW.
6. Enable Multi-Site with PIP on the active BGW by configuring the **dci-advertise-pip** command.

   The Multi-Site with PIP-enabled BGW advertises the EVPN EAD-per-ES route for the virtual ESI.

   The Multi-Site with PIP-enabled BGW advertises EVPN Type-2 and Type-5 routes with virtual ESI, next-hop as the PIP address, and PIP interface MAC as the RMAC (if applicable) toward DCI. There is no change with respect to advertising EVPN Type-2 and Type-5 routes toward the fabric.

   The remote BGW performs ESI-based MAC multipathing as MAC routes are received with ESI.
7. Unshut the same-site BGWs one at a time and enable Multi-Site with PIP by entering the **dci-advertise-pip** command.

   The remote BGW performs ESI-based MAC multipathing for MAC routes as ESI is the same from all same-site BGWs.

   On the remote BGW, BGP selects paths as multipath and downloads all next-hops for EVPN Type-5 routes.

Enhanced Convergence for vPC BGW CloudSec Deployments

Traditionally, single loopback interface is configured as NVE source interface, where both PIP and VIP of vPC complex are configured. Beginning with Cisco NX-OS Release 10.3(2)F, you can configure a separate loop back for CloudSec enabled vPC BGW. It is recommended to use separate loopback interfaces for source and anycast IP addresses under NVE for better convergence in vPC deployments. The IP address configured on the source-interface is the PIP of the vPC node, and the IP address configured on the anycast interface is the VIP of that vPC complex. Note that the secondary IP configured on the NVE source-interface will have no effect if the NVE anycast interface is also configured.

With separate loopbacks, the convergence for dual-attached EVPN Type-2 and Type-5 routes traffic destined for DCI side will be improved.

Migration to Anycast Interface

If a user wants to specify an anycast interface, the user needs to unconfigure the existing source-interface and reconfigure with both source and anycast interfaces. This will lead to temporary traffic loss. For all green field deployments, it is recommended to configure both the source and anycast interface to avoid the convergence problem specified.

NVE Interface Configuration with Enhanced Convergence for vPC BGW CloudSec Deployments

The user needs to specify anycast interface along with NVE source-interface on vPC BGW. In today’s VxLANv6 deployments, the provision to specify both source-interface and anycast interface is already present. In order to improve vPC convergence for VxLANv4, the anycast option is mandatory.

Configuration Example:

```
interface nve <number>
        source-interface <interface> [anycast <anycast-intf>]
```

iBGP Session Requirement

Underlay IPv4/IPv6 unicast iBGP session must be configured between vPC BGW peer nodes. This is to accommodate key propagation during the DCI isolation on any vPC BGW.

CHAPTER 20

Configuring VXLAN QoS

This chapter contains the following sections:

Information About VXLAN QoS

VXLAN QoS enables you to provide Quality of Service (QoS) capabilities to traffic that is tunneled in VXLAN.

Traffic in the VXLAN overlay can be assigned to different QoS properties:

- Classification traffic to assign different properties.
- Including traffic marking with different priorities.
- Queuing traffic to enable priority for the protected traffic.
- Policing for misbehaving traffic.
- Shaping for traffic that limits speed per interface.
- Properties traffic sensitive to traffic drops.

Note QoS allows you to classify the network traffic, police and prioritize the traffic flow, and provide congestion avoidance. For more information about QoS, see the Cisco Nexus 9000 Series NX-OS Quality of Service Configuration Guide, Release 9.2(x).

This section contains the following topics:

VXLAN QoS Terminology

This section defines VXLAN QoS terminology.

Table 8: VXLAN QoS Terminology

Term	Definition
Frames	Carries traffic at Layer 2. Layer 2 frames carry Layer 3 packets.
Packets	Carries traffic at Layer 3.
VXLAN packet	Carries original frame, encapsulated in VXLAN IP/UDP header.
Original frame	A Layer 2 or Layer 2 frame that carries the Layer 3 packet before encapsulation in a VXLAN header.
Decapsulated frame	A Layer 2 or a Layer 2 frame that carries a Layer 3 packet after the VXLAN header is decapsulated.
Ingress VTEP	The point where traffic is encapsulated in the VXLAN header and enters the VXLAN tunnel.
Egress VTEP	The point where traffic is decapsulated from the VXLAN header and exits the VXLAN tunnel.
Class of Service (CoS)	Refers to the three bits in an 802.1Q header that are used to indicate the priority of the Ethernet frame as it passes through a switched network. The CoS bits in the 802.1Q header are commonly referred to as the 802.1p bits. 802.1Q is discarded prior to frame encapsulation in a VXLAN header, where CoS value is not present in VXLAN tunnel. To maintain QoS when a packet enters the VXLAN tunnel, the type of service (ToS) and CoS values map to each other.
IP precedence	The 3 most significant bits of the ToS byte in the IP header.
Differentiated Services Code Point (DSCP)	The first six bits of the ToS byte in the IP header. DSCP is only present in an IP packet.
Explicit Congestion Notification (ECN)	The last two bits of the ToS byte in the IP header. ECN is only present in an IP packet.
QoS tags	Prioritization values carried in Layer 3 packets and Layer 2 frames. A Layer 2 CoS label can have a value ranging between zero for low priority and seven for high priority. A Layer 3 IP precedence label can have a value ranging between zero for low priority and seven for high priority. IP precedence values are defined by the three most significant bits of the 1-byte ToS byte. A Layer 3 DSCP label can have a value between 0 and 63. DSCP values are defined by the six most significant bits of the 1-byte IP ToS field.

Term	Definition
Classification	The process used for selecting traffic for QoS
Marking	The process of setting: a Layer 2 COS value in a frame, Layer 3 DSCP value in a packet, and Layer 3 ECN value in a packet. Marking is also the process of choosing different values for the CoS, DSCP, ECN field to mark packets so that they have the priority that they require during periods of congestion.
Policing	Limiting bandwidth used by a flow of traffic. Policing can mark or drop traffic.
MQC	The Cisco Modular QoS command line interface (MQC) framework, which is a modular and highly extensible framework for deploying QoS.

VXLAN QoS Features

The following topics describe the VXLAN QoS features that are supported in a VXLAN network:

Trust Boundaries

The trust boundary forms a perimeter on your network. Your network trusts (and does not override) the markings on your switch. The existing ToS values are trusted when received on in the VXLAN fabric.

Classification

You use classification to partition traffic into classes. You classify the traffic based on the port characteristics or the packet header fields that include IP precedence, differentiated services code point (DSCP), Layer 3 to Layer 4 parameters, and the packet length.

The values used to classify traffic are called match criteria. When you define a traffic class, you can specify multiple match criteria, you can choose to not match on a particular criterion, or you can determine the traffic class by matching any or all criteria.

Traffic that fails to match any class is assigned to a default class of traffic called class-default.

Marking

Marking is the setting of QoS information that is related to a packet. Packet marking allows you to partition your network into multiple priority levels or classes of service. You can set the value of a standard QoS field for COS, IP precedence, and DSCP. You can also set the QoS field for internal labels (such as QoS groups) that can be used in subsequent actions. Marking QoS groups is used to identify the traffic type for queuing and scheduling traffic.

Policing

Policing causes traffic that exceeds the configured rate to be discarded or marked down to a higher drop precedence.

Single-rate policers monitor the specified committed information rate (CIR) of traffic. Dual-rate policers monitor both CIR and peak information rate (PIR) of traffic.

Queuing and Scheduling

The queuing and scheduling process allows you to control the queue usage and the bandwidth that is allocated to traffic classes. You can then achieve the desired trade-off between throughput and latency.

You can limit the size of the queues for a particular class of traffic by applying either static or dynamic limits.

You can apply weighted random early detection (WRED) to a class of traffic, which allows packets to be dropped based on the QoS group. The WRED algorithm allows you to perform proactive queue management to avoid traffic congestion.

ECN can be enabled along with WRED on a particular class of traffic to mark the congestion state instead of dropping the packets. ECN marking in the VXLAN tunnel is performed in the outer header, and at the Egress VTEP is copied to decapsulated frame.

Traffic Shaping

You can shape traffic by imposing a maximum data rate on a class of traffic so that excess packets are retained in a queue to smooth (constrain) the output rate. In addition, minimum bandwidth shaping can be configured to provide a minimum guaranteed bandwidth for a class of traffic.

Traffic shaping regulates and smooths out the packet flow by imposing a maximum traffic rate for each port's egress queue. Packets that exceed the threshold are placed in the queue and are transmitted later. Traffic shaping is similar to Traffic Policing, but the packets are not dropped. Because packets are buffered, traffic shaping minimizes packet loss (based on the queue length), which provides better traffic behavior for TCP traffic.

By using traffic shaping, you can control the following:

- Access to available bandwidth.
- Ensure that traffic conforms to the policies established for it.
- Regulate the flow of traffic to avoid congestion that can occur when the egress traffic exceeds the access speed of its remote, target interface.

For example, you can control access to the bandwidth when the policy dictates that the rate of a given interface must not, on average, exceed a certain rate. Despite the access rate exceeding the speed.

Network QoS

The network QoS policy defines the characteristics of each CoS value, which are applicable network wide across switches. With a network QoS policy, you can configure the following:

- Pause behavior—You can decide whether a CoS requires the lossless behavior which is provided by using a priority flow control (PFC) mechanism that prevents packet loss during congestion) or not. You can configure drop (frames with this CoS value can be dropped) and no drop (frames with this CoS value cannot be dropped). For the drop and no drop configuration, you must also enable PFC per port. For more information about PFC, see "Configuring Priority Flow Control".

Pause behavior can be achieved in the VXLAN tunnel for a specific queue-group.

VXLAN Priority Tunneling

In the VXLAN tunnel, DSCP values in the outer header are used to provide QoS transparency in end-to-end of the tunnel. The outer header DSCP value is derived from the DSCP value with Layer 3 packets or the CoS value for Layer 2 frames. At the VXLAN tunnel egress point, the priority of the decapsulated traffic is chosen based on the mode. For more information, see Decapsulated Packet Priority Selection, on page 380.

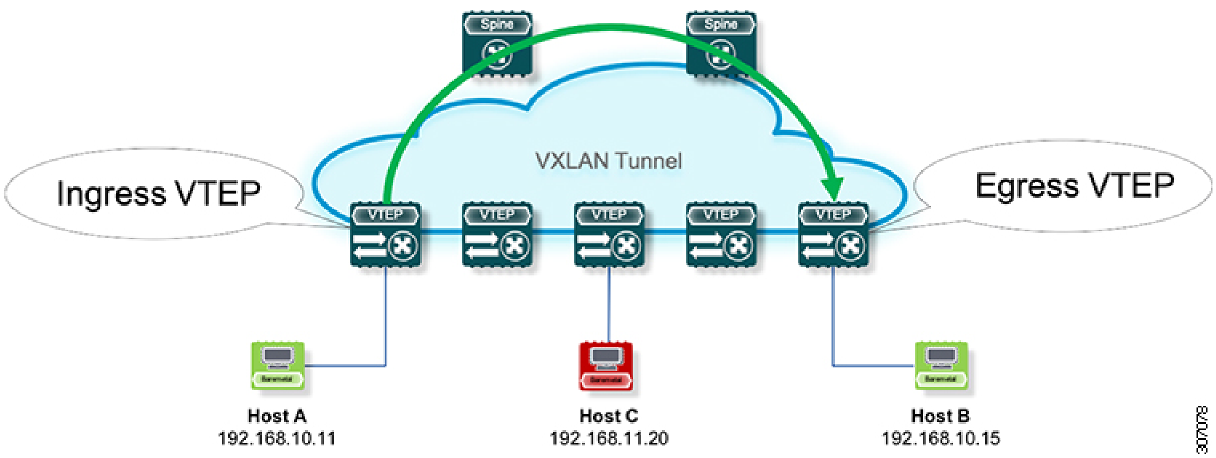
MQC CLI

All available QoS features for VXLAN QoS are managed from the modular QoS command-line interface (CLI). The Modular QoS CLI (MQC) allows you to define traffic classes (class maps), create and configure traffic policies (policy maps), and perform actions that are defined in the policy maps to interface (service policy).

VXLAN QoS Topology and Roles

This section describes the roles of network devices in implementing VXLAN QoS.

Figure 33: VXLAN Network



The network is bidirectional, but in the previous image, traffic is moving left to right.

In the VXLAN network, points of interest are ingress VTEPs where the original traffic is encapsulated in a VXLAN header. Spines are transporting hops that connect ingress and egress VTEPs. An egress VTEP is the point where VXLAN encapsulated traffic is decapsulated and egresses the VTEP as classical Ethernet traffic.

Note Ingress and egress VTEPs are the boundary between the VXLAN tunnel and the IP network.

This section contains the following topics:

Ingress VTEP and Encapsulation in the VXLAN Tunnel

At the ingress VTEP, the VTEP processes packets as follows:

Step 1 Layer 2 or Layer 3 traffic enters the edge of the VXLAN network.

Step 2 The switch receives the traffic from the input interface and uses the 802.1p bits or the DSCP value to perform any classification, marking, and policing. It also derives the outer DSCP value in the VXLAN header. For classification of incoming IP packets, the input service policy can also use access control lists (ACLs).

Step 3 For each incoming packet, the switch performs a lookup of the IP address to determine the next hop.

Step 4 The packet is encapsulated in the VXLAN header. The encapsulated packet's VXLAN header is assigned a DSCP value that is based on QoS rules.

Step 5 The switch forwards the encapsulated packets to the appropriate output interface for processing.

Step 6 The encapsulated packets, marked by the DSCP value, are sent to the VXLAN tunnel output interface.

Transport Through the VXLAN Tunnel

In the transport through a VXLAN tunnel, the switch processes the VXLAN packets as follows:

Step 1 The VXLAN encapsulated packets are received on an input interface of a transport switch. The switch uses the outer header to perform classification, marking, and policing.

Step 2 The switch performs a lookup on the IP address in the outer header to determine the next hop.

Step 3 The switch forwards the encapsulated packets to the appropriate output interface for processing.

Step 4 VXLAN sends encapsulated packets through the output interface.

Egress VTEP and Decapsulation of the VXLAN Tunnel

At the egress VTEP boundary of the VXLAN tunnel, the VTEP processes packets as follows:

Step 1 Packets encapsulated in VXLAN are received at the NVE interface of an egress VTEP, where the switch uses the inner header DSCP value to perform classification, marking, and policing.

Step 2 The switch removes the VXLAN header from the packet, and does a lookup that is based on the decapsulated packet's headers.

Step 3 The switch forwards the decapsulated packets to the appropriate output interface for processing.

Step 4 Before the packet is sent out, a DSCP value is assigned to a Layer 3 packet based on the decapsulation priority or based on marking Layer 2 frames.

Step 5 The decapsulated packets are sent through the outgoing interface to the IP network.

Classification at the Ingress VTEP, Spine, and Egress VTEP

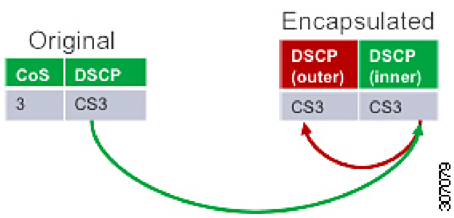
This section includes the following topics:

IP to VXLAN

At the ingress VTEP, the ingress point of the VXLAN tunnel, traffic is encapsulated in the VXLAN header. Traffic on an ingress VTEP is classified based on the priority in the original header. Classification can be performed by matching the CoS, DSCP, and IP precedence values or by matching traffic with the ACL based on the original frame data.

When traffic is encapsulated in the VXLAN, the Layer 3 packet's DSCP value is copied from the original header to the outer header of the VXLAN encapsulated packet. This behavior is illustrated in the following figure:

Figure 34: Copy of Priority from Layer-3 Packet to VXLAN Outer Header



For Layer 2 frames without the IP header, the DSCP value of the outer header is derived from the CoS-to-DSCP mapping present in the hardware illustrated in Default Settings for VXLAN QoS, on page 383. In this way, the original QoS attributes are preserved in the VXLAN tunnel. This behavior is illustrated in the following figure:

Figure 35: Copy of Priority from Layer-2 Frame to VXLAN Outer Header

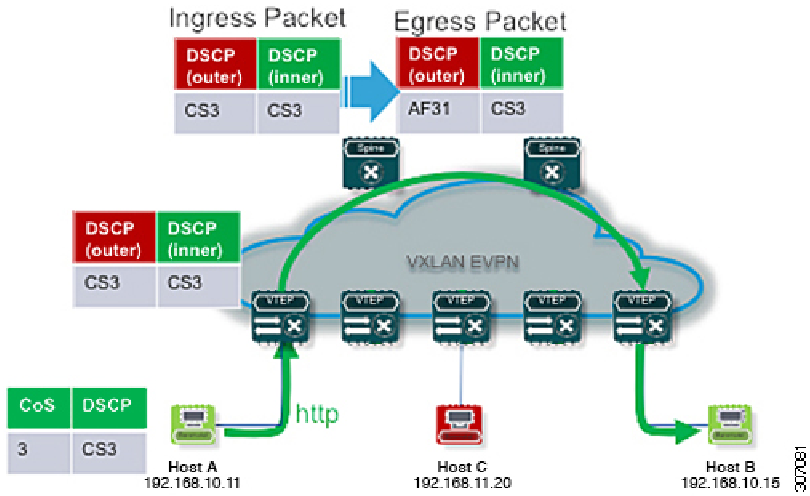
A Layer 2 frame, does not have a DSCP value present because the IP header is not present in the frame. After a Layer 2 frame is encapsulated, the original CoS value is not preserved in the VXLAN tunnel.

Inside the VXLAN Tunnel

Inside the VXLAN tunnel, traffic classification is based on the outer header DSCP value. Classification can be done matching the DCSP value or using ACLs for classification.

If VXLAN encapsulated traffic is crossing the trust boundary, marking can be changed in the packet to match QoS behavior in the tunnel. Marking can be performed inside of the VXLAN tunnel, where a new DSCP value is applied only on the outer header. The new DSCP value can influence different QoS behaviors inside the VXLAN tunnel. The original DSCP value is preserved in the inner header.

Figure 36: Marking Inside of the VXLAN Tunnel



VXLAN to IP

Classification at the egress VTEP is performed for traffic leaving the VXLAN tunnel. For classification at the egress VTEP, the inner header values are used. The inner DSCP value is used for priority-based classification. Classification can be performed using ACLs.

Classification is performed on the NVE interface for all VXLAN tunneled traffic.

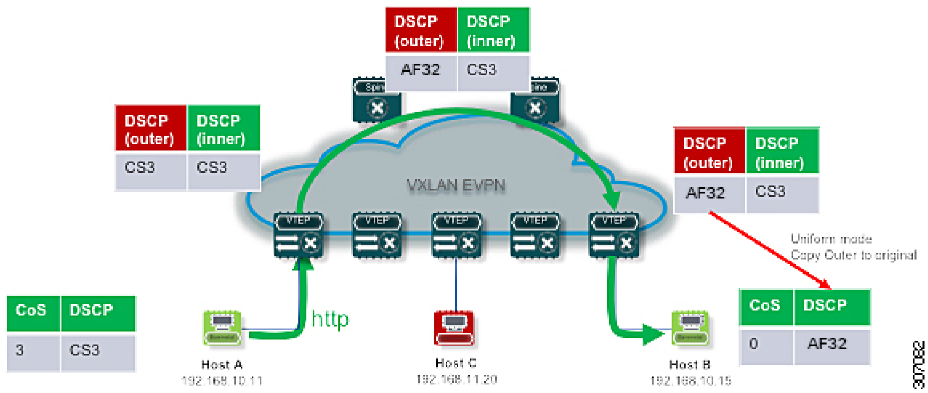
Marking and policing can be performed on the NVE interface for tunneled traffic. If marking is configured, newly marked values are present in the decapsulated packet. Because the original CoS value is not preserved in the encapsulated packet, marking can be performed for decapsulated packets for any devices that expect an 802.1p field for QoS in the rest of the network.

Decapsulated Packet Priority Selection

At the egress VTEP, the VXLAN header is removed from the packet and the decapsulated packet egresses the switch with the DSCP value. The switch assigns the DSCP value of the decapsulated packet based on two modes:

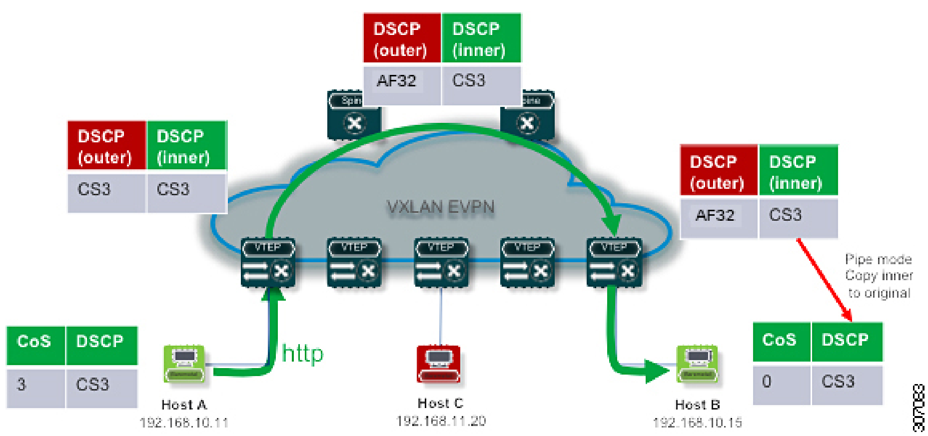
- Uniform mode – the DSCP value from the outer header of the VXLAN packet is copied to the decapsulated packet. Any change of the DSCP value in the VXLAN tunnel is preserved and present in the decapsulated packet. Uniform mode is the default mode of decapsulated packet priority selection.

Figure 37: Uniform Mode Outer DSCP Value is Copied to Decapsulated Packet DSCP Value for a Layer-3 Packet



- Pipe mode – the original DSCP value is preserved at the VXLAN tunnel end. At the egress VTEP, the system copies the inner DSCP value to the decapsulated packet DSCP value. In this way, the original DSCP value is preserved at the end of the VXLAN tunnel.

Figure 38: Pipe Mode Inner DSCP Value is Copied to Decapsulated Packet DSCP Value for Layer-3 Packet



Guidelines and Limitations for VXLAN QoS

Note The QoS policy must be configured end-to-end for this feature to work as designed.

VXLAN QoS has the following configuration guidelines and limitations:

- Cisco Nexus 9364C, 9300-EX, and 9300-FX/FX2/FX3 platform switches and Cisco Nexus 9500 platform switches with -EX/FX or -R/RX line cards support VXLAN QoS.
- Beginning with Cisco NX-OS Release 9.3(3), Cisco Nexus 9300-GX platform switches support VXLAN QoS in default mode.
- The following features are supported on Cisco Nexus 9504 and 9508 platform switches with -R/RX line cards:

  - Physical interface level queuing should work as normal L2/L3 queuing/QoS
  - IPv4 bridged case works in terms of copying inner ToS to outer VXLAN ToS

- The following features are not supported on Cisco Nexus 9504 and 9508 platform switches with -R and -RX line cards:
  - Policies on the NVE interface
  - IPv6 type of service (ToS) from inner to VXLAN outer copying
  - IPv4 routed cases for QoS. ToS from inner is not copied to outer VXLAN header

- For Cisco Nexus 9504 and 9508 platform switches with -RX line cards, the default mode is pipe for VXLAN decapsulation (inner packet DSCP not modified based on outer IP header DSCP value). This is a difference in behavior from other line cards types. If -RX line cards and other line cards are used in the same network, the **qos-mode pipe** command can be used in switches where non-RX line cards are present in order to have the same behavior. For details on the configuration command, see Configuring Type QoS on the Egress VTEP, on page 384.
- VXLAN QoS is supported in the EVPN fabric.
- The original IEEE 802.1Q header is not preserved in the VXLAN tunnel. The CoS value is not present in the inner header of the VXLAN-encapsulated packet.
- Statistics (counters) are present for the NVE interface.
- Egress policing is not supported on outgoing interface (uplink connecting to spine) of the encap (ingress) VXLAN VTEP.
- In a vPC, configure the change of the decapsulated packet priority selection on both peers.
- The service policy on an NVE interface can attach only in the input direction.
- If DSCP marking is present on the NVE interface, traffic to the BUD node preserves marking in the inner and outer headers. If a marking action is configured on the NVE interface, BUM traffic is marked with a new DSCP value on Cisco Nexus 9364C and 9300-EX platform switches.
- A classification policy applied to an NVE interface applies only on VXLAN-encapsulated traffic. For all other traffic, the classification policy must be applied on the incoming interface.
- To mark the decapsulated packet with a CoS value, a marking policy must be attached to the NVE interface to mark the CoS value to packets where the VLAN header is present.
- The following guidelines and limitations apply to VXLAN QoS configuration on the DCI handoff node:
  - Beginning with Cisco NX-OS Release 9.3(5), Cisco Nexus 9300-GX platform switches support VXLAN QoS configuration on the DCI handoff node.
  - VXLAN QoS configuration on the DCI handoff node does not support end-to-end priority flow control (PFC) for Cisco Nexus 9336C-FX2, 93240YC-FX2, and 9300-GX platform switches.
  - Microburst, dynamic packet prioritization (DPP), and approximate fair-drop (AFD) are supported on VXLAN-encapsulated packets.

- The following limitations apply to the VXLAN QoS policies when using a Border Gateway (BGW) Spine:

- If QoS policies are needed for intra-site BUM traffic for VNI with multicast underlay, and that multicast underlay group is also owned by a VNI defined on the BGW Spine, then the QoS policy must be applied to the NVE interface. QoS policies applied to fabric interfaces will not modify these flows since the NVE interface acts as an incoming interface.
- If QoS policies are needed for intra-site BUM traffic for VNI with multicast underlay, and that multicast group is not owned by a VNI defined on the BGW Spine, then the QoS policy must be applied to a fabric interface. QoS policies applied to the NVE interface will not modify these flows since the NVE is not considered an incoming interface.
- If the NVE interface of the BGW Spine owns a multicast group used for BUM traffic within the local fabric, QoS policies cannot be applied to both the fabric interfaces and NVE interface to differentiate treatment of intra-site and inter-site flows for that multicast group.

Default Settings for VXLAN QoS

The following table lists the default CoS-to-DSCP mapping in the ingress VTEP for Layer 2 frames:

Table 9: Default CoS-to-DSCP Mapping

CoS of Original Layer 2 Frame	DSCP of Outer VXLAN Header
0	0
1	8
2	16
3	26
4	32
5	46
6	48
7	56

Configuring VXLAN QoS

Configuration of VXLAN QoS is done using the MQC model. The same configuration that is used for the QoS configuration applies to VXLAN QoS. For more information about configuring QoS, see the Cisco Nexus 9000 Series NX-OS Quality of Service Configuration Guide, Release 9.2(x).

VXLAN QoS introduces a new service-policy attachment point which is NVE – Network Virtual Interface. At the egress VTEP, the NVE interface is the point where traffic is decapsulated. To account for all VXLAN traffic, the service policy must be attached to an NVE interface.

The next section describes the configuration of the classification at the egress VTEP, and **service-policy type qos** attachment to an NVE interface.

Configuring Type QoS on the Egress VTEP

Configuration of VXLAN QoS is done by using the MQC model. The same configuration is used for QoS configuration for VXLAN QoS. For more information about configuring QoS, see the Cisco Nexus 9000 Series NX-OS Quality of Service Configuration Guide, Release 9.2(x).

VXLAN QoS introduces a new service-policy attachment point which is the Network Virtual Interface (NVE). At the egress VTEP, the NVE interface points where traffic is decapsulated. To account for all VXLAN traffic, the service policy must be attached to an NVE interface.

This procedure describes the configuration of classification at the egress VTEP, and **service-policy type qos** attachment to an NVE interface.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**[no] class-map [type [qos]] | [match-all] | [match-any]** *class-map-name* **Example:** `switch(config)# class-map type qos class1`	Creates or accesses the class map *class--map-name* and enters **class-map** mode. The *class--map-name* argument can contain alphabetic, hyphen, or underscore characters, and can be up to 40 characters. (match-any is the default when the **no** option is selected and multiple match statements are entered.)
Step 3	**[no] match [access-group | cos | dscp | precedence]** *{name | 0-7 | 0-63 | 0-7}* **Example:** `switch(config-cmap-qos)# match dscp 26`	Configures the traffic class by matching packets based on access-list, **cos** value, **dscp** values, or IP **precedence** value
Step 4	**[no] policy-map type qos** *policy-map-name* **Example:** `switch(config-cmap-qos)# policy-map type qos policy`	Creates or accesses the policy map that is named *policy-map-name* and then enters policy-map mode. The policy-map name can contain alphabetic, hyphen, or underscore characters, is case sensitive, and can be up to 40 characters.
Step 5	**[no] class** *class-name* **Example:** `switch(config-pmap-qos)# class class1`	Creates a reference to class-name and enters policy-map class configuration mode. The class is added to the end of the policy map unless insert-before is used to specify the class to insert before. Use the class-default keyword to select all traffic that is not currently matched by classes in the policy map.
Step 6	**[no] set qos-group** *qos-group-value* **Example:** `switch(config-pmap-c-qos)# set qos-group 1`	Sets the QoS group value to *qos-group-value*. The value can range from 1 through 126. The **qos-group** is referenced in type queuing and type network-qos as matching criteria.

	Command or Action	Purpose
Step 7	**exit** **Example:** `switch(config-pmap-c-qos)# exit`	Exits class-map mode.
Step 8	**[no] interface nve** *nve-interface-number* **Example:** `switch(config)# interface nve 1`	Enters interface mode to configure the NVE interface.
Step 9	**[no] service-policy type qos input** *policy-map-name* **Example:** `switch(config-if-nve)# service-policy type qos input policy`	Adds a service-policy *policy-map-name* to the interface in the input direction. You can attach only one input policy to an NVE interface.
Step 10	(Optional) **[no] qos-mode [pipe]** **Example:** `switch(config-if-nve)# qos-mode pipe`	Selecting decapsulated packet priority selection and using pipe mode. Entering the **no** form of this command negates pipe mode and defaults to uniform mode.

Verifying the VXLAN QoS Configuration

Table 10: VXLAN QoS Verification Commands

Command	Purpose
show class map	Displays information about all configured class maps.
show policy-map	Displays information about all configured policy maps.
show running ipqos	Displays configured QoS configuration on the switch.

VXLAN QoS Configuration Examples

Ingress VTEP Classification and Marking

This example shows how to configure the **class-map type qos** command for classification matching traffic with an ACL. Enter the **policy-map type qos** command to put traffic in qos-group 1 and set the DSCP value. Enter the **service-policy type qos** command to attach to the ingress interface in the input direction to classify traffic matching the ACL.

```
access-list ACL_QOS_DSCP_CS3 permit ip any any eq 80

class-map type qos CM_QOS_DSCP_CS3
 match access-group name ACL_QOS_DSCP_CS3

policy-map type qos PM_QOS_MARKING
 class CM_QOS_DSCP_CS3
```

```
 set qos-group 1
 set dscp 24

interface ethernet1/1
 service-policy type qos input PM_QOS_MARKING
```

Transit Switch – Spine Classification

This example shows how to configure the **class-map type qos** command for classification matching DSCP 24 set on the ingress VTEP. Enter the **policy-map type qos** command to put traffic in qos-group 1. Enter the **service-policy type qos** command to attach to the ingress interface in the input direction to classify traffic matching criteria.

```
class-map type qos CM_QOS_DSCP_CS3
 match dscp 24

policy-map type qos PM_QOS_CLASS
 class CM_QOS_DSCP_CS3
 set qos-group 1

interface Ethernet 1/1
 service-policy type qos input PM_QOS_CLASS
```

Egress VTEP Classification and Marking

This example shows how to configure the **class-map type qos** command for classification matching traffic by DSCP value. Enter the **policy-map type qos** to place traffic in qos-group 1 and mark CoS value in outgoing frames. The **service-policy type qos** command is applied to the NVE interface in the input direction to classify traffic coming out of the VXLAN tunnel.

```
class-map type qos CM_QOS_DSCP_CS3
 match dscp 24

policy-map type qos PM_QOS_MARKING
 class CM_QOS_DSCP_CS3
 set qos-group 1
 set cos 3

interface nve 1
 service-policy type qos input PM_QOS_MARKING
```

Queuing

This example shows how to configure the **policy-map type queueing** command for traffic in qos-group 1. Assigning 50% of the available bandwidth to q1 mapped to qos-group 1 and attaching policy in the output direction to all ports using the **system qos** command.

```
policy-map type queuing PM_QUEUING
class type queuing c-out-8q-q7
      priority level 1
    class type queuing c-out-8q-q6
      bandwidth remaining percent 0
    class type queuing c-out-8q-q5
      bandwidth remaining percent 0
    class type queuing c-out-8q-q4
      bandwidth remaining percent 0
    class type queuing c-out-8q-q3
      bandwidth remaining percent 0
```

```
    class type queuing c-out-8q-q2
      bandwidth remaining percent 0
    class type queuing c-out-8q-q1
      bandwidth remaining percent 50
    class type queuing c-out-8q-q-default
      bandwidth remaining percent 50

system qos
 service-policy type queueing output PM_QUEUING
```

CHAPTER 21

Configuring vPC Fabric Peering

This chapter contains the following sections:

Information About vPC Fabric Peering

vPC Fabric Peering provides an enhanced dual-homing access solution without the overhead of wasting physical ports for vPC Peer Link. This feature preserves all the characteristics of a traditional vPC.

The following lists the vPC Fabric Peering solution:

- vPC Fabric Peering port-channel with virtual members (tunnels).
- vPC Fabric Peering (tunnel) with removal of the physical peer link requirement.
- vPC Fabric Peering up/down events are triggered based on route updates and fabric up/down.
- Uplink tracking for extended failure coverage.
- vPC Fabric Peering reachability via the routed network, such as the spine.
- Increased resiliency of the vPC control plane over TCP-IP (CFSoIP).
- Data plane traffic over the VXLAN tunnel.
- Communication between vPC member switches uses VXLAN encapsulation.
- Failure of all uplinks on a node result in vPC ports going down on that switch. In that scenario, vPC peer takes up the primary role and forwards the traffic.
- Uplink tracking with state dependency and up/down signalization for vPCs.
- Positive uplink state tracking drives vPC primary role election.
- For border leafs and spines, there is no need for per-VRF peering since network communication uses the fabric.
- Enhance forwarding to orphans hosts by extending the VIP/PIP feature to Type-2 routes.

- Infra-VLAN is not required for vPC fabric peering.

> **Note** The vPC Fabric Peering counts as three VTEPs unlike a normal vPC which counts as one VTEP.

Guidelines and Limitations for vPC Fabric Peering

The following are the vPC Fabric Peering guidelines and limitations:

- Cisco Nexus 9332C, 9364C, and 9300-EX/FX/FXP/FX2/FX3/GX/GX2 platform switches support vPC Fabric Peering. Cisco Nexus 9200 and 9500 platform switches do not support vPC Fabric Peering.

> **Note** For Cisco Nexus 9300-EX switches, mixed-mode multicast and ingress replication are not supported. VNIs must be configured with either multicast or IR underlay, but not both.

- vPC Fabric Peering requires TCAM carving of the region **ing-flow-redirect**. TCAM carving requires saving the configuration and reloading the switch prior to using the feature.

> **Note** This requirement applies only to Cisco Nexus 9300-EX, 9300-FX, 9300-FX2, and 9364C platform switches.

- Prior to reconfiguring the vPC Fabric Peering source and destination IP, the vPC domain must be shut down. Once the vPC Fabric Peering source and destination IP have been adjusted, the vPC domain can be enabled (**no shutdown**).
- The source and destination IP supported in **virtual peer-link destination** command are class A, B, and C. Class D and E are not supported for vPC Fabric Peering.
- The vPC Fabric Peering peer-link is established over the transport network (the spine layer of the fabric). As communication between vPC peers occurs in this manner, control plane information CFS messages used to synchronize port state information, VLAN information, VLAN-to-VNI mapping, host MAC addresses are transmitted over the fabric. CFS messages are marked with the appropriate DSCP value, which should be protected in the transport network. The following example shows a sample QoS configuration on the spine layer of Cisco Nexus 9000 Series switches.

  Classify traffic by matching the DSCP value (DSCP 56 is the default value):

  ```
  class-map type qos match-all CFS
    match dscp 56
  ```

  Set traffic to the qos-group that corresponds with the strict priority queue for the appropriate spine switch. In this example, the switch sends traffic to qos-group 7, which corresponds to the strict priority queue (Queue 7). Note that different Cisco Nexus platforms might have a different queuing structure.

  ```
  policy-map type qos CFS
    class CFS
  ```

```
    Set qos-group 7
```

Assign a classification service policy to all interfaces toward the VTEP (the leaf layer of the network):

```
interface Ethernet 1/1
  service-policy type qos input CFS
```

- Beginning with Cisco NX-OS Release 10.1(1), FEX Support is provided with vMCT for IPv4 underlay on Cisco Nexus 9300-EX/FX/FX2/FX3 platform switches.
- The vPC Fabric Peering domain is not supported in the role of a Multi-Site vPC BGW.
- Enhance forwarding to orphan hosts by extending the VIP/PIP feature to Type-2 routes.
- Layer 3 Tenant Routed Multicast (TRM) is supported. Layer 2/Layer 3 TRM (Mixed Mode) is not supported.
- If Type-5 routes are used with this feature, the **advertise-pip** command is a mandatory configuration.
- VTEPs behind vPC ports are not supported. This means that virtual peer-link peers cannot act as a transit node for the VTEPs behind the vPC ports.
- SVI and sub-interface uplinks are not supported.
- An orphan Type-2 host is advertised using PIP. A vPC Type-2 host is advertised using VIP. This is the default behavior for a Type-2 host.

  To advertise an orphan Type-5 route using PIP, you need to advertise PIP under BGP.
- Traffic from remote VTEP to orphan hosts would land on the actual node which has the orphans. Bouncing of the traffic is avoided.

Note When the vPC leg is down, vPC hosts are still advertised with the VIP IP.

- Immediately after converting from fabric peering to a physical peer link, make the following changes on both peers:
  1. Globally configure a TCAM region using the **hardware access-list tcam region ing-flow-redirect 0** command.
  2. Optionally, allocate the free space to other classes. For more information, see Understand How to Carve Nexus 9000 TCAM Space.
  3. Save the running configuration using the **copy running-config startup-config** command.
  4. Reload the switch.

Configuring vPC Fabric Peering

Ensure the vPC Fabric Peering DSCP value is consistent on both vPC member switches. Ensure that the corresponding QoS policy matches the vPC Fabric Peering DSCP marking.

All VLANs that require communication traversing the vPC Fabric Peering must have a VXLAN enabled (vn-segment); this includes the native VLAN.

Note For MSTP, VLAN 1 must be extended across vPC Fabric Peering if the peer-link and vPC legs have the default native VLAN configuration. This behavior can be achieved by extending VLAN 1 over VXLAN (vn-segment). If the peer-link and vPC legs have non-default native VLANs, those VLANs must be extended across vPC Fabric Peering by associating the VLANs with VXLAN (vn-segment).

Use the **show vpc virtual-peerlink vlan consistency** command for verification of the existing VLAN-to-VXLAN mapping used for vPC Fabric Peering.

peer-keepalive command for vPC Fabric Peering is supported with one of the following configurations:

- Management interface
- Dedicated Layer 3 link in default or non-default VRF
- Loopback interface reachable using the spine.

Configuring Features

Example uses OSPF as the underlay routing protocol.

```
configure terminal
nv overlay evpn
feature ospf
feature bgp
feature pim
feature interface-vlan
feature vn-segment-vlan-based
feature vpc

feature nv overlay
```

vPC Configuration

Note To change the vPC Fabric Peering source or destination IP, the vPC domain must be shutdown prior to modification. The vPC domain can be returned to operation after the modifying by using the **no shutdown** command.

Configuring TCAM Carving

```
hardware access-list tcam region ing-racl 0
hardware access-list tcam region ing-sup 768
hardware access-list tcam region ing-flow-redirect 512
```

Configuring the vPC Domain

```
vpc domain 100
peer-keepalive destination 192.0.2.1
virtual peer-link destination 192.0.2.100 source 192.0.2.20/32 [dscp <dscp-value>]
Warning: Appropriate TCAM carving must be configured for virtual peer-link vPC
peer-switch
peer-gateway
ip arp synchronize
```

```
ipv6 nd synchronize
exit
```

Note The **dscp** keyword in optional. Range is 1 to 63. The default value is 56.

Configuring vPC Fabric Peering Port Channel

No need to configure members for the following port channel.

```
interface port-channel 10
switchport
switchport mode trunk
vpc peer-link
```

```
interface loopback0
```

Note This loopback is not the NVE source-interface loopback (interface used for the VTEP IP address).

```
interface loopback 0
ip address 192.0.2.20/32
ip router ospf 1 area 0.0.0.0
```

Note You can use the loopback for BGP peering or a dedicated loopback. This lookback must be different that the loopback for peer keep alive.

Configuring the Underlay Interfaces

Both L3 physical and L3 port channels are supported. SVI and sub-interfaces are not supported.

```
router ospf 1
interface Ethernet1/16
ip address 192.0.2.2/24
ip router ospf 1 area 0.0.0.0
no shutdown
interface Ethernet1/17
port-type fabric
ip address 192.0.2.3/24
ip router ospf 1 area 0.0.0.0
no shutdown
interface Ethernet1/40
port-type fabric
ip address 192.0.2.4/24
ip router ospf 1 area 0.0.0.0
no shutdown
interface Ethernet1/41
port-type fabric
ip address 192.0.2.5/24
ip router ospf 1 area 0.0.0.0
no shutdown
```

Note All ports connected to spines must be port-type fabric.

VXLAN Configuration

Note Configuring **advertise virtual-rmac** (NVE) and **advertise-pip** (BGP) are required steps. For more information, see the Configuring vPC Multi-Homing, on page 221 chapter.

Configuring VLANs and SVI

```
vlan 10
vn-segment 10010
vlan 101
vn-segment 10101
interface Vlan101
no shutdown
mtu 9216
vrf member vxlan-10101
no ip redirects
ip forward
ipv6 address use-link-local-only
no ipv6 redirects
interface vlan10
no shutdown
mtu 9216
vrf member vxlan-10101
no ip redirects
ip address 192.0.2.102/24
ipv6 address 2001:DB8:0:1::1/64
no ipv6 redirects
fabric forwarding mode anycast-gateway
```

Configuring Virtual Port Channel

```
interface Ethernet1/3
switchport
switchport mode trunk
channel-group 100
no shutdown
exit
interface Ethernet1/39
switchport
switchport mode trunk
channel-group 101
no shutdown
interface Ethernet1/46
switchport
switchport mode trunk
channel-group 102
no shutdown
interface port-channel100
vpc 100
interface port-channel101
vpc 101
interface port-channel102
vpc 102
exit
```

Migrating from vPC to vPC Fabric Peering

This procedure contains the steps to migration from a regular vPC to vPC Fabric Peering.

Any direct Layer 3 link between vPC peers should be used only for peer-keep alive. This link should not be used to advertise paths for vPC Fabric Peering loopbacks.

Note This migration is disruptive.

Before you begin

We recommend that you shut all physical Layer 2 links between the vPC peers before migration. We also recommend that you map VLANs with vn-segment before or after migration.

SUMMARY STEPS

1. **configure terminal**
2. **show vpc**
3. **show port-channel summary**
4. **interface ethernet** *slot/port*
5. **no channel-group**
6. Repeat steps 4 and 5 for each interface.
7. **show running-config vpc**
8. **vpc domain** *domain-id*
9. **virtual peer-link destination** *dest-ip* **source** *source-ip*
10. **interface {ethernet | port-channel}** *value*
11. **port-type fabric**
12. (Optional) **show vpc fabric-ports**
13. **hardware access-list tcam region ing-flow-redirect** *tcam-size*
14. **copy running-config startup-config**
15. **reload**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**show vpc** **Example:** `switch(config)# show vpc`	Determine the number of members in the port channel.
Step 3	**show port-channel summary** **Example:** `switch(config)# show port-channel summary`	Determine the number of members.
Step 4	**interface ethernet** *slot/port* **Example:**	Specifies the interface you are configuring. **Note** This is the peer link port channel.

	Command or Action	Purpose
	`switch(config)# interface ethernet 1/4`	
Step 5	**no channel-group** **Example:** `switch(config-if)# no channel-group`	Remove vPC peer-link port-channel members. **Note** Disruption occurs following this step.
Step 6	Repeat steps 4 and 5 for each interface. **Example:**	
Step 7	**show running-config vpc** **Example:** `switch(config-if)# show running-config vpc`	Determine the vPC domain.
Step 8	**vpc domain** *domain-id* **Example:** `switch(config-if)# vpc domain 100`	Enter vPC domain configuration mode.
Step 9	**virtual peer-link destination** *dest-ip* **source** *source-ip* **Example:** `switch(config-vpc-domain)# virtual peer-link destination 192.0.2.1 source 192.0.2.100`	Specify the destination and source IP addresses for vPC fabric peering.
Step 10	**interface {ethernet | port-channel}** *value* **Example:** `switch(config-if)# interface Ethernet1/17`	Specifies the L3 underlay interface you are configuring.
Step 11	**port-type fabric** **Example:** `switch(config-if)# port-type fabric`	Configures port-type fabric for underlay interface. **Note** All ports connected to spines must be port-type fabric.
Step 12	(Optional) **show vpc fabric-ports** **Example:** `switch# show vpc fabric-ports`	Displays the fabric ports connected to spine.
Step 13	**hardware access-list tcam region ing-flow-redirect** *tcam-size* **Example:** `switch(config-vpc-domain)# hardware access-list tcam region ing-flow-redirect 512`	Perform TCAM carving. The minimum size for Ingress-Flow-redirect TCAM region size is 512. Also ensure it is configured in multiples of 512.
Step 14	**copy running-config startup-config** **Example:** `switch(config-vpc-domain)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

	Command or Action	Purpose
Step 15	**reload** **Example:** `switch(config-vpc-domain)# reload`	Reboots the switch.

Verifying vPC Fabric Peering Configuration

To display the status for the vPC Fabric Peering configuration, enter one of the following commands:

Table 11: vPC Fabric Peering Verification Commands

Command	Purpose
show vpc fabric-ports	Displays the fabric ports state.
show vpc	Displays information about vPC Fabric Peering mode.
show vpc virtual-peerlink vlan consistency	Displays the VLANs which are not associated with vn-segment.

Example of the show vpc fabric-ports Command

```
switch# show vpc fabric-ports
Number of Fabric port : 9
Number of Fabric port active : 9

Fabric Ports State
-------------------------------------
Ethernet1/9 UP
Ethernet1/19/1  ( port-channel151 ) UP
Ethernet1/19/2  ( port-channel151 ) UP
Ethernet1/19/3 UP
Ethernet1/19/4 UP
Ethernet1/20/1 UP
Ethernet1/20/2  ( port-channel152 ) UP
Ethernet1/20/3  ( port-channel152 ) UP
Ethernet1/20/4  ( port-channel152 ) UP
```

Example of the show vpc Command

```
switch# show vpc
Legend:
                 (*) - local vPC is down, forwarding via vPC peer-link

vPC domain id                     : 3
Peer status                       : peer adjacency formed ok
vPC keep-alive status             : peer is alive
Configuration consistency status  : success
Per-vlan consistency status       : success
Type-2 consistency status         : success
vPC role                          : primary
Number of vPCs configured         : 1
Peer Gateway                      : Enabled
Dual-active excluded VLANs        : -
```

```
Graceful Consistency Check        : Enabled
Auto-recovery status              : Enabled, timer is off.(timeout = 240s)
Delay-restore status              : Timer is off.(timeout = 30s)
Delay-restore SVI status          : Timer is off.(timeout = 10s)
Operational Layer3 Peer-router    : Disabled
Virtual-peerlink mode             : Enabled

vPC Peer-link status
---------------------------------------------------------------------
id    Port   Status Active vlans
--    ----   ------ -------------------------------------------------
1     Po100  up     1,56,98-600,1001-3401,3500-3525

vPC status
----------------------------------------------------------------------------
Id    Port          Status Consistency Reason               Active vlans
--    ------------  ------ ----------- ------               ---------------
101   Po101         up     success     success              98-99,1001-280
                                                            0

Please check "show vpc consistency-parameters vpc <vpc-num>" for the
consistency reason of down vpc and for type-2 consistency reasons for
any vpc.

ToR_B1#
```

Example of the show vpc virtual-peerlink vlan consistency Command

```
switch# show vpc virtual-peerlink vlan consistency
Following vlans are inconsistent
23
switch#
```

CHAPTER 22

Configuring Seamless Integration of EVPN with L3VPN (MPLS LDP)

This chapter contains the following sections:

Information About Configuring Seamless Integration of EVPN with L3VPN (MPLS LDP)

Data center deployments have adopted VXLAN EVPN for its benefits like EVPN control-plane learning, multitenancy, seamless mobility, redundancy, and easier POD additions. Similarly, the Core is either an LDP-based MPLS L3VPN network or transitioning from traditional an MPLS L3VPN LDP-based underlay to a more sophisticated solution like segment routing (SR). Segment routing is adopted for its benefits like unified IGP and MPLS control planes, simpler traffic engineering methods, easier configuration, and SDN adoption.

Guidelines and Limitations for Configuring Seamless Integration of EVPN with L3VPN (MPLS LDP)

The following are the guidelines and limitations for Configuring Seamless Integration of EVPN with L3VPN (MPLS LDP):

The following features are supported:

- Cisco Nexus 9504 and 9508 switches with -R and -RX line cards.
- Layer 3 orphans
- 256 peers/nodes within a VXLAN DC domain
- 24,000 ECMP routes is supported on -RX line cards.

Note If you enter the **no hardware profile mpls extended-ecmp** command, the mode is switched to 4 K ECMP routes. This is applicable only when the line card is -RX and the ECMP group has exactly 2 paths.

- The Egress RACL (e-RACL) TCAM and MPLS Extended ECMP features are mutually exclusive. To enable MPLS Extended ECMP (**hardware profile mpls extended-ecmp)** on the Cisco Nexus N9K-X9636C-RX line card, set the e-RACL TCAM carving to 0.

The following features are not supported:

- Subnet stretches across the DC domain
- vPC
- SVI/Subinterfaces

Configuring Seamless Integration of EVPN with L3VPN (MPLS LDP)

These configuration steps are required on a Border Leaf switch to import and re-originate the routes from a VXLAN domain to an MPLS domain and back to a VXLAN domain.

SUMMARY STEPS

1. **configure terminal**
2. **feature mpls l3vpn**
3. **feature mpls ldp**
4. **nv overlay evpn**
5. **router bgp** *number*
6. **address-family ipv4 unicast**
7. **redistribute direct route-map** *route-map-name*
8. **exit**
9. **address-family l2vpn evpn**
10. **exit**
11. **neighbor** *address* **remote-as** *number*
12. **update-source** *type/id*
13. **ebgp-multihop** *ttl-value*
14. **address-family ipv4 unicast**
15. **send-community extended**
16. **exit**
17. **address-family vpnv4 unicast**
18. **send-community extended**
19. **import l2vpn evpn reoriginate**
20. **neighbor** *address* **remote-as** *number*

21. **address-family ipv4 unicast**
22. **send-community extended**
23. **address-family ipv6 unicast**
24. **send-community extended**
25. **address-family l2vpn evpn**
26. **send-community extended**
27. **import vpn unicast reoriginate**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**feature mpls l3vpn** **Example:** `switch# feature mpls l3vpn`	Enables the MPLS Layer 3 VPN feature.
Step 3	**feature mpls ldp** **Example:** `switch# feature mpls ldp`	Enables the MPLS Label Distribution Protocol (LDP).
Step 4	**nv overlay evpn** **Example:** `switch(config)# nv overlay evpn`	Enables the EVPN control plane for VXLAN.
Step 5	**router bgp** *number* **Example:** `switch(config)# router bgp 100`	Configures BGP. The value of the *number* argument is from 1 to 4294967295.
Step 6	**address-family ipv4 unicast** **Example:** `switch(config-router)# address-family ipv4 unicast`	Configures the address family for IPv4.
Step 7	**redistribute direct route-map** *route-map-name* **Example:** `switch(config-router-af)# redistribute direct route-map passall`	Configures the directly connected route map.
Step 8	**exit** **Example:** `switch(config-router-af)# exit`	Exits command mode.
Step 9	**address-family l2vpn evpn** **Example:**	Configures the L2VPN address family.

	Command or Action	Purpose
	`switch(config-router)# `**`address-family l2vpn evpn`**	
Step 10	**exit** **Example:** `switch(config-router-af)# `**`exit`**	Exits command mode.
Step 11	**neighbor** *address* **remote-as** *number* **Example:** `switch(config-router)# `**`neighbor 108.108.108.108 remote-as 22`**	Configures a BGP neighbor. The range of the *number* argument is from 1 to 65535.
Step 12	**update-source** *type/id* **Example:** `switch(config-router-neighbor)# `**`update-source loopback100`**	Specifies the source of the BGP session and updates.
Step 13	**ebgp-multihop** *ttl-value* **Example:** `switch(config-router-neighbor)# `**`ebgp-multihop 10`**	Specifies the multihop TTL for the remote peer. The range of *ttl-value* is from 2 to 255.
Step 14	**address-family ipv4 unicast** **Example:** `switch(config-router-neighbor)# `**`address-family ipv4 unicast`**	Configures the unicast sub-address family.
Step 15	**send-community extended** **Example:** `switch(config-router-neighbor-af)# `**`send-community extended`**	Configures the community attribute for this neighbor.
Step 16	**exit** **Example:** `switch(config-router-neighbor-af)# `**`exit`**	Exits command mode.
Step 17	**address-family vpnv4 unicast** **Example:** `switch(config-router-neighbor)# `**`address-family vpnv4 unicast`**	Configures the address family for IPv4.
Step 18	**send-community extended** **Example:** `switch(config-router)# `**`send-community extended`**	Sends the extended community attribute.
Step 19	**import l2vpn evpn reoriginate** **Example:**	Reoriginates the route with a new RT.

	Command or Action	Purpose
	`switch(config-router)# import l2vpn evpn reoriginate`	
Step 20	**neighbor** *address* **remote-as** *number* **Example:** `switch(config-router)# neighbor 175.175.175.2 remote-as 1`	Defines the neighbor.
Step 21	**address-family ipv4 unicast** **Example:** `switch(config-router)# address-family ipv4 unicast`	Configures the address family for IPv4.
Step 22	**send-community extended** **Example:** `switch(config-router)# send-community extended`	Configures the community for BGP neighbors.
Step 23	**address-family ipv6 unicast** **Example:** `switch(config-router)# address-family ipv6 unicast`	Configures the IPv6 unicast address family. This is required for IPv6 over VXLAN with an IPv4 underlay.
Step 24	**send-community extended** **Example:** `switch(config-router)# send-community extended`	Configures the community for BGP neighbors.
Step 25	**address-family l2vpn evpn** **Example:** `switch(config-router)# address-family l2vpn evpn`	Configures the L2VPN address family.
Step 26	**send-community extended** **Example:** `switch(config-router)# send-community extended`	Configures the community for BGP neighbors.
Step 27	**import vpn unicast reoriginate** **Example:** `switch(config-router)# import vpn unicast reoriginate`	Reoriginates the route with a new RT.

Configuring Seamless Integration of EVPN with L3VPN (MPLS LDP)

CHAPTER 23

Configuring Seamless Integration of EVPN with L3VPN (MPLS SR)

This chapter contains the following sections:

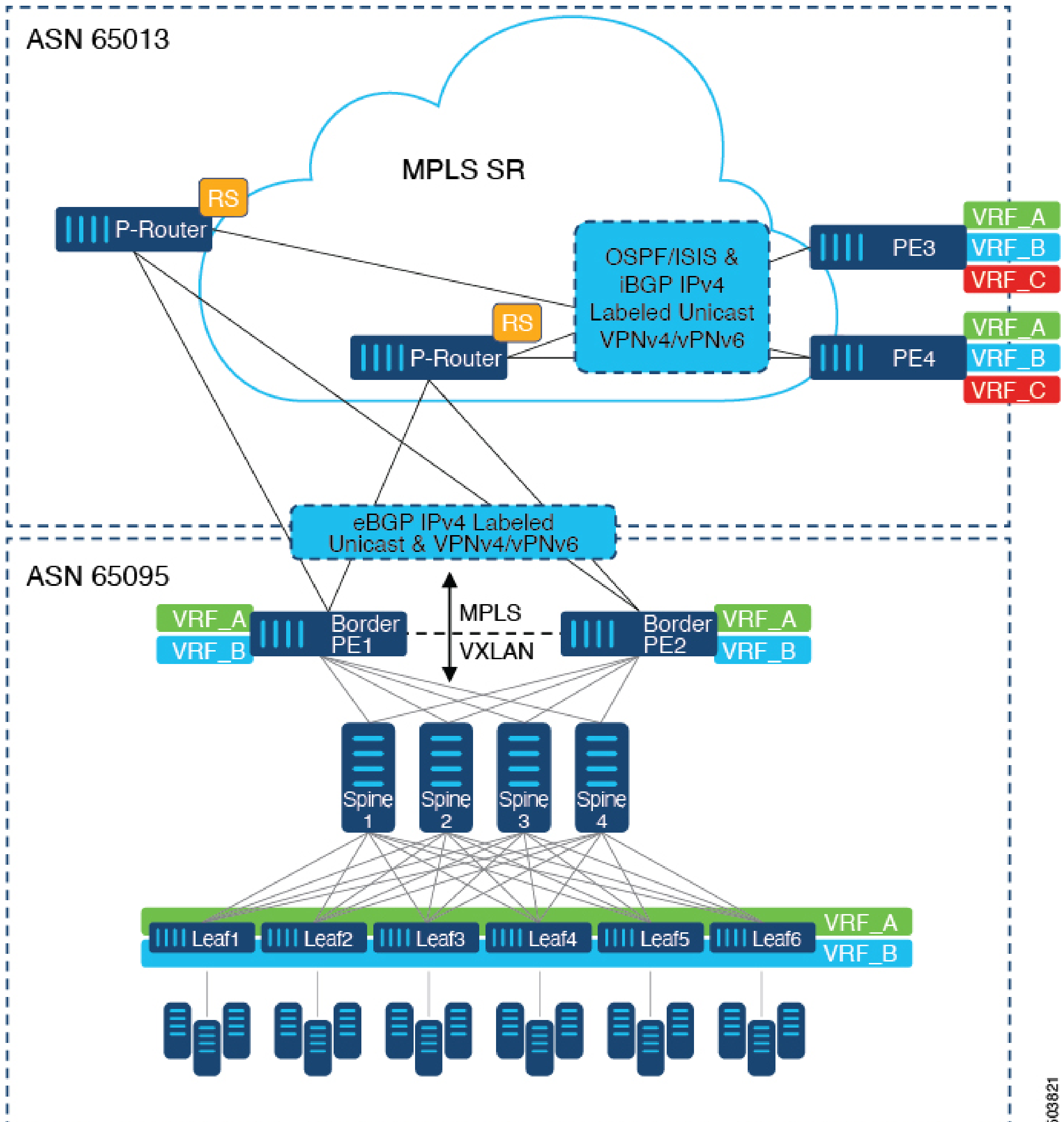
Information About Configuring Seamless Integration of EVPN with L3VPN (MPLS SR)

Data Center (DC) deployments have adopted VXLAN EVPN for its benefits such as EVPN control-plane learning, multi tenancy, seamless mobility, redundancy, and easier horizontal scaling. Similarly, the Core network transitions to different technologies with their respective capabilities. MPLS with Label Distribution Protocol (LDP) and Layer-3 VPN (L3VPN) is present in many Core networks interconnecting Data Centers. With the technology evolution, a transformation from the traditional MPLS L3VPN with LDP-based underlay to MPLS-based Segment Routing (SR) with L3VPN, became available. Segment Routing is adopted for its benefits such as:

- Unified IGP and MPLS control planes
- Simpler traffic engineering methods

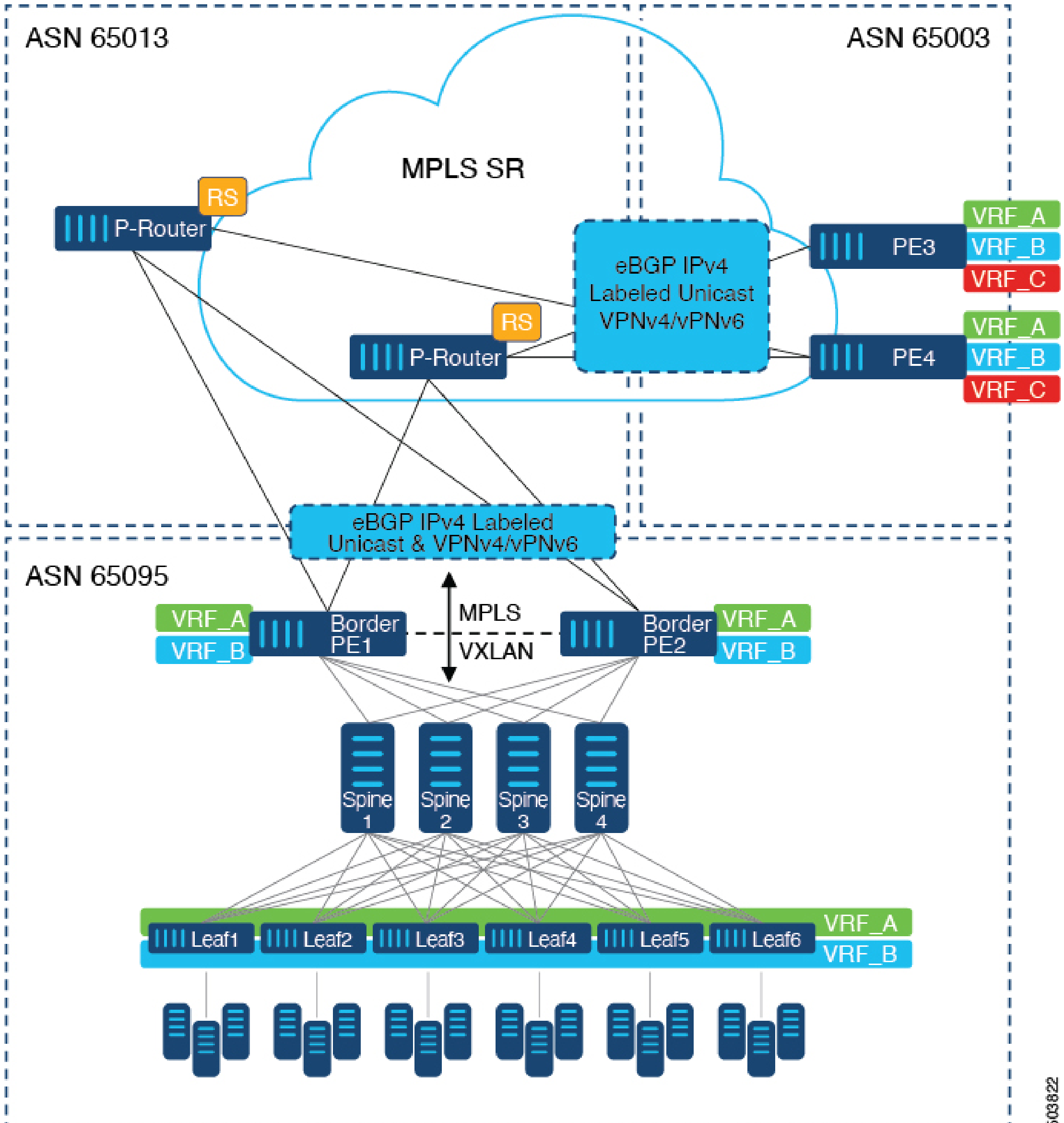
With the Data center (DC) established on VXLAN EVPN and the Core network requiring multi-tenant capable transport, there is a natural necessity to seamless integration. To provide this seamless integration between different control-plane protocols and encapsulations, in this case here from VXLAN to an MPLS-based Core network, the Cisco Nexus 9000 Series Switch provides the Border Provider Edge (Border PE) capability by interfacing the Data Center and the Core routers (Provider Routers or Provider Edge-Routers).

Figure 39: Topology with DC to Core Network Domain Separation



In the above figure, a single Data Center Fabric running VXLAN EVPN is depicted. The VRFs (VRF_A, VRF_B) present in the Data Center require to be extended over a WAN/Core running MPLS-based Segment Routing (MPLS-SR). The Data Center Fabrics Border switches acts as Border Provider Edge (Border PE1, Border PE2) interconnecting VXLAN BGP EVPN with MPLS-SR with L3VPN (VPNv4/VPNv6). The BPEs are interconnected with the Provider Router (P-Router) via eBGP using the IPv4 Labeled-Unicast as well as the VPNv4/VPNv6 Address-Family (AF). The P-Router act as BGP Route-Reflector for the mentioned AF and relays the necessary routes to the MPLS-SR Provider Edge (PE3, PE4) via iBGP. Beyond the usage of BGP as the control-plane, between the MPLS-SR nodes within the same Autonomous System (AS) uses a IGP (OSPF or ISIS) for label distribution. From the PEs shown in the above figure (PE3, PE4), Inter-AS Option A can be used to extend the Data Center or Core network VRFs to another external network. Even as this diagram shows only one Data Center, the MPLS-SR network can be used to interconnect multiple Data Center Fabrics.

Figure 40: Multiple Administrative Domains within the Core network



An alternative deployment scenario is when the Core network is separate into multiple Administrative Domains or Autonomous Systems (AS). In the above figure, a single Data Center Fabric running VXLAN EVPN is depicted. The VRFs (VRF_A, VRF_B) present in the Data Center requires to be extended over a WAN/Core running MPLS-based Segment Routing (MPLS-SR). The Data Center Fabrics Border switches acts as Border Provider Edge (Border PE1, Border PE2) interconnecting VXLAN BGP EVPN with MPLS-SR with L3VPN (VPNv4/VPNv6). The BPEs are interconnected with the Provider Router (P-Router) via eBGP using the IPv4 Labeled-Unicast as well as the VPNv4/VPNv6 Address-Family (AF). The P-Router act as BGP Route Server for the mentioned AF and relays the necessary routes to the MPLS-SR Provider Edge (PE3, PE4) via eBGP; no other control-plane protocol is used between the MPLS-SR nodes. Similar as in the previous scenario, the PEs (PE3, PE4) can operate with Inter-AS Option A to extend the Data Center or Core network VRFs to external network. Even as this diagram shows only one Data Center, the MPLS-SR network can be used to interconnect multiple Data Center Fabrics.

For additional information on MPLS SR, see the *Cisco Nexus 9000 Series NX-OS Label Switching Configuration Guide*.

Guidelines and Limitations for Configuring Seamless Integration of EVPN with L3VPN (MPLS SR)

Feature	Cisco Nexus 9300-FX2, FX3, GX Platform Switches	Cisco Nexus 9504 and 9508 switches with -R Line Cards	Comments
VXLAN EVPN to SR-L3VPN	Yes	Yes	Extend Layer 3 connectivity between different DC pods Underlay IGP/BGP with SR extensions.
VXLAN EVPN to SR-L3VPN	Yes	Yes	Extend Layer 3 connectivity between DC POD running VXLAN and any domain (DC or CORE) running SR.
VXLAN EVPN to MPLS L3VPN (LDP)	No	Yes	Underlay is LDP.

The following Cisco Nexus platform switches support seamless integration of EVPN with L3VPN (MPLS SR):

- 9336C-FX2 switches
- 93240YC-FX2 switches
- 9300-FX3 platform switches
- 9300-GX platform switches
- 9504 and 9508 platform switches with 96136YC-R and 9636C-RX line cards (The 9636C-R and 9636Q-R line cards are not supported.)

The following features are supported with seamless integration of EVPN with L3VPN (MPLS SR):

- Host Facing (Downlinks towards)
    - Individual Layer-3 interfaces (orphan ports)
    - Layer-3 Port-Channel
    - Layer-3 Sub-interfaces
    - Inter-AS Option A (often also called VRF-lite)
- Core Facing (Uplinks towards VXLAN)
    - Individual Layer-3 interfaces

    - Layer-3 Port-Channel

- Core Facing (Uplinks towards MPLS SR)
    - Individual Layer-3 interface
    - Per-VRF labels
    - VPN label statistics

- End-to-EndTime to Live (TTL) and Explicit Congestion Notification (ECN) with pipe-mode only.
- MPLS SegmentRouting and MPLS LDP cannot be configured at the same time on a Cisco Nexus 9504 and 9508 platform switches with Cisco Nexus 96136YC-R and Cisco Nexus 9636C-RX line cards.

The VXLAN-to-SR handoff QoS value is preserved during handoff and propagated from VXLAN tunnel packets to SR-tunneled packets for Cisco Nexus 9336C-FX2, 93240YC-FX2, 9300-FX3, and 9300-GX platform switches.

The following features are not supported with seamless integration of EVPN with L3VPN (MPLS SR):

- Distributed Anycast Gateway or First-Hop Redundancy Protocol like HSRP, VRRP or GLBP.
- vPC for redundant Host or Network Service attachment.
- SVI/Sub-interfaces for Core facing uplinks (MPLS or VXLAN).
- SVI/Sub-interfaces with configured MAC addresses.
- MPLS Segment Routing and Border Gateway (BGW for VXLAN Multi-Site) cannot be configured at the same time.
- Layer-2 for stretched Subnet across the MPLS-SR domain
- No-drop for VXLAN/SR and SR/VXLAN handoff, for Cisco Nexus 9336C-FX2, 93240YC-FX2, and 9300-FX3 platform switches
- Statistics, for Cisco Nexus 9504 and 9508 platform switches with 96136YC-R and 9636C-RX line cards
- Priority flow control (PFC), for Cisco Nexus 9336C-FX2, 93240YC-FX2, 9300-FX3, and 9300-GX platform switches

Configuring Seamless Integration of EVPN with L3VPN (MPLS SR)

The following procedure for Border Provider Edge (Border PE) imports and reoriginates the routes from the VXLAN domain to the MPLS domain and in the other direction.

SUMMARY STEPS

1. **configure terminal**
2. **feature-set mpls**
3. **nv overlay evpn**

4. **feature bgp**
5. **feature mpls l3vpn**
6. **feature mpls segment-routing**
7. **feature interface-vlan**
8. **feature vn-segment-vlan-based**
9. **feature nv overlay**
10. **router bgp** *autonomous-system-number*
11. **address-family ipv4 unicast**
12. **network** *address*
13. **allocate-label all**
14. **exit**
15. **neighbor** *address* **remote-as** *number*
16. **update-source** *type/id*
17. **address-family l2vpn evpn**
18. **send-community both**
19. **import vpn unicast reoriginate**
20. **exit**
21. **neighbor** *address* **remote-as** *number*
22. **update-source** *type/id*
23. **address-family ipv4 labeled-unicast**
24. **send-community both**
25. **exit**
26. **neighbor** *address* **remote-as** *number*
27. **update-source** *type/id*
28. **ebgp-multihop** *number*
29. **address-family vpnv4 unicast**
30. **send-community both**
31. **import l2vpn evpn reoriginate**
32. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**feature-set mpls** **Example:** `switch(config)# feature-set mpls`	Enables the MPLS feature set.
Step 3	**nv overlay evpn** **Example:** `switch(config)# nv overlay evpn`	Enables VXLAN.

	Command or Action	Purpose
Step 4	**feature bgp** **Example:** `switch(config)# feature bgp`	Enables BGP.
Step 5	**feature mpls l3vpn** **Example:** `switch(config)# feature mpls l3vpn`	Enables Layer 3 VPN. **Note** Feature mpls l3vpn requires feature mpls segment-routing.
Step 6	**feature mpls segment-routing** **Example:** `switch(config)# feature mpls segment-routing`	Enables Segment Routing.
Step 7	**feature interface-vlan** **Example:** `switch(config)# feature interface-vlan`	Enables the interface VLAN.
Step 8	**feature vn-segment-vlan-based** **Example:** `switch(config)# feature vn-segment-vlan-based`	Enables the VLAN-based VN segment.
Step 9	**feature nv overlay** **Example:** `switch(config)# feature nv overlay`	Enables VXLAN.
Step 10	**router bgp** *autonomous-system-number* **Example:** `switch(config)# router bgp 65095`	Configures BGP. The value of *autonomous-system-number* is from 1 to 4294967295.
Step 11	**address-family ipv4 unicast** **Example:** `switch(config-router)# address-family ipv4 unicast`	Configures the address family for IPv4.
Step 12	**network** *address* **Example:** `switch(config-router-af)# network 10.51.0.51/32`	Injects prefixes into BGP for the MPLS-SR domain. **Note** All viable next-hops for MPLS-SR tunnel deposition on the Border PE must be advertised via the network statement (/32 only).
Step 13	**allocate-label all** **Example:** `switch(config-router-af)# allocate-label all`	Configures label allocation for every prefix injected via the network statement.
Step 14	**exit** **Example:** `switch(config-router-af)# exit`	Exits command mode.

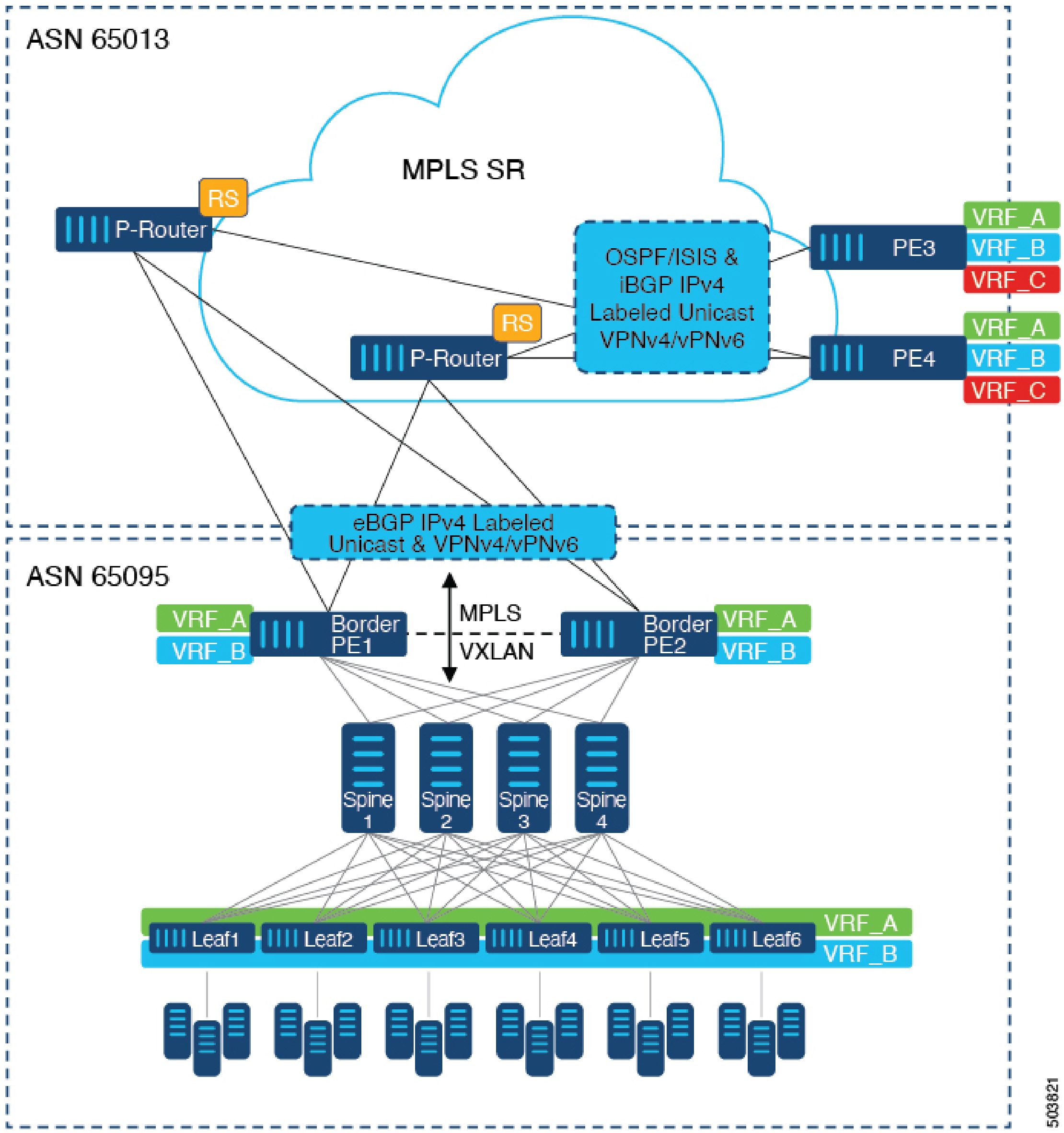
	Command or Action	Purpose
Step 15	**neighbor** *address* **remote-as** *number* **Example:** `switch(config-router)# neighbor 10.95.0.95 remote-as 65095`	Defines the iBGP neighbor IPv4 address and remote Autonomous-System (AS) number towards the Route-Reflector.
Step 16	**update-source** *type/id* **Example:** `switch(config-router)# update-source loopback0`	Defines the interface for eBGP peering.
Step 17	**address-family l2vpn evpn** **Example:** `switch(config-router)# address-family l2vpn evpn`	Configures the L2VPN EVPN address family.
Step 18	**send-community both** **Example:** `switch(config-router-af)# send-community both`	Configures the community for BGP neighbors.
Step 19	**import vpn unicast reoriginate** **Example:** `switch(config-router-af)# import vpn unicast reoriginate`	Reoriginates the route with a new Route-Target. It can be extended to use an optional route-map.
Step 20	**exit** **Example:** `switch(config-router-af)# exit`	Exits command mode.
Step 21	**neighbor** *address* **remote-as** *number* **Example:** `switch(config-router)# neighbor 10.51.131.131 remote-as 65013`	Defines the eBGP neighbor IPv4 address and remote Autonomous-System (AS) number towards the P-Router.
Step 22	**update-source** *type/id* **Example:** `switch(config-router)# update-source Ethernet1/1`	Defines the interface for eBGP peering.
Step 23	**address-family ipv4 labeled-unicast** **Example:** `switch(config-router)# address-family ipv4 labeled-unicast`	Configures the address family for IPv4 labeled-unicast.
Step 24	**send-community both** **Example:** `switch(config-router-af)# send-community both`	Configures the community for BGP neighbors.

	Command or Action	Purpose
Step 25	**exit** **Example:** `switch(config-router-af)# exit`	Exits command mode.
Step 26	**neighbor** *address* **remote-as** *number* **Example:** `switch(config-router)# neighbor 10.131.0.131 remote-as 65013`	Defines the eBGP neighbor IPv4 address and remote Autonomous-System (AS) number.
Step 27	**update-source** *type/id* **Example:** `switch(config-router)# update-source loopback0`	Defines the interface for eBGP peering.
Step 28	**ebgp-multihop** *number* **Example:** `switch(config-router)# ebgp-multihop 5`	Specifies multihop TTL for the remote peer. The range of *number*is from 2 to 255.
Step 29	**address-family vpnv4 unicast** **Example:** `switch(config-router)# address-family vpnv4 unicast`	Configures the address family for VPNv4 or VPNv6.
Step 30	**send-community both** **Example:** `switch(config-router-af)# send-community both`	Configures the community for BGP neighbors.
Step 31	**import l2vpn evpn reoriginate** **Example:** `switch(config-router-af)# import l2vpn evpn reoriginate`	Reoriginates the route with a new Route-Target. It can be extended to use an optional route-map.
Step 32	**exit** **Example:** `switch(config-router-af)# exit`	Exits command mode.

Example Configuration for Configuring Seamless Integration of EVPN with L3VPN (MPLS SR)

Scenario - 1 with DC to Core Network Domain Separation and IGP within MPLS-SR network.

Figure 41: Topology with DC to Core Network Domain Separation



The following is a sample CLI configuration that is required to import and reoriginate the routes from the VXLAN domain to the MPLS domain and in the reverse direction. The sample CLI configuration represents only the necessary configuration for the respective roles.

Border PE

```
hostname BL51-N9336FX2
install feature-set mpls

feature-set mpls

feature bgp
feature mpls l3vpn
feature mpls segment-routing
```

```
feature ospf
feature interface-vlan
feature vn-segment-vlan-based
feature nv overlay

nv overlay evpn

mpls label range 16000 23999 static 6000 8000

segment-routing
  mpls
    connected-prefix-sid-map
      address-family ipv4
        10.51.0.51/32 index 51

vlan 2000
  vn-segment 50000

vrf context VRF_A
  vni 50000
  rd auto
  address-family ipv4 unicast
    route-target both auto
    route-target both auto evpn
    route-target import 50000:50000
    route-target export 50000:50000
  address-family ipv6 unicast
    route-target both auto
    route-target both auto evpn
    route-target import 50000:50000
    route-target export 50000:50000

interface Vlan2000
  no shutdown
  vrf member VRF_A
  no ip redirects
  ip forward
  ipv6 address use-link-local-only
  no ipv6 redirects

interface nve1
  no shutdown
  host-reachability protocol bgp
  source-interface loopback1
  member vni 50000 associate-vrf

interface Ethernet1/1
  description TO_P-ROUTER
  ip address 10.51.131.51/24
  mpls ip forwarding
  no shutdown

interface Ethernet1/36
  description TO_SPINE
  ip address 10.95.51.51/24
  ip router ospf 10 area 0.0.0.0
  no shutdown

interface loopback0
  description ROUTER-ID & SR-LOOPBACK
  ip address 10.51.0.51/32
  ip router ospf UNDERLAY area 0.0.0.0

interface loopback1
```

```
  description NVE-LOOPBACK
  ip address 10.51.1.51/32
  ip router ospf UNDERLAY area 0.0.0.0

router ospf UNDERLAY
  router-id 10.51.0.51

router bgp 65095
  address-family ipv4 unicast
    network 10.51.0.51/32
    allocate-label all
!
  neighbor 10.95.0.95
    remote-as 65095
    update-source loopback0
    address-family l2vpn evpn
      send-community
      send-community extended
      import vpn unicast reoriginate
!
  neighbor 10.51.131.131
    remote-as 65013
    update-source Ethernet1/1
    address-family ipv4 labeled-unicast
      send-community
      send-community extended
!
  neighbor 10.131.0.131
    remote-as 65013
    update-source loopback0
    ebgp-multihop 5
    address-family vpnv4 unicast
      send-community
      send-community extended
      import l2vpn evpn reoriginate
    address-family vpnv6 unicast
      send-community
      send-community extended
      import l2vpn evpn reoriginate

!
  vrf VRF_A
    address-family ipv4 unicast
      redistribute direct route-map fabric-rmap-redist-subnet
```

P-Router

```
hostname P131-N9336FX2
install feature-set mpls

feature-set mpls

feature bgp
feature isis
feature mpls l3vpn
feature mpls segment-routing

mpls label range 16000 23999 static 6000 8000

segment-routing
  mpls
    connected-prefix-sid-map
      address-family ipv4
        10.131.0.131/32 index 131
```

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

interface Ethernet1/1
  description TO_BORDER-PE
  ip address 10.51.131.131/24
  ip router isis 10
  mpls ip forwarding
  no shutdown

interface Ethernet1/11
  description TO_PE
  ip address 10.52.131.131/24
  ip router isis 10
  mpls ip forwarding
  no shutdown

interface loopback0
  description ROUTER-ID & SR-LOOPBACK
  ip address 10.131.0.131/32
  ip router isis 10

router isis 10
  net 49.0000.0000.0131.00
  is-type level-2
  address-family ipv4 unicast
    segment-routing mpls

router bgp 65013
  event-history detail
  address-family ipv4 unicast
    allocate-label all
!
  neighbor 10.51.131.51
    remote-as 65095
    update-source Ethernet1/1
    address-family ipv4 labeled-unicast
      send-community
      send-community extended
!
  neighbor 10.51.0.51
    remote-as 65095
    update-source loopback0
    ebgp-multihop 5
    address-family vpnv4 unicast
      send-community
      send-community extended
      route-map RM_NH_UNCH out
    address-family vpnv6 unicast
      send-community
      send-community extended
      route-map RM_NH_UNCH out
!
  neighbor 10.52.131.52
    remote-as 65013
    update-source Ethernet1/11
    address-family ipv4 labeled-unicast
      send-community
      send-community extended
!
  neighbor 10.52.0.52
    remote-as 65013
    update-source loopback0
    address-family vpnv4 unicast
```

```
      send-community
      send-community extended
      route-reflector-client
      route-map RM_NH_UNCH out
    address-family vpnv6 unicast
      send-community
      send-community extended
      route-reflector-client
      route-map RM_NH_UNCH out
```

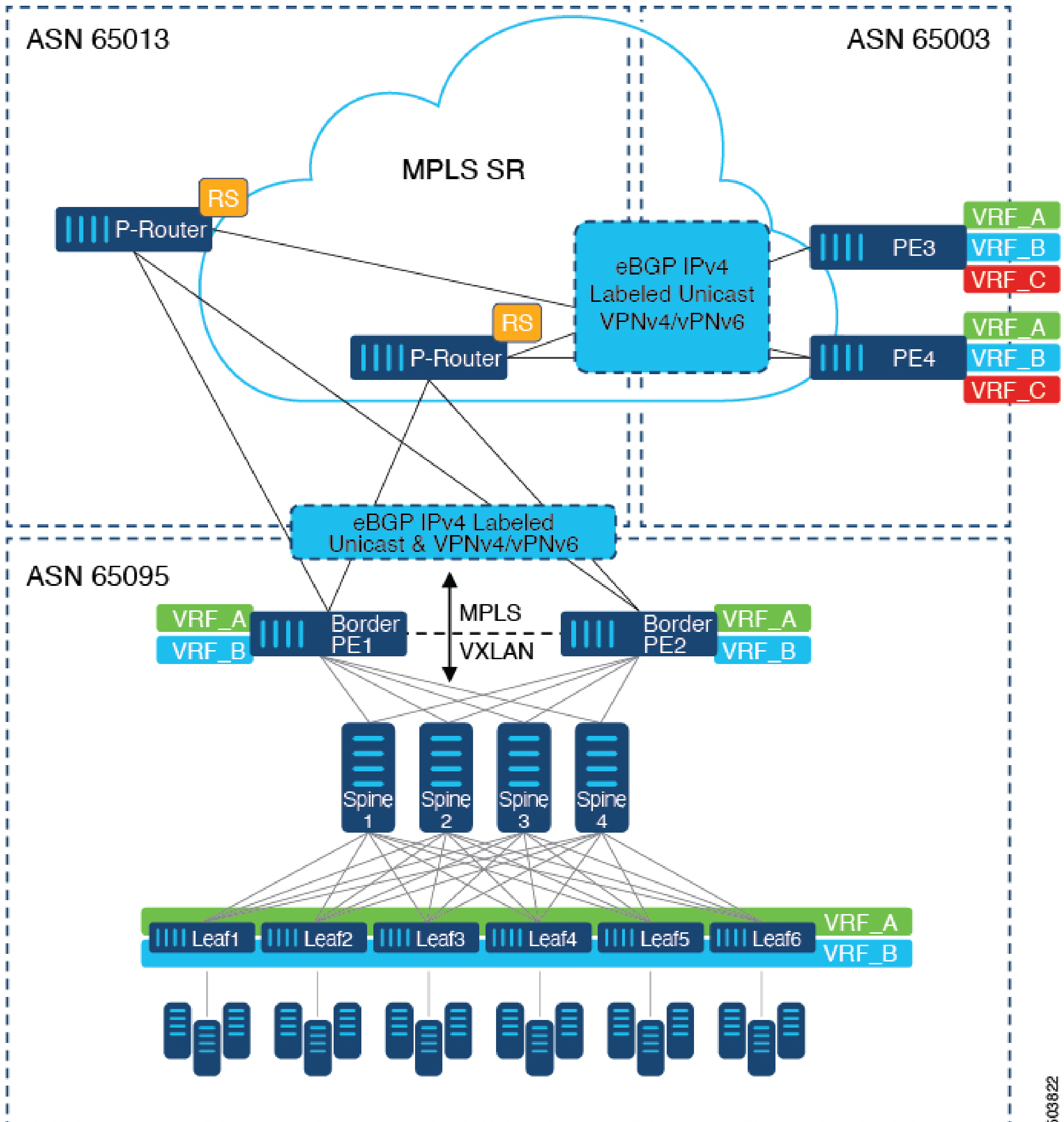
Provider Edge (PE)

```
hostname L52-N93240FX2
install feature-set mpls

feature-set mpls

feature bgp
feature isis
feature mpls l3vpn
feature mpls segment-routing

mpls label range 16000 23999 static 6000 8000

segment-routing
  mpls
    connected-prefix-sid-map
      address-family ipv4
        10.52.0.52/32 index 52

vrf context VRF_A
  rd auto
  address-family ipv4 unicast
    route-target import 50000:50000
    route-target export 50000:50000
  address-family ipv6 unicast
    route-target import 50000:50000
    route-target export 50000:50000

interface Ethernet1/49
  description TO_P-ROUTER
  ip address 10.52.131.52/24
  ip router isis 10
  mpls ip forwarding
  no shutdown

interface loopback0
  description ROUTER-ID & SR-LOOPBACK
  ip address 10.52.0.52/32
  ip router isis 10

router isis 10
  net 49.0000.0000.0052.00
  is-type level-2
  address-family ipv4 unicast
    segment-routing mpls

router bgp 65013
  address-family ipv4 unicast
    network 10.52.0.52/32
    allocate-label all
!
  neighbor 10.52.131.131
    remote-as 65013
    update-source Ethernet1/49
```

```
    address-family ipv4 labeled-unicast
      send-community
      send-community extended
!
  neighbor 10.131.0.131
    remote-as 65013
    update-source loopback0
    address-family vpnv4 unicast
      send-community
      send-community extended
    address-family vpnv6 unicast
      send-community
      send-community extended
!
  vrf VRF_A
    address-family ipv4 unicast
      redistribute direct route-map fabric-rmap-redist-subnet
```

Scenario - 2 with DC to Core and within Core Network Domain Separation (eBGP within MPLS-SR network).

Figure 42: Multiple Administrative Domains within the Core network



The following is a sample CLI configuration that is required to import and reoriginate the routes from the VXLAN domain to the MPLS domain and in the reverse direction. The sample CLI configuration represents only the nodes that are different from Scenario #1, which are the P-Router and the Provider Edge (PE) roles. The Border PE remains the same for both scenarios.

P-Router

```
hostname P131-N9336FX2
install feature-set mpls

feature-set mpls

feature bgp
feature mpls l3vpn
feature mpls segment-routing
```

```
mpls label range 16000 23999 static 6000 8000

segment-routing
  mpls
    connected-prefix-sid-map
      address-family ipv4
        10.131.0.131/32 index 131

route-map RM_NH_UNCH permit 10
  set ip next-hop unchanged

interface Ethernet1/1
  description TO_BORDER-PE
  ip address 10.51.131.131/24
  mpls ip forwarding
  no shutdown

interface Ethernet1/11
  description TO_PE
  ip address 10.52.131.131/24
  mpls ip forwarding
  no shutdown

interface loopback0
  description ROUTER-ID & SR-LOOPBACK
  ip address 10.131.0.131/32
  ip router isis 10

router bgp 65013
  event-history detail
  address-family ipv4 unicast
    network 10.131.0.131/32
    allocate-label all
!
  address-family vpnv4 unicast
    retain route-target all
  address-family vpnv6 unicast
    retain route-target all
!
  neighbor 10.51.131.51
    remote-as 65095
    update-source Ethernet1/1
    address-family ipv4 labeled-unicast
      send-community
      send-community extended
!
  neighbor 10.51.0.51
    remote-as 65095
    update-source loopback0
    ebgp-multihop 5
    address-family vpnv4 unicast
      send-community
      send-community extended
      route-map RM_NH_UNCH out
    address-family vpnv6 unicast
      send-community
      send-community extended
      route-map RM_NH_UNCH out
!
  neighbor 10.52.131.52
    remote-as 65003
    update-source Ethernet1/11
    address-family ipv4 labeled-unicast
```

```
      send-community
      send-community extended
!
  neighbor 10.52.0.52
    remote-as 65003
    update-source loopback0
    ebgp-multihop 5
    address-family vpnv4 unicast
      send-community
      send-community extended
      route-map RM_NH_UNCH out
    address-family vpnv6 unicast
      send-community
      send-community extended
      route-map RM_NH_UNCH out
```

Provider Edge (PE)

```
hostname L52-N93240FX2
install feature-set mpls

feature-set mpls

feature bgp
feature mpls l3vpn
feature mpls segment-routing

mpls label range 16000 23999 static 6000 8000

segment-routing
  mpls
    connected-prefix-sid-map
      address-family ipv4
        10.52.0.52/32 index 52

vrf context VRF_A
  rd auto
  address-family ipv4 unicast
    route-target import 50000:50000
    route-target export 50000:50000
  address-family ipv6 unicast
    route-target import 50000:50000
    route-target export 50000:50000

interface Ethernet1/49
  description TO_P-ROUTER
  ip address 10.52.131.52/24
  mpls ip forwarding
  no shutdown

interface loopback0
  description ROUTER-ID & SR-LOOPBACK
  ip address 10.52.0.52/32
  ip router isis 10

router bgp 65003
  address-family ipv4 unicast
    network 10.52.0.52/32
    allocate-label all
!
  neighbor 10.52.131.131
    remote-as 65013
    update-source Ethernet1/49
    address-family ipv4 labeled-unicast
      send-community
```

```
      send-community extended
!
  neighbor 10.131.0.131
    remote-as 65013
    update-source loopback0
    ebgp-multihop 5
    address-family vpnv4 unicast
      send-community
      send-community extended
    address-family vpnv6 unicast
      send-community
      send-community extended
!
  vrf VRF_A
    address-family ipv4 unicast
      redistribute direct route-map fabric-rmap-redist-subnet
```

CHAPTER 24

Configuring Seamless Integration of EVPN with L3VPN SRv6

This chapter contains the following sections:

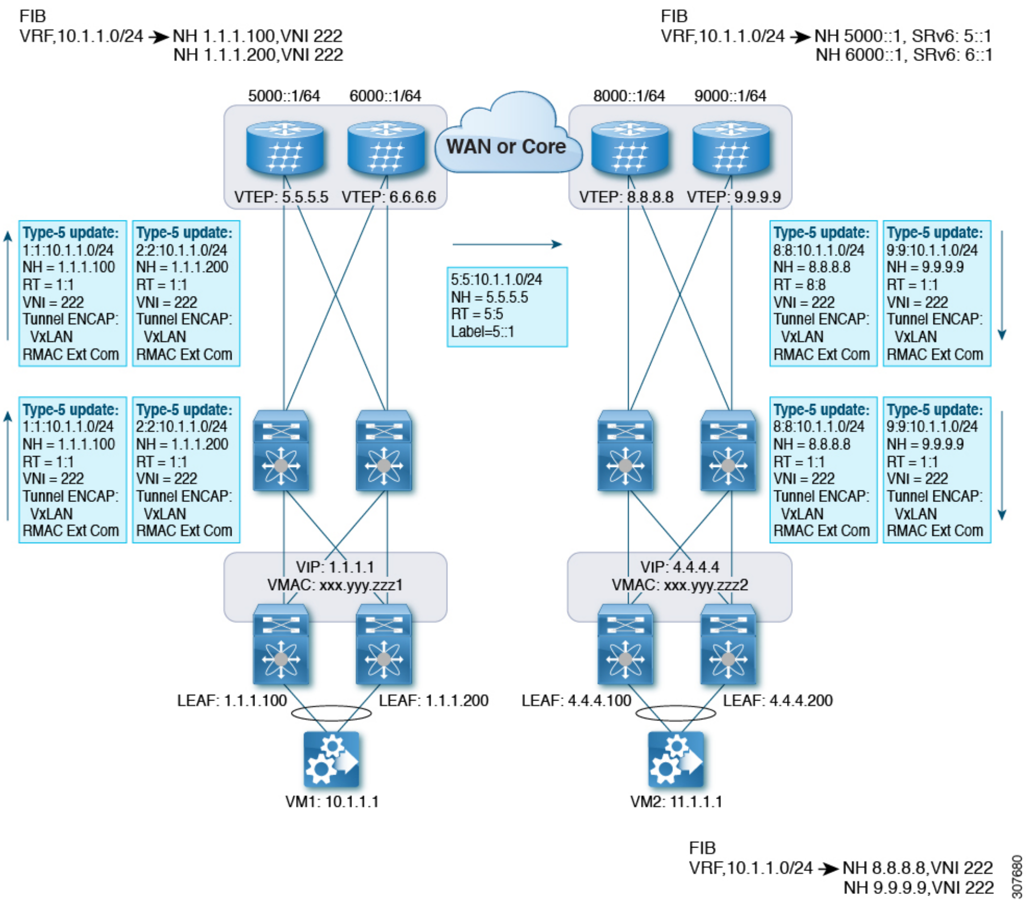
About Seamless Integration of EVPN with L3VPN SRv6 Handoff

Data Center (DC) deployments have adopted VXLAN EVPN for its benefits such as EVPN control-plane learning, multitenancy, seamless mobility, redundancy, and easier POD additions. Similarly, the CORE is either an IP-based L3VPN SRv6 network or transitioning from the IPv6-based L3VPN underlay to a more sophisticated solution like IPv6 Segment Routing (SRv6) for IPv6. SRv6 is adopted for its benefits such as:

- Simpler traffic engineering (TE) methods
- Easier configuration
- SDN adoption

With two different technologies, one within the data center (DC) and one in the Core, there is traffic handoff from VXLAN to an SRv6 core that becomes a necessity at the DCI nodes, which sit at the edge of the DC domain and interface with the Core edge router.

Figure 43: BGP EVPN VXLAN to L3VPN SRv6 Handoff



For traffic ingressing the EVPN-VxLAN fabric, the BGP EVPN routes get imported into a local VRF which contains the RD of the VRF. The bestpath is calculated and installed in the VRF's RIB, then inserted into the L3VPN SRv6 table. Along with the bestpath, the VRF's RD and per-VRF SRv6 SID are included. The L3VPN SRv6 route target is sent with the route, which is advertised to the L3VPN SRv6 peer.

For traffic egressing the EVPN VxLAN fabric, the BGP L3VPN SRv6 routes get imported into a local VRF which contains the RD of the VRF. The bestpath is calculated and installed in the VRF's RIB, then inserted into the EVPN table. Along with the bestpath, the VRF's RD and VNI are included. The EVPN-VXLAN route target is sent with the route, which is advertised to the EVPN-VxLAN peer.

Guidelines and Limitations for EVPN to L3VPN SRv6 Handoff

This feature has the following guidelines and limitations:

- The same RD import is supported for L3VPN SRV6 fabrics.
- The same RD import is not supported for EVPN VXLAN fabrics.
- On a handoff device, do not use the same RD import on the EVPN VXLAN side.
- Beginning with Cisco NX-OS Release 9.3(3), support is added for the following switches:

- Cisco Nexus C93600CD-GX
- Cisco Nexus C9364C-GX
- Cisco Nexus C9316D-GX

Importing L3VPN SRv6 Routes into EVPN VXLAN

The process of handing off routes from the L3VPN SRv6 domain to the EVPN VXLAN fabric requires configuring the import condition for L3VPN SRv6 routes. Routes can be either IPv4 or IPv6. This task configures unidirectional route advertisement into the EVPN VXLAN fabric. For bidirectional advertisement, you must explicitly configure the import condition for the L3VPN SRv6 domain.

Before you begin

Make sure you have a fully configured L3VPN SRv6 fabric. For more information, see "Configuring Layer 3 VPN over SRv6" in the *Cisco Nexus 9000 Series NX-OS SRv6 Configuration Guide*.

SUMMARY STEPS

1. **config terminal**
2. **router bgp** *as-number*
3. **neighbor bgp** *ipv6-address* **remote-as** *as-number*
4. **address family vpnv4 unicast** or **address family vpnv6 unicast**
5. **import l2vpn evpn route-map** *name* **[reoriginate]**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**config terminal**
	Example:
	`switch-1# config terminal`
	`Enter configuration commands, one per line. End with CNTL/Z.`
	`switch-1(config)#`
		Enter configuration mode.

Step 2	**router bgp** *as-number*
	Example:
	`switch-1(config)# router bgp 100`
	`switch-1(config-router)#`
		Enter BGP router configuration mode.

Step 3	**neighbor bgp** *ipv6-address* **remote-as** *as-number*
	Example:
	`switch-1(config-router)# neighbor 1234::1 remote-as 200`
	`switch-1(config-router-neighbor)#`
		Enter BGP router configuration mode.

	Command or Action	Purpose
Step 4	**address family vpnv4 unicast** or **address family vpnv6 unicast** **Example:** `switch-1(config-router-neighbor)# ` **`address-family vpnv4 unicast`** `switch-1(config-router-neighbor-af)#` **Example:** `switch-1(config-router-neighbor)# ` **`address-family vpnv6 unicast`** `switch-1(config-router-neighbor-af)#`	Configure the IPv4 or IPv6 address family for unicast traffic that the EVPN VXLAN will handoff to L3VPN SRv6.
Step 5	**import l2vpn evpn route-map** *name* **[reoriginate]** **Example:** `switch-1(config-router-neighbor-af)# ` **`import l2vpn evpn route-map test reoriginate`** `switch-1(config-router-neighbor-af)#`	Configure the IPv4 or IPv6 address family for unicast traffic that EVPN VXLAN will handoff to L3VPN SRv6. This command enables routes learned from L3VPN SRv6 domain to be advertised to the EVPN VXLAN domain. Using the optional **reoriginate** keyword advertises only domain-specific RTs.

What to do next

For bidirectional route advertisement, configure importing EVPN VXLAN routes into the L3VPN SRv6 domain.

Importing EVPN VXLAN Routes into L3VPN SRv6

The process of handing off routes from the EVPN VXLAN fabric to the L3VPN SRv6 domain requires configuring the import condition for EVPN VXLAN routes. Routes can be either IPv4 or IPv6. This task configures unidirectional route advertisement into the L3VPN SRv6 fabric. For bidirectional advertisement, you must explicitly configure the import condition for the EVPN VXLAN domain.

Before you begin

Make sure you have a fully configured L3VPN SRv6 fabric. For more information, see "Configuring Layer 3 VPN over SRv6" in the *Cisco Nexus 9000 Series NX-OS SRv6 Configuration Guide*.

SUMMARY STEPS

1. **config terminal**
2. **router bgp** *as-number*
3. **neighbor** *ipv6-address* **remote-as** *as-number*
4. **address-family l2vpn evpn**
5. **import vpn unicast route-map** *name* **[reoriginate]**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**config terminal**
	Example:
	`switch-1# config terminal`
	`Enter configuration commands, one per line. End with CNTL/Z.`
	`switch-1(config)#`
		Enter configuration mode.

Step 2	**router bgp** *as-number*
	Example:
	`switch-1(config)# router bgp 200`
	`switch-1(config-router)#`
		Enter BGP router configuration mode.

Step 3	**neighbor** *ipv6-address* **remote-as** *as-number*
	Example:
	`switch-1(config-router)# neighbor 1234::1 remote-as 100`
	`switch-1(config-router-neighbor)#`
		Enter BGP router configuration mode.

Step 4	**address-family l2vpn evpn**
	Example:
	`switch(config-router-neighbor)# address-family l2vpn evpn`
	`switch(config-router-neighbor-af)#`
		Configure the address family for unicast traffic that EVPN VXLAN will handoff to L3VPN SRv6.

Step 5	**import vpn unicast route-map** *name* **[reoriginate]**
	Example:
	`switch-1(config-router-neighbor-af)# import vpn unicast route-map test reoriginate`
	`switch-1(config-router-neighbor-af)#`
		Configure the IPv4 or IPv6 address family for unicast traffic that EVPN VXLAN will handoff to L3VPN SRv6. This command enables routes learned from the EVPN VXLAN domain to be advertised to the L3VPN SRv6 domain. Using the optional **reoriginate** keyword advertises only domain-specific RTs.

What to do next

For bidirectional route advertisement, configure importing L3VPN SRv6 routes into the EVPN VXLAN fabric.

Example Configuration for VXLAN EVPN to L3VPN SRv6 Handoff

```
feature vn-segment-vlan-based
feature nv overlay
feature interface-vlan
nv overlay evpn
feature srv6

vrf context customer1
    vni 10000
    rd auto
```

```
    address-family ipv4 unicast
     route-target both 1:1
     route-target both auto evpn
    address-family ipv6 unicast
     route-target both 1:1
     route-target both auto evpn

segment-routing
  srv6
    encapsulation
     source-address loopback1
    locators
     locator DCI_1
      prefix café:1234::/64

interface loopback0
  ip address 1.1.1.0/32

interface loopback1
  ip address 1.1.1.1/32
  ipv6 address 4567::1/128

interface nve1
  source-interface loopback0
  member vni 10000 associate-vrf
  host-reachability protocol bgp

vlan 100
  vn-segment 10000

interface vlan 100
 ip forward
 ipv6 address use-link-local-only
 vrf member customer1

router bgp 65000
   segment-routing srv6
    locator DCI_1
   neighbor 2.2.2.2 remote-as 200
      remote-as 75000
      address-family l2vpn evpn
      import vpn route-map | reoriginate
   neighbor 1234::1 remote-as 100
      remote-as 65000
      address-family vpnv4 unicast
      import l2vpn evpn route-map | reoriginate
      address-family vpnv6 unicast
      import l2vpn evpn route-map | reoriginate

vrf customer
  segment-routing srv6
  alloc-mode per-vrf
  address-family ipv4 unicast
  address-family ipv6 unicast
```

CHAPTER 25

Configuring Seamless Integration of EVPN (TRM) with MVPN

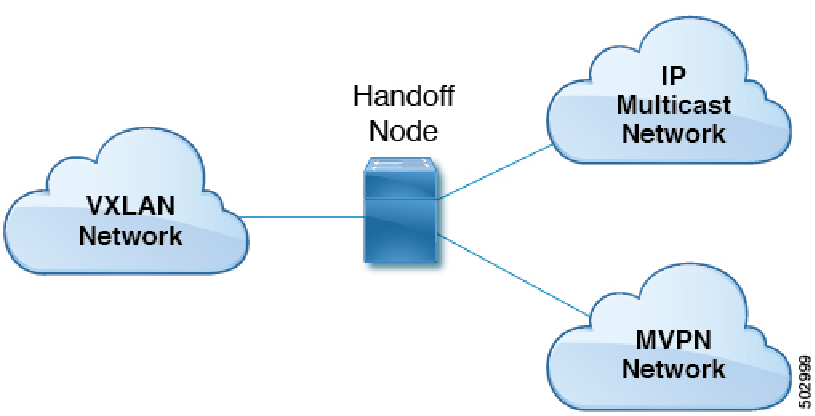
This chapter contains the following sections:

About Seamless Integration of EVPN (TRM) with MVPN (Draft Rosen)

Seamless integration of EVPN (TRM) with MVPN (draft rosen) enables packets to be handed off between a VXLAN network (TRM or TRM Multi-Site) and an MVPN network. To support this feature, VXLAN TRM and MVPN must be supported on a Cisco Nexus device node, the handoff node.

The handoff node is the PE for the MVPN network and the VTEP for the VXLAN network. It connects to the VXLAN, MVPN, and IP multicast networks, as shown in the following figure.

Figure 44: VXLAN - MVPN Handoff Network



Sources and receivers can be in any of the three networks (VXLAN, MVPN, or IP multicast).

All multicast traffic (that is, the tenant traffic from the VXLAN, MVPN, or multicast network) is routed from one domain to another domain. The handoff node acts as the central node. It performs the necessary packet forwarding, encapsulation, and decapsulation to send the traffic to the respective receivers.

Supported RP Positions

The rendezvous point (RP) for the customer (overlay) network can be in any of the three networks (VXLAN, MVPN, or IP multicast).

Table 12: Supported RP Locations

RP Locations	Description
RP in IP network	• The RP can be connected only to the MVPN PE and not to the handoff nodes. • The RP can be connected only to the VXLAN handoff nodes. • The RP can be connected to both the MVPN PE and VXLAN.
RP internal to VXLAN fabric	All VTEPs are RPs inside the VXLAN fabric. All MVPN PEs use the RP configured on the VXLAN fabric.
RP on VXLAN MVPN handoff node	The RP is the VXLAN MVPN handoff node.
RP in MVPN network	The RP is external to the VXLAN network. It's configured on one of the nodes in the MPLS cloud, other than the handoff node.
RP Everywhere (PIM Anycast RP or MSDP-based Anycast RP)	The Anycast RP can be configured on the VXLAN leaf. The RP set can be configured on the handoff node or any MVPN PE.

Guidelines and Limitations for Seamless Integration of EVPN (TRM) with MVPN

This feature has the following guidelines and limitations:

- Only Cisco Nexus 9504 and 9508 platform switches with the N9K-X9636C-RX line card support seamless integration of EVPN (TRM) with MVPN. Other -R Series line cards can't function as the handoff node.
- The handoff node can have local (directly connected) multicast sources or receivers for the customer network.
- Any existing underlay properties, such as ASM/SSM for MVPN or ASM for TRM, are supported on the handoff node.
- The handoff node supports PIM SSM and ASM for the overlay.

- Inter-AS option A is supported on the handoff node toward the IP multicast network.
- If the number of MDT source loopback IP addresses and NVE loopback IP addresses exceeds the maximum limit, traffic drops might occur.
- The following functionality isn't supported for seamless integration of EVPN (TRM) with MVPN:
    - vPC on the handoff node
    - VXLAN ingress replication
    - SVIs and subinterfaces as core-facing interfaces for MVPN
    - Inter-AS options B and C on MVPN nodes
    - PIM SSM as a VXLAN underlay
    - Bidirectional PIM as an underlay or overlay
    - ECMP with a mix of MPLS and IP paths
- Any existing limitations for VXLAN, TRM, and MVPN also apply to seamless integration of EVPN (TRM) with MVPN.

Configuring the Handoff Node for Seamless Integration of EVPN (TRM) with MVPN

This section documents the configurations that are required on the handoff node. Configurations for other nodes (such as VXLAN leafs and spines, MVPN PE, and RS/RR) are the same as in previous releases.

PIM/IGMP Configuration for the Handoff Node

Follow these guidelines when configuring PIM/IGMP for the handoff node:

- Make sure that the Rendezvous Point (RP) is different for TRM and the MVPN underlay, as shown in the following example.

```
ip pim rp-address 90.1.1.100 group-list 225.0.0.0/8 --- TRM Underlay
ip pim rp-address 91.1.1.100 group-list 233.0.0.0/8 --- MVPN Underlay
```

- Use a common RP for overlay multicast traffic.
- The RP can be in static, PIM Anycast, or PIM MSDP mode. The following example shows the RP configuration inside the VRF:

```
vrf context vrfVxLAN5001
  vni 5001
  ip pim rp-address 111.1.1.1 group-list 226.0.0.0/8
  ip pim rp-address 112.2.1.1 group-list 227.0.0.0/8
```

- Enable IGMP snooping for VXLAN traffic using the **ip igmp snooping vxlan** command.
- Enable PIM sparse mode on all source interfaces and interfaces required to carry PIM traffic.

BGP Configuration for the Handoff Node

Follow these guidelines when configuring BGP for the handoff node:

- Add all VXLAN leafs as L2EVPN and TRM neighbors; include the redundant handoff node. If a route reflector is used, add only RR as a neighbor.
- Add all MVPN PEs as VPN neighbors. In MDT mode, add the MVPN PEs as MDT neighbors.
- Import configuration to advertise unicast routes from L2EVPN neighbors to VPN neighbors and vice versa.
- The BGP source identifier can be different or the same as the source interfaces used for the VTEP identifier (configured under the NVE interface)/MVPN PE identifier.

```
feature bgp
address-family ipv4 mdt
address-family ipv4 mvpn

neighbor 2.1.1.1
  address-family ipv4 mvpn
    send-community extended
  address-family l2vpn evpn
    send-community extended
    import vpn unicast reoriginate

neighbor 30.30.30.30
  address-family vpnv4 unicast
    send-community
    send-community extended
    next-hop-self
    import l2vpn evpn reoriginate
  address-family ipv4 mdt
    send-community extended
    no next-hop-third-party
```

- Never use Inter-AS option B between MVPN peers. Instead, configure the **no allocate-label option-b** command under the VPNv4 unicast address family.

```
address-family vpnv4 unicast
    no allocate-label option-b
```

- Set maximum paths should be set in EBGP mode.

```
address-family l2vpn evpn
    maximum-paths 8
vrf vrfVxLAN5001
    address-family ipv4 unicast
        maximum-paths 8
```

- If handoff nodes are deployed in dual mode, use the **route-map** command to avoid advertising prefixes associated with orphan hosts under the VPN address family.

```
ip prefix-list ROUTES_CONNECTED_NON_LOCAL seq 2 premit 15.14.0.15/32

route-map ROUTES_CONNECTED_NON_LOCAL deny
    match ip address prefix-list ROUTES_CONNECTED_NON_LOCAL

neighbor 8.8.8.8
    remote-as 100
```

```
    update-source loopback1
    address-family vpnv4 unicast
      send-community
      send-community extended
      route-map ROUTES_CONNECTED_NON_LOCAL out
```

VXLAN Configuration for the Handoff Node

Follow these guidelines when configuring VXLAN for the handoff node:

- Enable the following features:

```
feature nv overlay
feature ngmvpn
feature interface-vlan
feature vn-segment-vlan-based
```

- Configure the required L3 VNI:

```
L3VNIs are mapped to tenant VRF.
vlan 2501
  vn-segment 5001 <-- Associate VNI to a VLAN.
```

- Configure the NVE interface:

```
interface nve1
  no shutdown
  host-reachability protocol bgp
  source-interface loopback1 <-- This interface should not be the same as the MVPN
source interface.
  global suppress-arp
member vni 5001 associate-vrf <-- L3VNI
  mcast-group 233.1.1.1 <-- The underlay multicast group for VXLAN should be different
 from the MVPN default/data MDT.
```

- Configure the tenant VRF:

```
vrf context vrfVxLAN5001
  vni 5001 <-- Associate VNI to VRF.
  rd auto
address-family ipv4 unicast
    route-target both auto
    route-target both auto mvpn
    route-target both auto evpn

interface Vlan2501 <-- SVI interface associated with the L3VNI
  no shutdown
  mtu 9216 <-- The overlay header requires 58 byes, so the max tenant traffic is
(Configured MTU - 58).
  vrf member vrfVxLAN5001
  no ip redirects
  ip forward
  ipv6 forward
  no ipv6 redirects
  ip pim sparse-mode <-- PIM is enabled.

interface Vlan2 <-- SVI interface associated with L2 VNI
  no shutdown
  vrf member vrfVxLAN5001
```

```
 no ip redirects
 ip address 100.1.1.1/16
 no ipv6 redirects
 ip pim sparse-mode <-- PIM enabled on L2VNI
fabric forwarding mode anycast-gateway
```

MVPN Configuration for the Handoff Node

Follow these guidelines when configuring MVPN for the handoff node:

- Enable the following features:

```
install feature-set mpls
allow feature-set mpls
feature-set mpls
feature mpls l3vpn
feature mvpn
feature mpls ldp
```

- MPLS LDP Configuration:
  - Enable MPLS LDP (**mpls ip**) on all interfaces that are MPLS links.
  - Do not advertise loopback interfaces used for VXLAN as MPLS prefixes.
    - Configure a prefix list that contains IP addresses that identify the MVPN PE node.

```
ip prefix-list LDP-LOOPBACK seq 51 permit 9.1.1.10/32
ip prefix-list LDP-LOOPBACK seq 52 permit 9.1.2.10/32
```

    - Configure label allocation only for MVPN PE identifiers.

```
mpls ldp configuration
  explicit-null
  advertise-labels for LDP-LOOPBACK
  label allocate global prefix-list LDP-LOOPBACK
```

- Tenant VRF Configuration:
  - For the default MDT mode, make the underlay multicast group the same for all tenant multicast traffic under the VRF.

```
vrf context vrfVxLAN5001
  vni 5001
  mdt default 225.1.100.1
  mdt source loopback100 <-- If the source interface is not configured, the BGP
identifier is used as the source interface.
  mdt asm-use-shared-tree <-- If the underlay is configured in ASM mode
  no mdt enforce-bgp-mdt-safi <-- Enabled by befault but should be negated if BGP
MDT should not be used for discovery.
  mdt mtu <mtu-value> <-- Overlay ENCAP Max MTU value
```

  - For the data MDT mode, configure a unique multicast group-set for a subset of or all tenant multicast traffic.

```
mdt data 229.1.100.2/32 immediate-switch
mdt data 232.1.10.4/24 immediate-switch
```

```
route-map DATA_MDT_MAP permit 10
  match ip multicast group 237.1.1.1/32
mdt data 235.1.1.1/32 immediate-switch route-map DATA_MDT_MAP
```

- Enable MVPN tunnel statistics.

```
hardware profile mvpn-stats module all
```

CoPP Configuration for the Handoff Node

Both TRM and MVPN are heavily dependent on the control plane. Make sure to set the CoPP policy bandwidth as per the topology.

The following CoPP classes are used for TRM and MVPN traffic:

- **copp-system-p-class-multicast-router** (The default bandwidth is 3000 pps.)
- **copp-system-p-class-l3mc-data** (The default bandwidth is 3000 pps.)
- **copp-system-p-class-l2-default** (The default bandwidth is 50 pps.)
- **copp-class-normal-igmp** (The default bandwidth is 6000 pps.)

The following configuration example shows CoPP policies that can be configured to avoid control packet drops with multicast route scale.

Note The policer values in this example are approximations and might not be optimal for all topologies or traffic patterns. Configure the CoPP policies according to the MVPN/TRM traffic pattern.

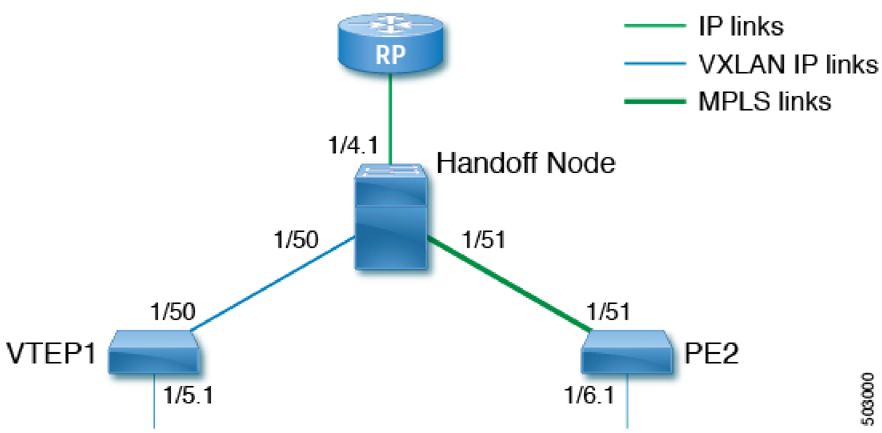
```
copp copy profile strict prefix custom
  policy-map type control-plane custom-copp-policy-strict
    class custom-copp-class-normal-igmp
      police cir 6000 pps bc 512 packets conform transmit violate drop
  control-plane
    service-policy input custom-copp-policy-strict

copp copy profile strict prefix custom
  policy-map type control-plane custom-copp-policy-strict
    class custom-copp-class-multicast-router
      police cir 6000 pps bc 512 packets conform transmit violate drop
  control-plane
    service-policy input custom-copp-policy-strict

copp copy profile strict prefix custom
  policy-map type control-plane custom-copp-policy-strict
    class copp-system-p-class-l3mc-data
      police cir 3000 pps bc 512 packets conform transmit violate drop
  control-plane
    service-policy input custom-copp-policy-strict

copp copy profile strict prefix custom
  policy-map type control-plane custom-copp-policy-strict
    class custom-copp-class-l2-default
      police cir 9000 pps bc 512 packets conform transmit violate drop
  control-plane
```

```
  service-policy input custom-copp-policy-strict
```

Configuration Example for Seamless Integration of EVPN (TRM) with MVPN

The following figure shows a sample topology with a VXLAN network on the left, an MVPN network on the right, and a centralized handoff node.

Figure 45: Sample Topology for Seamless Integration of EVPN (TRM) with MVPN



The following example show sample configurations for the VTEP, handoff node, and PE in this topology.

Configuration on VTEP1:

```
feature ngmvpn
feature interface-vlan
feature vn-segment-vlan-based
feature nv overlay
feature pim
nv overlay evpn
ip pim rp-address 90.1.1.100 group-list 225.0.0.0/8
ip pim ssm range 232.0.0.0/8

vlan 555
  vn-segment 55500

route-map ALL_ROUTES permit 10
interface nve1
  no shutdown
  host-reachability protocol bgp
  source-interface loopback2
  member vni 55500 associate-vrf
    mcast-group 225.3.3.3

interface loopback1
  ip address 196.196.196.196/32

interface loopback2
  ip address 197.197.197.197/32
  ip pim sparse-mode
```

```
feature bgp
router bgp 1
    address-family l2vpn evpn
        maximum-paths 8
        maximum-paths ibgp 8
    neighbor 2.1.1.2
        remote-as 1
        update-source loopback 1
        address-family ipv4 unicast
          send-community extended
        address-family ipv6 unicast
          send-community extended
        address-family ipv4 mvpn
          send-community extended
        address-family l2vpn evpn
          send-community extended
    vrf vrfVxLAN5023
        address-family ipv4 unicast
          advertise l2vpn evpn
          redistribute direct route-map ALL_ROUTES
          maximum-paths 8
          maximum-paths ibgp 8

vrf context vpn1
  vni 55500
  ip pim rp-address 27.27.27.27 group-list 224.0.0.0/4
  ip pim ssm range 232.0.0.0/8
  ip multicast multipath s-g-hash next-hop-based
rd auto
  address-family ipv4 unicast
    route-target both auto
    route-target both auto mvpn
    route-target both auto evpn

interface Vlan555
  no shutdown
  vrf member vpn1
  ip forward
  ip pim sparse-mode

interface Ethernet 1/50
  ip pim sparse-mode

interface Ethernet1/5.1
  encapsulation dot1q 90
  vrf member vpn1
  ip address 10.11.12.13/24
  ip pim sparse-mode
  no shutdown
```

Configuration on the handoff node:

```
install feature-set mpls
  allow feature-set mpls
feature-set mpls
feature ngmvpn
feature bgp
feature pim
feature mpls l3vpn
feature mvpn
feature mpls ldp
feature interface-vlan
feature vn-segment-vlan-based
feature nv overlay
```

```
nv overlay evpn

ip pim rp-address 90.1.1.100 group-list 225.0.0.0/8
ip pim rp-address 91.1.1.100 group-list 232.0.0.0/8

interface loopback1
  ip address 90.1.1.100 /32
  ip pim sparse-mode

interface loopback2
  ip address 91.1.1.100 /32
  ip pim sparse-mode

ip prefix-list LDP-LOOPBACK seq 2 permit 20.20.20.20/32
ip prefix-list LDP-LOOPBACK seq 3 permit 30.30.30.30/32
mpls ldp configuration
    advertise-labels for LDP-LOOPBACK
    label allocate label global prefix-list LDP-LOOPBACK

interface Ethernet 1/50
    ip pim sparse-mode

interface Ethernet 1/51
    ip pim sparse-mode
    mpls ip

interface Ethernet1/4.1
  encapsulation dot1q 50
  vrf member vpn1
  ip pim sparse-mode
  no shutdown

interface loopback0
  ip address 20.20.20.20/32
  ip pim sparse-mode

vlan 555
  vn-segment 55500

route-map ALL_ROUTES permit 10

interface nve1
  no shutdown
  host-reachability protocol bgp
  source-interface loopback3
  member vni 55500 associate-vrf
    mcast-group 225.3.3.3

interface loopback3
  ip address 198.198.198.198/32
  ip pim sparse-mode

vrf context vpn1
  vni 55500
  ip pim rp-address 27.27.27.27 group-list 224.0.0.0/4
  ip pim ssm range 232.0.0.0/8
  ip multicast multipath s-g-hash next-hop-based
  mdt default 232.1.1.1
  mdt source loopback 0
  rd auto
  address-family ipv4 unicast
    route-target both auto
    route-target both auto mvpn
    route-target both auto evpn
```

```
interface Vlan555
  no shutdown
  vrf member vpn1
  ip forward
  ip pim sparse-mode

router bgp 1
    address-family l2vpn evpn
        maximum-paths 8
        maximum-paths ibgp 8
    address-family vpnv4 unicast
        no allocate-label option-b
    address-family ipv4 mdt
    address-family ipv4 mvpn
        maximum-paths 8
        maximum-paths ibgp 8
    neighbor 196.196.196.196
        remote-as 1
        address-family ipv4 unicast
          send-community extended
        address-family ipv6 unicast
          send-community extended
        address-family ipv4 mvpn
          send-community extended
        address-family l2vpn evpn
          send-community extended
          import vpn unicast reoriginate

router bgp 1
    neighbor 30.30.30.30
        remote-as 100
        update-source loopback0
        ebgp-multihop 255
        address-family ipv4 unicast
          send-community extended
        address-family vpnv4 unicast
          send-community
          send-community extended
          next-hop-self
          import l2vpn evpn reoriginate
        address-family ipv4 mdt
          send-community extended
          no next-hop-third-party
```

Configuration on PE2:

```
install feature-set mpls
  allow feature-set mpls
feature-set mpls
feature bgp
feature pim
feature mpls l3vpn
feature mpls ldp
feature interface-vlan

ip pim rp-address 91.1.1.100 group-list 232.0.0.0/8
ip prefix-list LDP-LOOPBACK seq 2 permit 20.20.20.20/32
ip prefix-list LDP-LOOPBACK seq 3 permit 30.30.30.30/32
mpls ldp configuration
    advertise-labels for LDP-LOOPBACK
    label allocate label global prefix-list LDP-LOOPBACK

interface Ethernet 1/51
```

```
    ip pim sparse-mode
    mpls ip

interface Ethernet1/6.1
  encapsulation dot1q 50
  vrf member vpn1
  ip pim sparse-mode
  no shutdown

interface loopback0
  ip address 30.30.30.30/32
  ip pim sparse-mode

vrf context vpn1
  ip pim rp-address 27.27.27.27 group-list 224.0.0.0/4
  ip pim ssm range 232.0.0.0/8
  ip multicast multipath s-g-hash next-hop-based
  mdt default 232.1.1.1
  mdt source loopback 0
  rd auto
  address-family ipv4 unicast
    route-target both auto
    route-target both auto mvpn
    route-target both auto evpn

router bgp 100
      router-id 30.30.30.30
      address-family vpnv4 unicast
            additional-paths send
            additional-paths receive
            no allocate-label option-b
      neighbor 20.20.20.20
            remote-as 1
            update-source loopback0
            address-family vpnv4 unicast
                send-community
                send-community extended
            address-family ipv4 mdt
                send-community extended
                no next-hop-third-party
```

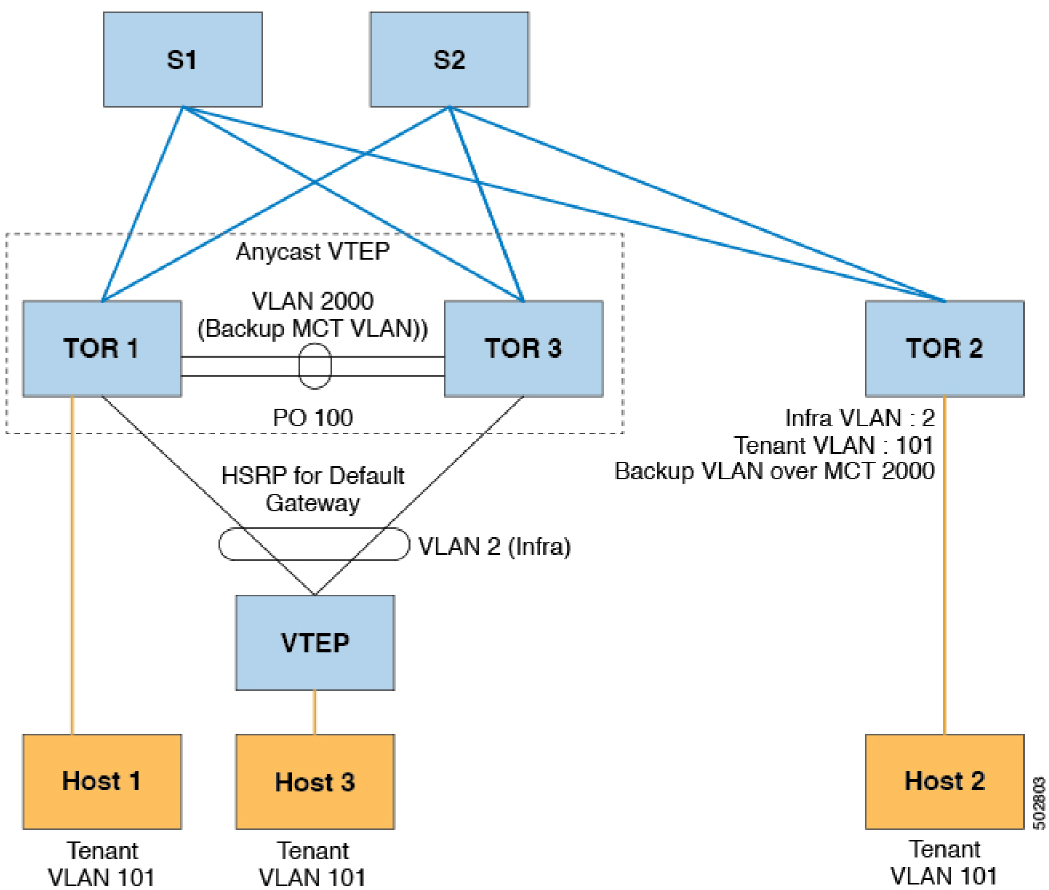
APPENDIX A

Configuring Bud Node

This chapter contains the following sections:

VXLAN Bud Node Over vPC Overview

Figure 46: Underlay Network Based on PIM-SM and OSPF



Note For bud-node topologies, the source IP of the VTEP behind vPC must be in the same subnet as the infra VLAN. This SVI should have proxy ARP enabled. For example:

```
Interface Vlan2
ip proxy-arp
```

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

For Cisco Nexus 9200, 9300-EX, and 9300-FX/FX2/FX3 and 9300-GX platform switches, use the **system nve infra-vlans** command to configure any VLANs that are used as infra-VLANs.

VXLAN Bud Node Over vPC Topology Example

- Enable the required features:

```
feature ospf
feature pim
feature interface-vlan
feature vn-segment-vlan-based
feature hsrp
feature lacp
feature vpc
feature nv overlay
```

- Configuration for PIM anycast RP.

  In this example, 1.1.1.1 is the anycast RP address.

```
ip pim rp-address 1.1.1.1 group-list 225.0.0.0/8
```

- VLAN configuration

  In this example, tenant VLANs 101-103 are mapped to vn-segments.

```
vlan 1-4,101-103,2000
vlan 101
  vn-segment 10001
vlan 102
  vn-segment 10002
vlan 103
  vn-segment 10003
```

- vPC configuration

```
vpc domain 1
  peer-switch
  peer-keepalive destination 172.31.144.213
  delay restore 180
  peer-gateway
  ipv6 nd synchronize
  ip arp synchronize
```

- Infra VLAN SVI configuration

```
interface Vlan2
  no shutdown
  no ip redirects
  ip proxy-arp
  ip address 10.200.1.252/24
  no ipv6 redirects
  ip router ospf 1 area 0.0.0.0
  ip pim sparse-mode
  ip igmp static-oif route-map match-mcast-groups
  hsrp version 2
  hsrp 1
    ip 10.200.1.254
```

- Route-maps for matching multicast groups

  Each VXLAN multicast group needs to have a static OIF on the backup SVI MCT.

```
route-map match-mcast-groups permit 1
  match ip multicast group 225.1.1.1/32
```

- Backup SVI over MCT configuration
  - Configuration Option 1:

    ```
    interface Vlan2000
    no shutdown
    ip address 20.20.20.1/24
    ip router ospf 1 area 0.0.0.0
    ip pim sparse-mode
    ```

  - Configuration Option 2:

    ```
    interface Vlan2000
    no shutdown
    ip address 20.20.20.1/24
    ip router ospf 1 area 0.0.0.0
    ip pim sparse-mode
    ```

- vPC interface configuration that carries the infra VLAN

```
interface port-channel1
  switchport mode trunk
  switchport trunk allowed vlan 2
  vpc 1
```

- MCT configuration

```
interface port-channel100
  switchport mode trunk
  spanning-tree port type network
  vpc peer-link
```

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.

NVE configuration

Option 1

```
interface nve1
  no shutdown
  source-interface loopback0
  member vni 10001 mcast-group 225.1.1.1
  member vni 10002 mcast-group 225.1.1.1
  member vni 10003 mcast-group 225.1.1.1
```

Option 2

```
interface nve1
  no shutdown
  source-interface loopback0
  global mcast-group 225.1.1.1
  member vni 10001
  member vni 10002
  member vni 10003
```

- Loopback interface configuration

```
interface loopback0
  ip address 101.101.101.101/32
  ip address 99.99.99.99/32 secondary
  ip router ospf 1 area 0.0.0.0
  ip pim sparse-mode
```

- Show commands

```
tor1# sh 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      10001    225.1.1.1         Up    DP   L2 [101]
nve1      10002    225.1.1.1         Up    DP   L2 [102]
nve1      10003    225.1.1.1         Up    DP   L2 [103]

tor1# sh nve peers
Interface Peer-IP          State LearnType Uptime   Router-Mac
--------- ---------------  ----- --------- -------- -----------------
nve1      10.200.1.1       Up    DP        00:07:23 n/a
nve1      10.200.1.2       Up    DP        00:07:18 n/a
nve1      102.102.102.102  Up    DP        00:07:23 n/a


tor1# sh ip mroute 225.1.1.1
IP Multicast Routing Table for VRF "default"

(*, 225.1.1.1/32), uptime: 00:07:41, ip pim nve static igmp
  Incoming interface: Ethernet2/1, RPF nbr: 10.1.5.2
  Outgoing interface list: (count: 3)
    Vlan2, uptime: 00:07:23, igmp
    Vlan2000, uptime: 00:07:31, static
    nve1, uptime: 00:07:41, nve

(10.200.1.1/32, 225.1.1.1/32), uptime: 00:07:40, ip mrib pim nve
  Incoming interface: Vlan2, RPF nbr: 10.200.1.1
  Outgoing interface list: (count: 3)
    Vlan2, uptime: 00:07:23, mrib, (RPF)
    Vlan2000, uptime: 00:07:31, mrib
    nve1, uptime: 00:07:40, nve

(10.200.1.2/32, 225.1.1.1/32), uptime: 00:07:41, ip mrib pim nve
  Incoming interface: Vlan2, RPF nbr: 10.200.1.2
  Outgoing interface list: (count: 3)
    Vlan2, uptime: 00:07:23, mrib, (RPF)
    Vlan2000, uptime: 00:07:31, mrib
    nve1, uptime: 00:07:41, nve

(99.99.99.99/32, 225.1.1.1/32), uptime: 00:07:41, ip mrib pim nve
  Incoming interface: loopback0, RPF nbr: 99.99.99.99
  Outgoing interface list: (count: 3)
    Vlan2, uptime: 00:07:23, mrib
    Vlan2000, uptime: 00:07:31, mrib
    Ethernet2/5, uptime: 00:07:39, pim
(102.102.102.102/32, 225.1.1.1/32), uptime: 00:07:40, ip mrib pim nve
  Incoming interface: Ethernet2/1, RPF nbr: 10.1.5.2
  Outgoing interface list: (count: 1)
    nve1, uptime: 00:07:40, nve


tor1# sh vpc
Legend:
                 - local vPC is down, forwarding via vPC peer-link

vPC domain id                     : 1
Peer status                       : peer adjacency formed ok
vPC keep-alive status             : peer is alive
Configuration consistency status  : success
Per-vlan consistency status       : success
Type-2 consistency status         : success
```

```
vPC role                          : secondary, operational primary
Number of vPCs configured         : 4
Peer Gateway                      : Enabled
Dual-active excluded VLANs        : -
Graceful Consistency Check        : Enabled
Auto-recovery status              : Disabled
Delay-restore status              : Timer is off.(timeout = 180s)
Delay-restore SVI status          : Timer is off.(timeout = 10s)

vPC Peer-link status
---------------------------------------------------------------------
id   Port   Status Active vlans
--   ----   ------ -------------------------------------------------
1    Po100  up     1-4,101-103,2000

vPC status
----------------------------------------------------------------------
id   Port   Status Consistency Reason                     Active vlans
--   ----   ------ ----------- ------                     ------------
1    Po1    up     success     success                    2

2    Po2    up     success     success                    2

tor1# sh vpc consistency-parameters global

    Legend:
        Type 1 : vPC will be suspended in case of mismatch

Name                        Type  Local Value            Peer Value
-------------               ----  ---------------------- -----------------------
Vlan to Vn-segment Map      1     3 Relevant Map(s)      3 Relevant Map(s)
STP Mode                    1     Rapid-PVST             Rapid-PVST
STP Disabled                1     None                   None
STP MST Region Name         1     ""                     ""
STP MST Region Revision     1     0                      0
STP MST Region Instance to  1
 VLAN Mapping
STP Loopguard               1     Disabled               Disabled
STP Bridge Assurance        1     Enabled                Enabled
STP Port Type, Edge         1     Normal, Disabled,      Normal, Disabled,
BPDUFilter, Edge BPDUGuard        Disabled               Disabled
STP MST Simulate PVST       1     Enabled                Enabled
Nve Oper State, Secondary   1     Up, 99.99.99.99, DP    Up, 99.99.99.99, DP
IP, Host Reach Mode
Nve Vni Configuration       1     10001-10003            10001-10003
Interface-vlan admin up     2     2,2000                 2,2000
Interface-vlan routing      2     1-4,2000               1-4,2000
capability
Allowed VLANs               -     1-4,101-103,2000       1-4,101-103,2000
Local suspended VLANs       -     -
```

APPENDIX **B**

DHCP Relay in VXLAN BGP EVPN

This chapter contains the following sections:

DHCP Relay in VXLAN BGP EVPN Overview

DHCP relay is utilized to forward DHCP packets between the hosts and DHCP server. The VXLAN VTEP can act as a relay agent, providing DHCP relay services in a multi-tenant VXLAN environment.

With DHCP Relay, DHCP messages require to be sent through the same Switch in both directions. GiAddr (Gateway IP Address) for DHCP Relay is commonly used for Scope Selection and DHCP response messages. In any VXLAN fabric with Distributed IP Anycast Gateway, DHCP messages can be returned to ANY Switch hosting the respective Gateway IP Address (GiAddr).

Solution requires a different way of Scope Selection and Unique IP Address for each Switch. Unique Loopback Interface per Switch will become GiAddr for responding to correct Switch. Option 82 (dhcp option vpn) will be used for Scope Selection based on L2VNI.

In a multi-tenant EVPN environment, DHCP relay uses the following sub-options of Option 82:

- Sub-option 151(0x97) - Virtual Subnet Selection (Defined in RFC#6607)

  Used to convey VRF related information to the DHCP server in an MPLS-VPN and VXLAN EVPN multi-tenant environment.

- Sub-option 11(0xb) - Server ID Override (Defined in RFC#5107)

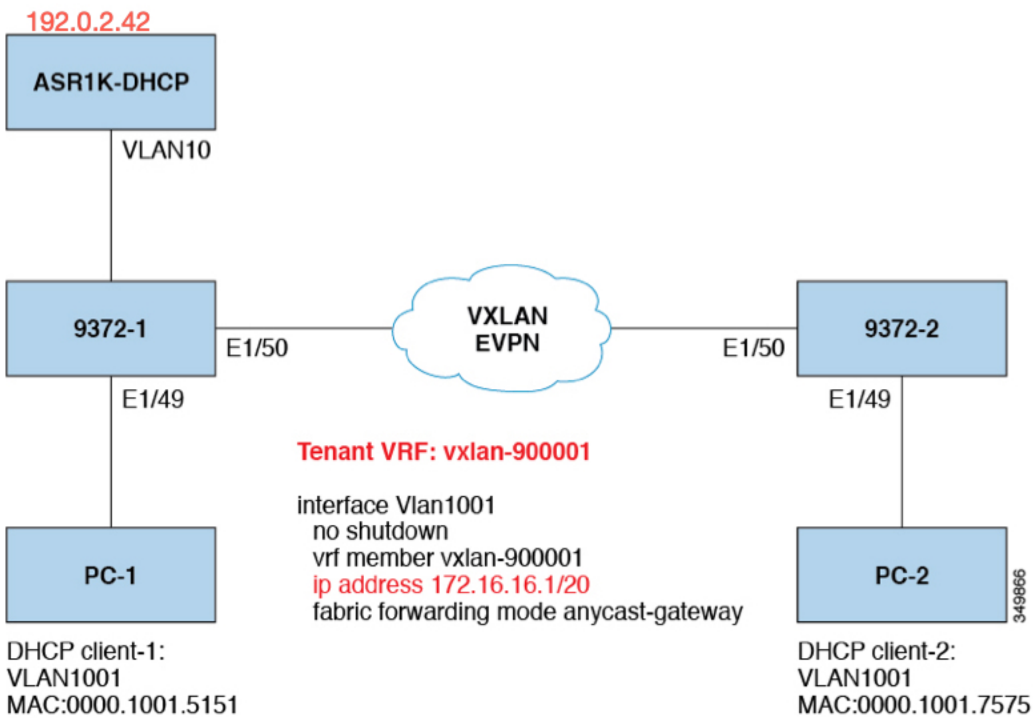
  The server identifier (server ID) override sub-option allows the DHCP relay agent to specify a new value for the server ID option, which is inserted by the DHCP server in the reply packet. This sub-option allows the DHCP relay agent to act as the actual DHCP server such that the renew requests will come to the relay agent rather than the DHCP server directly. The server ID override sub-option contains the incoming interface IP address, which is the IP address on the relay agent that is accessible from the client. Using this information, the DHCP client sends all renew and release request packets to the relay agent. The relay agent adds all of the appropriate sub-options and then forwards the renew and release request packets to the original DHCP server. For this function, Cisco's proprietary implementation is sub-option 152(0x98). You can use the **ip dhcp relay sub-option type cisco** command to manage the function.

- Sub-option 5(0x5) - Link Selection (Defined in RFC#3527)

  The link selection sub-option provides a mechanism to separate the subnet/link on which the DHCP client resides from the gateway address (giaddr), which can be used to communicate with the relay agent by the DHCP server. The relay agent will set the sub-option to the correct subscriber subnet and the DHCP server will use that value to assign an IP address rather than the giaddr value. The relay agent will set the giaddr to its own IP address so that DHCP messages are able to be forwarded over the network. For this function, Cisco's proprietary implementation is sub-option 150(0x96). You can use the **ip dhcp relay sub-option type cisco** command to manage the function.

DHCP Relay in VXLAN BGP EVPN Example

Figure 47: Example Topology



Topology characteristics:

- Switches 9372-1 and 9372-2 are VTEPs connected to the VXLAN fabric.
- Client1 and client2 are DHCP clients in vlan1001. They belong to tenant VRF vxlan-900001.
- The DHCP server is ASR1K, a router that sits in vlan10.
- DHCP server configuration

```
ip vrf vxlan900001
ip dhcp excluded-address vrf vxlan900001 172.16.16.1 172.16.16.9
ip dhcp pool one
 vrf vxlan900001
 network 172.16.16.0 255.240.0.0
 defaultrouter 172.16.16.1
```

DHCP Relay on VTEPs

The following are common deployment scenarios:

- Client on tenant VRF and server on Layer 3 default VRF.
- Client on tenant VRF (SVI X) and server on the same tenant VRF (SVI Y).
- Client on tenant VRF (VRF X) and server on different tenant VRF (VRF Y).
- Client on tenant VRF and server on non-default non-VXLAN VRF.

The following sections below move vlan10 to different VRFs to depict different scenarios.

Client on Tenant VRF and Server on Layer 3 Default VRF

Put DHCP server (192.0.2.42) into the default VRF and make sure it is reachable from both 9372-1 and 9372-2 through the default VRF.

```
9372-1# sh run int vl 10

!Command: show running-config interface Vlan10
!Time: Mon Aug 24 07:51:16 2018

version 7.0(3)I1(3)

interface Vlan10
  no shutdown
  ip address 192.0.2.25/24
  ip router ospf 1 area 0.0.0.0

9372-1# ping 192.0.2.42 cou 1

PING 192.0.2.42 (192.0.2.42): 56 data bytes
64 bytes from 192.0.2.42: icmp_seq=0 ttl=254 time=0.593 ms
- 192.0.2.42 ping statistics -
1 packets transmitted, 1 packets received, 0.00% packet loss
roundtrip min/avg/max = 0.593/0.592/0.593 ms

9372-2# ping 192.0.2.42 cou 1
PING 192.0.2.42 (192.0.2.42): 56 data bytes
64 bytes from 192.0.2.42: icmp_seq=0 ttl=252 time=0.609 ms
- 192.0.2.42 ping statistics -
1 packets transmitted, 1 packets received, 0.00% packet loss
round-trip min/avg/max = 0.609/0.608/0.609 ms
```

DHCP Relay Configuration

- 9372-1

```
9372—1# sh run dhcp

!Command: show running—config dhcp
!Time: Mon Aug 24 08:26:00 2018

version 7.0(3) I1(3)
feature dhcp

service dhcp
ip dhcp relay
ip dhcp relay information option
ip dhcp relay information option vpn
ipv6 dhcp relay

interface Vlanl001
  ip dhcp relay address 192.0.2.42 use—vrf default
```

- 9372-2

```
9372-2# sh run dhcp

!Command: show running—config dhcp
!Time: Mon Aug 24 08:26:16 2018

version 7.0(3)11(3)
feature dhcp

service dhcp
ip dhcp relay
ip dhcp relay information option
ip dhcp relay information option vpn
ipv6 dhcp relay

interfaoe Vlanl001
  ip dhcp relay address 192.0.2.42 use—vrf default
```

Debug Output

- The following is a packet dump for DHCP interact sequences.

```
9372-1# ethanalyzer local interface inband display-filter
"udp.srcport==67 or udp.dstport==67" limit-captured frames 0

Capturing on inband
20180824 08:35:25.066530 0.0.0.0 -> 255.255.255.0 DHCP DHCP Discover - Transaction ID
0x636a38fd
20180824 08:35:25.068141 192.0.2.25 -> 192.0.2.42 DHCP DHCP Discover - Transaction ID
0x636a38fd
20180824 08:35:27.069494 192.0.2.42 -> 192.0.2.25 DHCP DHCP Offer Transaction - ID
0x636a38fd
20180824 08:35:27.071029 172.16.16.1 -> 172.16.16.11 DHCP DHCP Offer Transaction - ID
0x636a38fd
20180824 08:35:27.071488 0.0.0.0 -> 255.255.255.0 DHCP DHCP Request Transaction - ID
```

```
0x636a38fd
20180824 08:35:27.072447 192.0.2.25 -> 192.0.2.42 DHCP DHCP Request Transaction - ID
0x636a38fd
20180824 08:35:27.073008 192.0.2.42 -> 192.0.2.25 DHCP DHCP ACK Transaction - ID
0x636a38fd
20180824 08:35:27.073692 172.16.16.1 -> 172.16.16.11 DHCP DHCP ACK Transaction - ID
0x636a38fd
```

Note Ethanalyzer might not capture all DHCP packets because of inband interpretation issues when you use the filter. You can avoid this by using SPAN.

- DHCP Discover packet 9372-1 sent to DHCP server.

  giaddr is set to 192.0.2.25 (ip address of vlan10) and suboptions 5/11/151 are set accordingly.

```
Bootp flags: 0x0000 (unicast)
client IP address: 0.0.0.0 (0.0.0.0)
Your (client) IP address: 0.0.0.0 (0.0.0.0)
Next server IP address: 0.0.0.0 (0.0.0.0)
Relay agent IP address: 192.0.2.25 (192.0.2.25)
client MAC address Hughes_01:51:51 (00:00:10:01:51:51)
client hardware address padding: 00000000000000000000
Server host name not given
Boot file name not given
Magic cookie: DHCP
Option: (53) DHCP Message Type
  Length: 1
  DHCP: Discover (1)
Option: (55) Parameter Request List
  Length: 4
  Parameter Request List Item: (1) Subnet Mask
  Parameter Request List Item: (3) Router
  Parameter Request List Item: (58) Renewal Time Value
  Parameter Request List Item: (59) Rebinding Time Value
Option: (61) client identifier
  Length: 7
  Hardware type: Ethernet (0x01)
  Client MAC address: Hughes_01:51:51 (00:00:10:01:51:51)
Option: (82) Agent Information Option
  Length: 47
Option 82 Suboption: (1) Agent Circuit ID
  Length: 10
  Agent Circuit ID: 01080006001e88690030
Option 82 Suboption: (2) Agent Remote ID
  Length: 6
  Agent Remote ID: f8c2882333a5
Option 82 Suboption: (151) VRF name/VPN ID
Option 82 Suboption: (11) Server ID Override
  Length: 4
  Server ID Override: 172.16.16.1 (172.16.16.1)
Option 82 Suboption: (5) Link selection
  Length: 4
  Link selection: 172.16.16.0 (172.16.16.0)
```

```
ASR1K-DHCP# sh ip dhcp bin
```

```
Bindings from all pools not associated with VRF:
IP address ClientID/ Lease expiration Type State Interface
         Hardware address/
         User name

Bindings from VRF pool vxlan900001:
IP address ClientID/ Lease expiration Type State Interface
         Hardware address/
         User name
172.16.16.10 0100.0010.0175.75 Aug 25 2018 09:21 AM Automatic Active GigabitEthernet2/1/0
172.16.16.11 0100.0010.0151.51 Aug 25 2018 08:54 AM Automatic Active GigabitEthernet2/1/0

9372-1# sh ip route vrf vxlan900001
IP Route Table for VRF "vxlan900001"
'*' denotes best ucast nexthop
'**' denotes best mcast nexthop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

10.11.11.11/8, ubest/mbest: 2/0, attached
  *via 10.11.11.11, Lo1, [0/0], 18:31:57, local
  *via 10.11.11.11, Lo1, [0/0], 18:31:57, direct
10.22.22.22/8, ubest/mbest: 1/0
  *via 1.2.2.2%default, [200/0], 18:31:57, bgp65535,internal, tag 65535 (evpn)segid:
900001 tunnelid: 0x2020202
encap: VXLAN

172.16.16.0/20, ubest/mbest: 1/0, attached
 *via 172.16.16.1, Vlan1001, [0/0], 18:31:57, direct
172.16.16.1/32, ubest/mbest: 1/0, attached
 *via 172.16.16.1, Vlan1001, [0/0], 18:31:57, local
172.16.16.10/32, ubest/mbest: 1/0
 *via 1.2.2.2%default, [200/0], 00:00:47, bgp65535,internal, tag 65535 (evpn)segid:
900001 tunnelid: 0x2020202
encap: VXLAN

172.16.16.11/32, ubest/mbest: 1/0, attached
 *via 172.16.16.11, Vlan1001, [190/0], 00:28:10, hmm

9372-1# ping 172.16.16.11 vrf vxlan900001 count 1
PING 172.16.16.11 (172.16.16.11): 56 data bytes
64 bytes from 172.16.16.11: icmp_seq=0 ttl=63 time=0.846 ms
- 172.16.16.11 ping statistics -
1 packets transmitted, 1 packets received, 0.00% packet loss
round-trip min/avg/max = 0.846/0.845/0.846 ms

9372-1# ping 172.16.16.10 vrf vxlan900001 count 1
PING 172.16.16.10 (172.16.16.10): 56 data bytes
64 bytes from 172.16.16.10: icmp_seq=0 ttl=62 time=0.874 ms
- 172.16.16.10 ping statistics -
1 packets transmitted, 1 packets received, 0.00% packet loss
round-trip min/avg/max = 0.874/0.873/0.874 ms
```

Client on Tenant VRF (SVI X) and Server on the Same Tenant VRF (SVI Y)

Put DHCP server (192.0.2.42) into VRF of vxlan-900001 and make sure it is reachable from both 9372-1 and 9372-2 through VRF of vxlan-900001.

```
9372-1# sh run int vl 10

!Command: show running-config interface Vlan10
!Time: Mon Aug 24 09:10:26 2018

version 7.0(3)I1(3)

interface Vlan10
  no shutdown
  vrf member vxlan-900001
  ip address 192.0.2.25/24
```

Because 172.16.16.1 is an anycast address for vlan1001 configured on all the VTEPs, we need to pick up a unique address as the DHCP relay packet's source address to make sure the DHCP server can deliver a response to the original DHCP Relay agent. In this scenario, we use loopback1 and we need to make sure loopback1 is reachable from everywhere of VRF vxlan-900001.

```
9372-1# sh run int lo1

!Command: show running-config interface loopback1
!Time: Mon Aug 24 09:18:53 2018

version 7.0(3)I1(3)

interface loopback1
  vrf member vxlan-900001
  ip address 10.11.11.11/8

9372-1# ping 192.0.2.42 vrf vxlan900001 source 10.11.11.11 cou 1
PING 192.0.2.42 (192.0.2.42) from 10.11.11.11: 56 data bytes
64 bytes from 192.0.2.42: icmp_seq=0 ttl=254 time=0.575 ms
- 192.0.2.42 ping statistics -
1 packets transmitted, 1 packets received, 0.00% packet loss
round-trip min/avg/max = 0.575/0.574/0.575 ms

9372-2# sh run int lo1

!Command: show running-config interface loopback1
!Time: Mon Aug 24 09:19:30 2018

version 7.0(3)I1(3)

interface loopback1
  vrf member vxlan900001
  ip address 10.22.22.22/8

9372-2# ping 192.0.2.42 vrf vxlan-900001 source 10.22.22.22 cou 1
PING 192.0.2.42 (192.0.2.42) from 10.22.22.22: 56 data bytes
64 bytes from 192.0.2.42: icmp_seq=0 ttl=253 time=0.662 ms
- 192.0.2.42 ping statistics -
1 packets transmitted, 1 packets received, 0.00% packet loss
round-trip min/avg/max = 0.662/0.662/0.662 ms
```

DHCP Relay Configuration

- 9372-1

```
9372—1# sh run dhcp

!Command: show running-config dhcp
!Time: Mon Aug 24 08:26:00 2018

version 7.0(3)11(3)
feature dhcp

service dhcp
ip dhcp relay
ip dhcp relay information option
I4ip dhcp relay information option vpn
ipv6 dhcp relay

interface Vlanl00l
  ip dhcp relay address 192.0.2.42
  ip dhcp relay source—interface loopback1
```

- 9372-2

```
9372—2# sh run dhcp

!Command: show running-config dhcp
!Time: Mon Aug 24 08:26:16 2018

version 7.0(3) 11(3)
feature dhcp

service dhcp
ip dhcp relay
ip dhcp relay information option
ip dhcp relay information option vpn
ipv6 dhcp relay

interface Vlanl00l
  ip dhcp relay address 192.0.2.42
  ip dhcp relay source—interface loopback1
```

Debug Output

- The following is a packet dump for DHCP interact sequences.

```
9372-1# ethanalyzer local interface inband display-filter
"udp.srcport==67 or udp.dstport==67" limit-captured frames 0

Capturing on inband
20180824 09:31:38.129393 0.0.0.0 -> 255.255.255.0 DHCP DHCP Discover - Transaction ID
0x860cd13
20180824 09:31:38.129952 10.11.11.11 -> 192.0.2.42 DHCP DHCP Discover - Transaction ID
 0x860cd13
20180824 09:31:40.130134 192.0.2.42 -> 10.11.11.11 DHCP DHCP Offer - Transaction ID
0x860cd13
20180824 09:31:40.130552 172.16.16.1 -> 172.16.16.11 DHCP DHCP Offer - Transaction ID
0x860cd13
20180824 09:31:40.130990 0.0.0.0 -> 255.255.255.0 DHCP DHCP Request - Transaction ID
0x860cd13
20180824 09:31:40.131457 10.11.11.11 -> 192.0.2.42 DHCP DHCP Request - Transaction ID
0x860cd13
20180824 09:31:40.132009 192.0.2.42 -> 10.11.11.11 DHCP DHCP ACK - Transaction ID
```

```
0x860cd13
20180824 09:31:40.132268 172.16.16.1 -> 172.16.16.11 DHCP DHCP ACK - TransactionID
0x860cd13
```

Note Ethanalyzer might not capture all DHCP packets because of inband interpretation issues when you use the filter. You can avoid this by using SPAN.

- DHCP Discover packet 9372-1 sent to DHCP server.

  giaddr is set to 10.11.11.11(loopback1) and suboptions 5/11/151 are set accordingly.

```
Bootstrap Protocol
  Message type: Boot Request (1)
  Hardware type: Ethernet (0x01)
  Hardware address length: 6
  Hops: 1
  Transaction ID: 0x0860cd13
  Seconds elapsed: O
  Bootp flags: 0x0000 (unicast)
  Client IP address: 0.0.0.0 (0.0.0.0)
  Your (client) IP address: 0.0.0.0 (0.0.0.0)
  Next server IP address: 0.0.0.0 (0.0.0.0)
  Relay agent iP address: 10.11.11.11 (10.11.11.11)
  Client MAC address: Hughes_01:51:51 (00:00:10:01:51:51)
  Client hardware address padding: 00000000000000000000
  Server host name not given
  Boot file name not given
  Magic cookie: DHCP
  Option: (53) DHCP Message Type
    Length: 1
    DHCP: Discover (1)
  Option: (55) Parameter Request List
  Option: (61) Client Identifier
  Option: (82) Agent Information Option
    Length: 47
  Option 82 suboption: (1) Aqent Circuit ID
  Option 82 suboption: (151) Agent Remote ID
  Option 82 suboption: (11) Server ID Override
    Length: 4
    Server ID override: 172.16.16.1 (172.16.16.1)
  Option 82 suboption: (5) Link selection
    Length: 4
    Link selection: 172.16.16.0 (172.16.16.0)
```

```
ASR1K-DHCP# sh ip dhcp bin
Bindings from all pools not associated with VRF:
IP address ClientID/Lease expiration Type State Interface
         Hardware address/
         User name

Bindings from VRF pool vxlan-900001:
IP address ClientID/Lease expiration Type State Interface
         Hardware address/
         User name
```

```
172.16.16.10 0100.0010.0175.75 Aug 25 2018 10:02 AM Automatic Active GigabitEthernet2/1/0
172.16.16.11 0100.0010.0151.51 Aug 25 2018 09:50 AM Automatic Active GigabitEthernet2/1/0

9372-1# sh ip route vrf vxlan-900001
IP Route Table for VRF "vxlan-900001"
'*' denotes best ucast nexthop
'**' denotes best mcast nexthop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

10.11.11.11/8, ubest/mbest: 2/0, attached
  *via 10.11.11.11, Lo1, [0/0], 19:13:56, local
  *via 10.11.11.11, Lo1, [0/0], 19:13:56, direct
10.22.22.22/8, ubest/mbest: 1/0
  *via 2.2.2.2%default, [200/0], 19:13:56, bgp65535,internal, tag 65535 (evpn)segid:
900001 tunnelid: 0x2020202
encap: VXLAN
172.16.16.0/20, ubest/mbest: 1/0, attached
  *via 172.16.16.1, Vlan1001, [0/0], 19:13:56, direct
172.16.16.1/32, ubest/mbest: 1/0, attached
  *via 172.16.16.1, Vlan1001, [0/0], 19:13:56, local
172.16.16.10/32, ubest/mbest: 1/0
  *via 2.2.2.2%default, [200/0], 00:01:27, bgp65535,
internal, tag 65535 (evpn)segid: 900001 tunnelid: 0x2020202
encap: VXLAN
172.16.16.11/32, ubest/mbest: 1/0, attached
  *via 172.16.16.11, Vlan1001, [190/0], 00:13:56, hmm
192.0.2.20/24, ubest/mbest: 1/0, attached
  *via 192.0.2.25, Vlan10, [0/0], 00:36:08, direct
192.0.2.25/24, ubest/mbest: 1/0, attached
  *via 192.0.2.25, Vlan10, [0/0], 00:36:08, local
9372-1# ping 172.16.16.10 vrf vxlan-900001 cou 1
PING 172.16.16.10 (172.16.16.10): 56 data bytes
64 bytes from 172.16.16.10: icmp_seq=0 ttl=62 time=0.808 ms
- 172.16.16.10 ping statistics -
1 packets transmitted, 1 packets received, 0.00% packet loss
round-trip min/avg/max = 0.808/0.808/0.808 ms

9372-1# ping 172.16.16.11 vrf vxlan-900001 cou 1
PING 172.16.16.11 (172.16.16.11): 56 data bytes
64 bytes from 172.16.16.11: icmp_seq=0 ttl=63 time=0.872 ms
- 172.16.16.11 ping statistics -
1 packets transmitted, 1 packets received, 0.00% packet loss
round-trip min/avg/max = 0.872/0.871/0.872 ms
```

Client on Tenant VRF (VRF X) and Server on Different Tenant VRF (VRF Y)

The DHCP server is placed into another tenant VRF vxlan-900002 so that DHCP response packets can access the original relay agent. We use loopback2 to avoid any anycast ip address that is used as the source address for the DHCP relay packets.

```
9372-1# sh run int vl 10
!Command: show runningconfig interface Vlan10
!Time: Tue Aug 25 08:48:22 2018
```

```
version 7.0(3)I1(3)
interface Vlan10
  no shutdown
  vrf member vxlan900002
  ip address 192.0.2.40/24

9372-1# sh run int lo2
!Command: show runningconfig interface loopback2
!Time: Tue Aug 25 08:48:57 2018
version 7.0(3)I1(3)
interface loopback2
  vrf member vxlan900002
  ip address 10.33.33.33/8

9372-2# sh run int lo2
!Command: show runningconfig interface loopback2
!Time: Tue Aug 25 08:48:44 2018
version 7.0(3)I1(3)
interface loopback2
  vrf member vxlan900002
  ip address 10.44.44.44/8

9372-1# ping 192.0.2.42 vrf vxlan-900002 source 10.33.33.33 cou 1
PING 192.0.2.42 (192.0.2.42) from 10.33.33.33: 56 data bytes
64 bytes from 192.0.2.42: icmp_seq=0 ttl=254 time=0.544 ms
- 192.0.2.42 ping statistics -
1 packets transmitted, 1 packets received, 0.00% packet loss
round-trip min/avg/max = 0.544/0.544/0.544 ms

9372-2# ping 192.0.2.42 vrf vxlan-900002 source 10.44.44.44 count 1
PING 192.0.2.42 (192.0.2.42) from 10.44.44.44: 56 data bytes
64 bytes from 192.0.2.42: icmp_seq=0 ttl=253 time=0.678 ms
- 192.0.2.42 ping statistics -
1 packets transmitted, 1 packets received, 0.00% packet loss
round-trip min/avg/max = 0.678/0.678/0.678 ms
```

DHCP Relay Configuration

- 9372-1

```
9372—1# sh run dhcp

!Command: show running—config dhcp
!Time: Mon Aug 24 08:26:00 2018

version 7.0(3) Ii (3)
feature dhcp

service dhcp
ip dhcp relay
ip dhcp relay information option
ip dhcp relay information option vpn
ipv6 dhcp relay

interface Vlan1OOl
  ip dhcp relay address 192.0.2.42 use—vrf vxlan—900002
  ip dhcp relay source—interface loopback2
```

- 9372-2

```
!Command: show running-config dhcp
!Time: Mon Aug 24 08:26:16 2018

version 7.0(3)11(3)
feature dhcp

service dhcp
ip dhcp relay
ip dhcp relay information option
ip dhcp relay information option vpn
ipv6 dhcp relay

interface VlaniOOl
  ip dhcp relay address 192.0.2.42 use-vrf vxlan—900002
  ip dhcp relay source—interface loopback2
```

Debug Output

- The following is a packet dump for DHCP interact sequences.

```
9372-1# ethanalyzer local interface inband display-filter "udp.srcport==67 or
udp.dstport==67" limit-captured-frames 0
Capturing on inband
20180825 08:59:35.758314 0.0.0.0 -> 255.255.255.0 DHCP DHCP Discover - Transaction ID
0x3eebccae
20180825 08:59:35.758878 10.33.33.33 -> 192.0.2.42 DHCP DHCP Discover - Transaction ID
 0x3eebccae
20180825 08:59:37.759560 192.0.2.42 -> 10.33.33.33 DHCP DHCP Offer - Transaction ID
0x3eebccae
20180825 08:59:37.759905 172.16.16.1 -> 172.16.16.11 DHCP DHCP Offer - Transaction ID
0x3eebccae
20180825 08:59:37.760313 0.0.0.0 -> 255.255.255.0 DHCP DHCP Request - Transaction ID
0x3eebccae
20180825 08:59:37.760733 10.33.33.33 -> 192.0.2.42 DHCP DHCP Request - Transaction ID
0x3eebccae
20180825 08:59:37.761297 192.0.2.42 -> 10.33.33.33 DHCP DHCP ACK - Transaction ID
0x3eebccae
20180825 08:59:37.761554 172.16.16.1 -> 172.16.16.11 DHCP DHCP ACK - Transaction ID
0x3eebccae
```

- DHCP Discover packet 9372-1 sent to DHCP server.

  giaddr is set to 10.33.33.33 (loopback2) and suboptions 5/11/151 are set accordingly.

```
Bootstrap Protocol
  Message type: Boot Request (1)
  Hardware type: Ethernet (0x01)
  Hardware address length: 6
  Hops: 1
  Transaction ID: Ox3eebccae
  Seconds elapsed: O
  Bootp flags: 0x0000 (unicast)
  Client IP address: 0.0.0.0 (0.0.0.0)
  Your (client) IP address: 0.0.0.0 (0.0.0.0)
  Next server IP address: 0.0.0.0 (0.0.0.0)
  Relay agent IP address: 10.33.33.33 (10.33.33.33)
  Client MAC address: i-iughes_01:51:51 (00:00:10:01:51:51)
```

```
Client hardware address padding: 00000000000000000000
Server host name not given
Boot file name not given
Magic cookie: DHCP
Option: (53) DHCP Message Type
  Length: 1
  DHCP: Discover (1)
Option: (55) Parameter Request List
Option: (61) client identifier
Option: (82) Agent Information option
  Length: 47
Option 82 Suboption: (1) Agent circuit W
Option 82 suboption: (2) Agent Remote 10
Option 82 suboption: (151) VRF name/VPN ID
Option 82 Suboption: (11) Server ID Override
  Length: 4
  Server ID Override: 172.16.16.1 (172.16.16.1)
Option 82 Suboption: (5) Link selection
  Length: 4
  Link selection: 172.16.16.0 (172.16.16.0)
```

Client on Tenant VRF and Server on Non-Default Non-VXLAN VRF

The DHCP server is placed into the management VRF and is reachable the through M0 interface. The IP address changes to 10.122.164.147 accordingly.

```
9372-1# sh run int m0
!Command: show running-config interface mgmt0
!Time: Tue Aug 25 09:17:04 2018
version 7.0(3)I1(3)
interface mgmt0
  vrf member management
  ip address 10.122.165.134/8

9372-1# ping 10.122.164.147 vrf management cou 1
PING 10.122.164.147 (10.122.164.147): 56 data bytes
64 bytes from 10.122.164.147: icmp_seq=0 ttl=251 time=1.024 ms
- 10.122.164.147 ping statistics -
1 packets transmitted, 1 packets received, 0.00% packet loss
round-trip min/avg/max = 1.024/1.024/1.024 ms

9372-2# sh run int m0
!Command: show running-config interface mgmt0
!Time: Tue Aug 25 09:17:47 2018
version 7.0(3)I1(3)
interface mgmt0
  vrf member management
  ip address 10.122.165.148/8

9372-2# ping 10.122.164.147 vrf management cou 1
PING 10.122.164.147 (10.122.164.147): 56 data bytes
64 bytes from 10.122.164.147: icmp_seq=0 ttl=251 time=1.03 ms
- 10.122.164.147 ping statistics -
1 packets transmitted, 1 packets received, 0.00% packet loss
```

```
round-trip min/avg/max = 1.03/1.03/1.03 ms
```

DHCP Relay Configuration

- 9372-1

```
9372—1# sh run dhcp 9372—2# sh run dhcp

!Command: show running-config dhcp
!Time: Mon Aug 24 08:26:00 2018

version 7.0(3)11(3)
feature dhcp

service dhcp
ip dhcp relay
ip dhcp relay information option
ip dhcp relay information option vpn
ipv6 dhcp relay

interface VlanlOOl
  ip dhcp relay address 10.122.164.147 use—vrf management
```

- 9372-2

```
9372-2# sh run dhcp
!Command: show running-config dhcp
!Time: Tue Aug 25 09:17:47 2018

version 7.0(3)11(3)
feature dhcp

service dhcp
ip dhcp relay
ip dhcp relay information option
ip dhcp relay information option vpn
ipv6 dhcp relay

interface VlanlOOl
ip dhcp relay address 10.122.164.147 use—vrf management
```

Debug Output

- The following is a packet dump for DHCP interact sequences.

```
9372-1# ethanalyzer local interface inband display-filter "udp.srcport==67 or
udp.dstport==67" limit-captured-frames 0
Capturing on inband
20180825 09:30:54.214998 0.0.0.0 -> 255.255.255.0 DHCP DHCP Discover - Transaction ID
0x28a8606d
20180825 09:30:56.216491 172.16.16.1 -> 172.16.16.11 DHCP DHCP Offer - Transaction ID
0x28a8606d
20180825 09:30:56.216931 0.0.0.0 -> 255.255.255.0 DHCP DHCP Request - Transaction ID
0x28a8606d
20180825 09:30:56.218426 172.16.16.1 -> 172.16.16.11 DHCP DHCP ACK - Transaction ID
0x28a8606d
```

```
9372-1# ethanalyzer local interface mgmt display-filter "ip.src==10.122.164.147 or
ip.dst==10.122.164.147" limit-captured-frames 0
Capturing on mgmt0
20180825 09:30:54.215499 10.122.165.134 -> 10.122.164.147 DHCP DHCP Discover - Transaction
 ID 0x28a8606d
20180825 09:30:56.216137 10.122.164.147 -> 10.122.165.134 DHCP DHCP Offer - Transaction
 ID 0x28a8606d
20180825 09:30:56.217444 10.122.165.134 -> 10.122.164.147 DHCP DHCP Request - Transaction
 ID 0x28a8606d
20180825 09:30:56.218207 10.122.164.147 -> 10.122.165.134 DHCP DHCP ACK - Transaction
ID 0x28a8606d
```

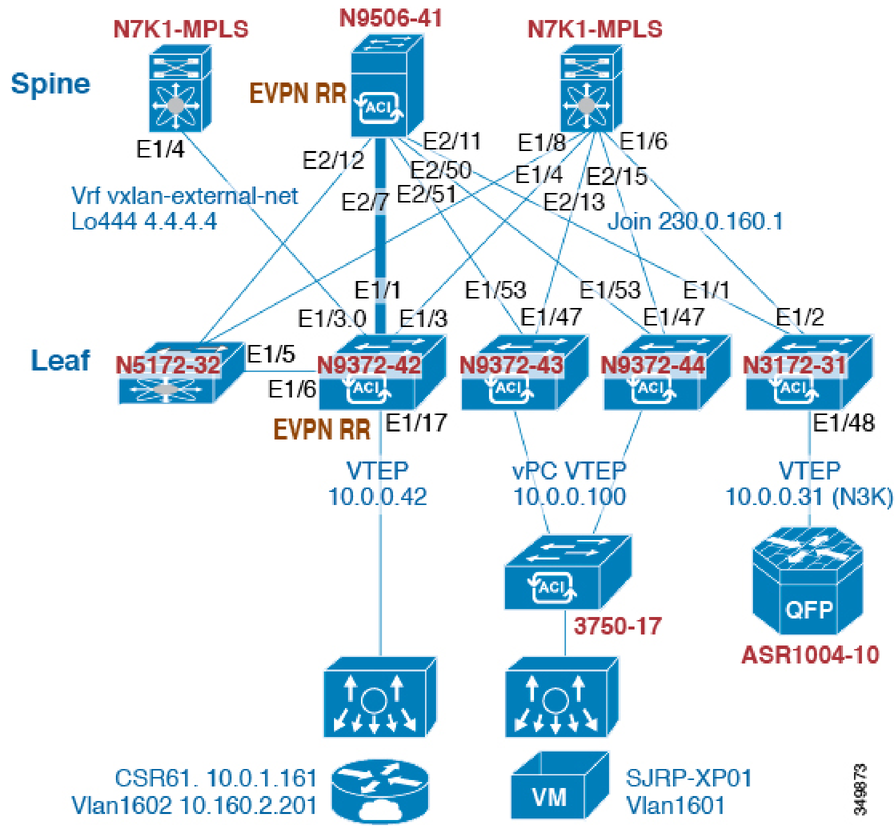
- DHCP Discover packet 9372-1 sent to DHCP server.

  giaddr is set to 10.122.165.134 (mgmt0) and suboptions 5/11/151 are set accordingly.

```
Bootstrap Protocol
  Message type: Boot Request (1)
  Hardware type: Ethernet (0x01)
  Hardware address length: 6
  Hops: 1
  Transaction ID: 0x28a8606d
  Seconds elapsed: O
  Bootp flags: 0x0000 (Unicast)
  Client IP address: 0.0.0.0 (0.0.0.0)
  Your (client) IP address: 0.0.0.0 (0.0.0.0)
  Next server IP address: 0.0.0.0 (0.0.0.0)
  Relay agent IP address: 10.122.165.134 (10.122.165.134)
  Client MAC address: Hughes_01:51:51 (00:00:10:01:51:51)
  Client hardware address padding: 00000000000000000000
  Server host name not given
  Boot file name not given
  Magic cookie: DHCP
  Option: (53) DHCP Message Type
    Length: 1
    DHCP: Discover (1)
  Option: (55) Parameter Request List
  Option: (61) Client identifier
  Option: (82) Agent Information Option
    Length: 47
    Option 82 Suboption: (1) Agent Circuit ID
    Option 82 Suboption: (2) Agent Remote ID
    Option 82 Suboption: (151) VRF name/VPN ID
    Option 82 Suboption: (11) Server ID Override
      Length: 4
      Server ID Override: 172.16.16.1 (172.16.16.1)
    Option 82 Suboption: (5) Link selection
      Length: 4
      Link selection: 172.16.16.0 (172.16.16.0)
```

Configuring vPC Peers Example

The following is an example of how to configure routing between vPC peers in the overlay VLAN for a DHCP relay configuration.



- Enable DHCP service.

```
service dhcp
```

- Configure DHCP relay.

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

- Create loopback under VRF where you need DHCP relay service.

```
interface loopback601
  vrf member evpn-tenant-kk1
  ip address 192.0.2.36/24
  ip router ospf 1 area 0      /* Only required for vPC VTEP. */
```

- Advertise LoX into the Layer 3 VRF BGP.

```
Router bgp 2
vrf X
   network 10.1.1.42/8
```

- Configure DHCP relay on the SVI under the VRF.

```
interface Vlan1601
  vrf member evpn-tenant-kk1
  ip address 10.160.1.254/8
  fabric forwarding mode anycast-gateway
  ip dhcp relay address 10.160.2.201
  ip dhcp relay source-interface loopback601
```

- Configure Layer 3 VNI SVI with **ip forward**.

```
interface Vlan1600
  vrf member evpn-tenant-kk1
    ip forward
```

- Create the routing VLAN/SVI for the vPC VRF.

Note Only required for vPC VTEP

```
Vlan 1605
interface Vlan1605
  vrf member evpn-tenant-kk1
  ip address 10.160.5.43/8
  ip router ospf 1 area 10.10.10.41
```

- Create the VRF routing.

Note Only required for vPC VTEP.

```
router ospf 1
vrf evpn-tenant-kk1
    router-id 10.160.5.43
```

vPC VTEP DHCP Relay Configuration Example

To address a need to configure a VLAN that is allowed across the MCT/peer-link, such as a vPC VLAN, an SVI can be associated to the VLAN and is created within the tenant VRF. This becomes an underlay peering, with the underlay protocol, such as OSPF, that needs the tenant VRF instantiated under the routing process.

Alternatively, instead of placing the SVI within the routing protocol and instantiate the Tenant-VRF under the routing process, you can use the static routes between the vPC peers across the MCT. This approach ensures that the reply from the server returns to the correct place and each VTEP uses a different loopback interface for the GiAddr.

The following are examples of these configurations:

- Configuration of SVI within underlay routing:

```
/* vPC Peer-1 */

router ospf UNDERLAY
vrf tenant-vrf

interface Vlan2000
  no shutdown
  mtu 9216
  vrf member tenant-vrf
  ip address 192.168.1.1/16
  ip router ospf UNDERLAY area 0.0.0.0


/* vPC Peer-2 */

router ospf UNDERLAY
vrf tenant-vrf

interface Vlan2000
  no shutdown
  mtu 9216
  vrf member tenant-vrf
  ip address 192.168.1.2/16
  ip router ospf UNDERLAY area 0.0.0.0
```

- Configuration of SVI using static routes between vPC peers across the MCT:

```
/* vPC Peer-1 */

interface Vlan2000
  no shutdown
  mtu 9216
  vrf member tenant-vrf
  ip address 192.168.1.1/16

vrf context tenant-vrf
ip route 192.168.1.2/16 192.168.1.1


/* vPC Peer-2 */

interface Vlan2000
  no shutdown
  mtu 9216
  vrf member tenant-vrf
  ip address 192.168.1.2/16

vrf context tenant-vrf
ip route 192.168.1.1/16 192.168.1.2
```

APPENDIX C

Configuring Layer 4 - Layer 7 Network Services Integration

This chapter contains the following sections:

About VXLAN Layer 4 - Layer 7 Services

This chapter covers insertion of Layer 4 – Layer 7 network services (firewall, load balancer, and so on) in a VXLAN fabric.

As opposed to traditional 3-tier network topologies, in which L4-L7 services are connected to the switches hosting the default gateway (aggregation/distribution), L4-L7 services in VXLAN fabrics are typically connected to the leaf or border switches, often referred to as *services leafs*.

You can attach a L4-L7 services device to a VXLAN fabric in various ways. This chapter addresses the considerations you must take depending on how the L4-L7 services device is attached and the requirements of the device and the network.

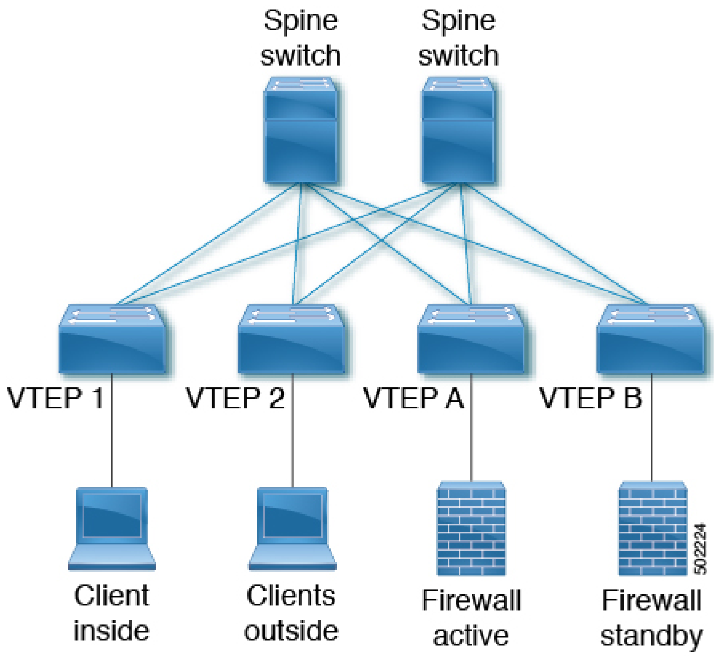
Integrating Layer 3 Firewalls in VXLAN Fabrics

This section provides details on how to integrate a firewall within a VXLAN EVPN fabric. A Layer-3 firewall involves separating different security zones.

When integrating a Layer-3 firewall in a VXLAN EVPN fabric with a distributed Anycast Gateway, each of these zones must correspond to a VRF/tenant on the fabric. The traffic within a tenant is routed by the fabric. Traffic between the tenants is routed by the firewall. This scenario often refers to an inter-tenant or tenant edge firewall.

Consider two zones: an inside zone and an outside zone. This scenario requires a VRF definition on the fabric. You can call the VRFs the inside VRF and the outside VRF. Traffic between subnets within the same VRF is routed on the VXLAN fabric using the distributed gateway. Traffic between VRFs is routed by the firewall where the rules are applied.

Figure 48: Topology Overview with Firewall Attachment



Single-Attached Firewall with Static Routing

If the firewall does not support running a routing protocol, you must have static routes on each VTEP pointing to the firewall as the next hop. The firewall also has static routes pointing to the Anycast Gateway IP as the next hop. The challenge with a static route is that the VTEP with an active firewall must be the one advertising the routes to the fabric. One way to accomplish this is to track the active firewall reachability via HMM and use this tracking to advertise routes into the fabric. When the active firewall is connected to VTEP A, VTEP A has a static route that tracks where the route is advertised if the firewall IP is learned as the HMM route. When the firewall fails and the standby firewall takes over, VTEP A learns the firewall IP using BGP, and VTEP B learns the firewall IP using HMM. VTEP A withdraws the route, and VTEP B advertises the route into the fabric. See the following example.

VTEP A and VTEP B:

```
Vlan 10
 Name inside
 Vn-segment 10010

Vlan 20
 Name outside
 Vn-segment 10020


Interface VLAN 10
 Description inside_vlan
 VRF member INSIDE
 IP address 10.1.1.254/24
 fabric forwarding mode anycast-gateway

Interface VLAN 20
 Description outside_vlan
 VRF member OUTSIDE
```

```
 IP address 20.1.1.254/24
 fabric forwarding mode anycast-gateway

interface nve1
 no shutdown
 host-reachability protocol bgp
 source-interface loopback1
 member vni 10010
  mcastgroup 239.1.1.1
 member vni 10020
  mcastgroup 239.1.1.1
 member vni 1001000 associate-vrf
 member vni 1002000 associate-vrf


track 10 ip route 10.1.1.1/32 reachability hmm
  vrf member INSIDE
!
VRF context INSIDE
 Vni 1001000
 IP route 20.1.1.0/24 10.1.1.1 track 10


track 20 ip route 20.1.1.1/32 reachability hmm
  vrf member OUTSIDE

!
VRF context OUTSIDE
 Vni 1001000
 IP route 10.1.1.0/24 20.1.1.1 track 20


VTEPA# show track 10 Track 10
IP Route 20.1.1.1/32 Reachability Reachability is UP

VTEPA# show ip route 20.1.1.0/24 vrf INSIDE
IP Route Table for VRF "INSIDE"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

20.1.1.0/24, ubest/mbest: 1/0
  *via 10.1.1.1 [1/0], 00:00:08, static

Firewall Failure on VTEP A caused the track to go down causing VTEP A to withdraw the static
 route.

VTEPA# show track 20 Track 20
IP Route 20.1.1.1/32 Reachability Reachability is DOWN

VTEPA# show ip route 20.1.1.0/24 vrf INSIDE
IP Route Table for VRF "RED"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

Route not found
```

Recursive Static Routes Distributed to the Rest of the Fabric

With this approach, the static routes are configured wherever the inside or outside VRF exists. As the next-hop is reachable through host routes (EVPN Route-Type2), the change of the active firewall to standby and vice versa is only seen locally and doesn’t introduce any churn to the other VXLAN fabric. This approach can help to better scale and improve convergence.

Any VTEP:

```
VRF context OUTSIDE
 Vni 1002000
 IP route 10.1.1.0/24 20.1.1.1
! static route on VTEP pointing to Firewall next hop
! firewall VIP 20.1.1.1

VRF context INSIDE
 Vni 1001000
 IP route 20.1.1.0/24 10.1.1.1
! static route on VTEP pointing to Firewall next hop
! firewall VIP 10.1.1.1
```

Redistribute Static Routes into BGP and Advertise to the Rest of the Fabric

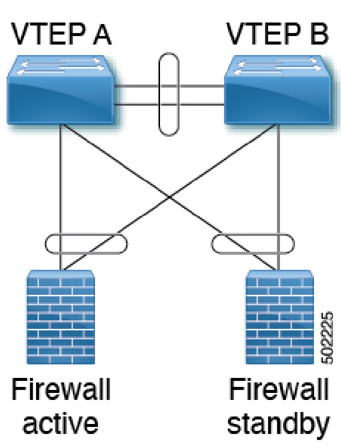
Through redistribution, we make the route toward the active firewall shown to the VTEP where it resides. The route is seen as a prefix route (EVPN Route-Type5), and as such, only the route toward the VTEP with the active firewall is seen. In the case of a firewall active/standby change, the tracking needs to detect the change and inform all of the remote VTEPs of this change. This behavior is equal to a route "delete" followed by an "add." This approach needs to notify all VTEPs with the VRF, and hence a wider churn can be seen.

VTEP A and VTEP B:

```
router bgp 65000
 vrf OUTSIDE
  address-family ipv4 unicast
   redistribute static route-map Static-to-BGP
```

Dual-Attached Firewall with Static Routing

Figure 49: Dual-Attached Firewall with Static Routing



VTEP A and VTEP B:

```
Vlan 10
 Name inside
 Vn-segment 10010

Vlan 20
 Name outside
 Vn-segment 10020

interface nve1
 no shutdown
 host-reachability protocol bgp
 source-interface loopback1
 member vni 10010
  mcastgroup 239.1.1.1
 member vni 10020
  mcastgroup 239.1.1.1
 member vni 1001000 associate-vrf
 member vni 1002000 associate-vrf


Interface VLAN 10
 Description inside_vlan
 VRF member INSIDE
 IP address 10.1.1.254/24
 fabric forwarding mode anycast-gateway

Interface VLAN 20
 Description outside_vlan
 VRF member OUTSIDE
 IP address 20.1.1.254/24
 fabric forwarding mode anycast-gateway

VRF context INSIDE
 Vni 1001000
 IP route 20.1.1.0/24 10.1.1.1
! static route on VTEP pointing to Firewall next hop
! firewall VIP 10.1.1.1
VRF context OUTSIDE
 Vni 1002000
 IP route 10.1.1.0/24 20.1.1.1
! static route on VTEP pointing to Firewall next hop
! firewall VIP 20.1.1.1


router bgp 65000
 vrf INSIDE
  address-family ipv4 unicast
   redistribute static route-map INSIDE-to-BGP
 vrf OUTSIDE
  address-family ipv4 unicast
   redistribute static route-map OUTSIDE-to-BGP
```

Single-Attached Firewall with eBGP Routing

If the firewall supports BGP, one option is to use BGP as a protocol between the firewall and the service VTEP. Peering using the anycast IP is not supported. The recommended design is to use dedicated loopback IPs on each VTEP and peer using the loopback. As long as the loopback interfaces are not advertised via

EVPN, the same IP address could be used on all of the belonging VTEPs. We recommend using individual IP addresses on a per-VTEP basis.

Reachability to the loopback from the firewall can be configured using a static route on the firewall, pointing to the Anycast Gateway IP on the VTEPs.

In the following example, an eBGP peering is established from the VTEPs, which are in AS 65000, and the firewall in AS 65002. The BGP peering with iBGP is not supported.

> **Note** When having eBGP peering to active/standby firewalls connected to different VTEPs, **export-gateway-ip** must be enabled.
>
> Do not use Anycast Gateway for BGP peerings.

VTEP A:

```
Vlan 10
 Name inside
 Vn-segment 10010

Vlan 20
 Name outside
 Vn-segment 10020


Interface VLAN 10
 Description inside_vlan
 VRF member INSIDE
IP address 10.1.1.254/24
fabric forwarding mode anycast-gateway

Interface loopback100
 Vrf member INSIDE
 Ip address 172.16.1.253/32

 Interface VLAN 20
 Description outside_vlan
 VRF member OUTSIDE
 IP address 20.1.1.254/24
 fabric forwarding mode anycast-gateway

 Interface loopback101
 Vrf member OUTSIDE
 Ip address 172.18.1.253/32


router bgp 65000
 vrf INSIDE
 ! peer with Firewall Inside
 neighbor 10.1.1.0/24 remote-as 65123
 update-source loopback100
 ebgp-multihop 5
 address-family ipv4 unicast
  local-as 65051 no-prepend replace-as

 vrf OUTSIDE
 ! peer with Firewall Outside
 neighbor 20.1.1.0/24 remote-as 65123
 update-source loopback101
 ebgp-multihop 5
```

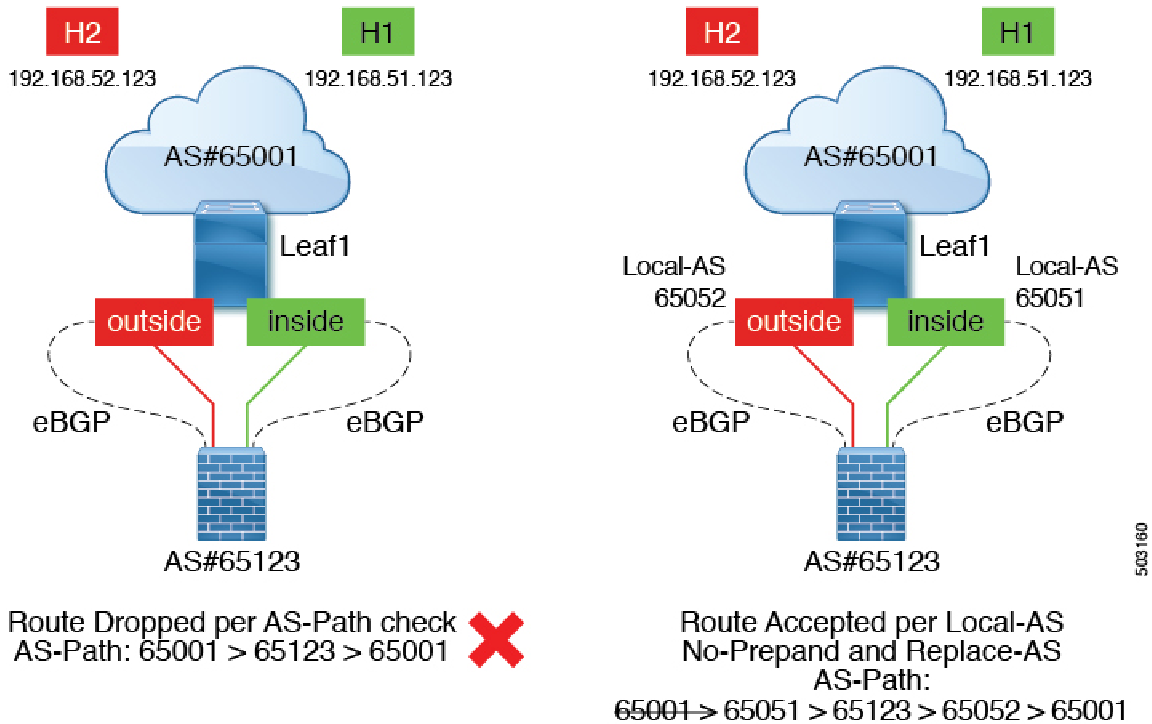
```
 address-family ipv4 unicast
  local-as 65052 no-prepend replace-as
```

VTEP B:

```
Vlan 10
 Name inside
 Vn-segment 10010

Vlan 20
 Name outside
 Vn-segment 10020
Interface VLAN 10
 Description inside_vlan
 VRF member INSIDE
IP address 10.1.1.254/24
fabric forwarding mode anycast-gateway

Interface loopback100
 Vrf member INSIDE
 Ip address 172.16.1.254/32

 Interface VLAN 20
 Description outside_vlan
 VRF member OUTSIDE
 IP address 20.1.1.254/24
 fabric forwarding mode anycast-gateway

 Interface loopback101
 Vrf member OUTSIDE
 Ip address 172.18.1.254/32


router bgp 65000
 vrf INSIDE
 ! peer with Firewall Inside
 neighbor 10.1.1.0/24 remote-as 65123
 update-source loopback100
 ebgp-multihop 5
 address-family ipv4 unicast
  local-as 65051 no-prepend replace-as

 vrf OUTSIDE
 ! peer with Firewall Outside
 neighbor 20.1.1.0/24 remote-as 65123
 update-source loopback101
 ebgp-multihop 5
 address-family ipv4 unicast
  local-as 65052 no-prepend replace-as
```

With the VXLAN fabric generally being in a single BGP Autonomous System (AS), the AS of the inside VRF and the outside VRF is the same. BGP does not install routes that are received from its own AS. Therefore, we need to adjust the AS-path to override this rule. Various approaches exist, including disabling the rule that BGP drops routes from its own AS, which has further implications to the network. To keep all of the BGP protection mechanics in place, the “local-as” approach allows you to mimic routes being originated from a different AS. We recommend inserting the “local-as #ASN# no-prepend replace-as” on each firewall peering with different “local-as” per VRF.

Figure 50: eBGP AS-Path Check



Dual-Attached Firewall with eBGP Routing

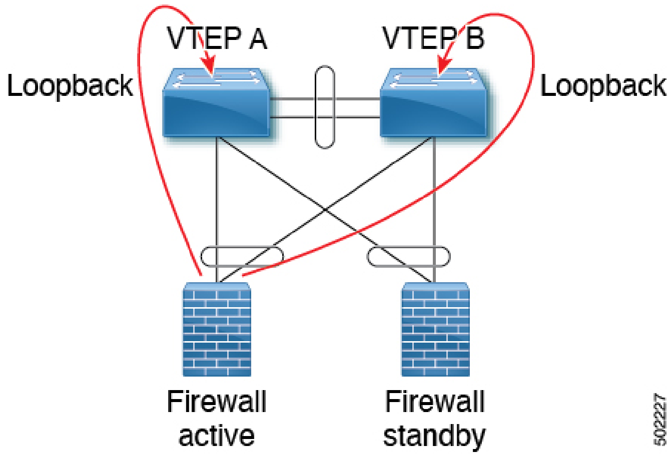
If the firewall supports BGP, one option is to use BGP as a protocol between the firewall and the service VTEP. Peering using the anycast IP is not supported. The recommended design is to use dedicated loopback IPs on each VTEP and peer using the loopback. As long as the loopback interfaces are not advertised via EVPN, the same IP address could be used on all of the belonging VTEPs. We recommend using individual IP addresses on a per-VTEP basis. For vPC environments, it is required.

Reachability to the loopback from the firewall can be configured using a static route on the firewall, pointing to the Anycast Gateway IP on the VTEPs.

In vPC deployments, you must have a per-VRF peering via a vPC peer-link. In addition to the per-VRF peering, you can enable the advertisement of prefix routes (EVPN Route-Type 5) using the **advertise-pip** command. For vPC with fabric peering, the per-VRF peering is not necessary, and the advertisement of prefix routes (EVPN Route-Type5) is required.

In the following example, an eBGP peering is established from the VTEPs, which are in AS 65000, and the firewall in AS 65002. The BGP peering with iBGP is not supported.

Figure 51: Dual-Attached Firewall with eBGP



Note When having eBGP peering to active/standby firewalls connected to different VTEPs, **export-gateway-ip** must be enabled.

Do not use Anycast Gateway for BGP peerings.

VTEP A:

```
Vlan 10
 Name inside
 Vn-segment 10010

Vlan 20
 Name outside
 Vn-segment 10020


Interface VLAN 10
 Description inside_vlan
 VRF member INSIDE
IP address 10.1.1.254/24
fabric forwarding mode anycast-gateway

Interface loopback100
 Vrf member INSIDE
 Ip address 172.16.1.253/32

 Interface VLAN 20
 Description outside_vlan
 VRF member OUTSIDE
 IP address 20.1.1.254/24
 fabric forwarding mode anycast-gateway

 Interface loopback101
 Vrf member OUTSIDE
 Ip address 172.18.1.253/32


router bgp 65000
vrf INSIDE
 ! peer with Firewall Inside
```

```
 neighbor 10.1.1.0/24 remote-as 65123
 update-source loopback100
 ebgp-multihop 5
 address-family ipv4 unicast
  local-as 65051 no-prepend replace-as

 vrf OUTSIDE
 ! peer with Firewall Outside
 neighbor 20.1.1.0/24 remote-as 65123
 update-source loopback101
 ebgp-multihop 5
 address-family ipv4 unicast
  local-as 65052 no-prepend replace-as
```

VTEP B:

```
Vlan 10
 Name inside
 Vn-segment 10010

Vlan 20
 Name outside
 Vn-segment 10020


Interface VLAN 10
 Description inside_vlan
 VRF member INSIDE
IP address 10.1.1.254/24
fabric forwarding mode anycast-gateway

Interface loopback100
 Vrf member INSIDE
 Ip address 172.16.1.254/32

 Interface VLAN 20
 Description outside_vlan
 VRF member OUTSIDE
 IP address 20.1.1.254/24
 fabric forwarding mode anycast-gateway

 Interface loopback101
 Vrf member OUTSIDE
 Ip address 172.18.1.254/32


router bgp 65000
 vrf INSIDE
 ! peer with Firewall Inside
 neighbor 10.1.1.0/24 remote-as 65123
 update-source loopback100
 ebgp-multihop 5
 address-family ipv4 unicast
  local-as 65051 no-prepend replace-as

 vrf OUTSIDE
 ! peer with Firewall Outside
 neighbor 20.1.1.0/24 remote-as 65123
 update-source loopback101
 ebgp-multihop 5
 address-family ipv4 unicast
  local-as 65052 no-prepend replace-as
```

Per-VRF Peering via vPC Peer-Link

VTEP A and VTEP B:

```
vlan 3966
! vlan use for peering between the vPC VTEPS

vlan 3967
! vlan use for peering between the vPC VTEPS

system nve infra-vlans 3966,3967

interface vlan 3966
vrf memner INSIDE
 ip address 100.1.1.1/31

interface vlan 3967
 vrf memner OUTSIDE
 ip address 100.1.2.1/31

router bgp 65000
 vrf INSIDE
 neighbor 100.1.1.0 remote-as 65000
 update-source vlan 3966
 next-hop self
 address-family ipv4 unicast

 vrf OUTSIDE
 neighbor 100.1.2.0 remote-as 65000
 update-source vlan 3967
 next-hop self
 address-family ipv4 unicast
```

The routes learned in each VRF are advertised to the rest of the fabric via BGP EVPN updates.

Single-Attached Firewall with OSPF

The following example shows a configuration snippet from VTEP A running OSPF peering with the firewall.

SVIs are defined on the VTEP for both inside and outside VRFs. The VTEP peers with the firewall on each of these VRFs dynamically learn routing information to go from one VRF to the other.

VTEP A and VTEP B:

```
vlan 10
 name inside
 vn-segment 10010

vlan 20
 name outside
 vn-segment 10020

interface VLAN 10
 Description inside_vlan
 VRF member INSIDE
 IP address 10.1.1.254/24
 IP router ospf 1 area 0
 fabric forwarding mode anycast-gateway

Interface VLAN 20
 Description outside_vlan
 VRF member OUTSIDE
```

```
 IP address 20.1.1.254/24
 IP router ospf 1 area 0
 fabric forwarding mode anycast-gateway

interface nve1
 no shutdown
 host-reachability protocol bgp
 source-interface loopback1
 member vni 10010
  mcastgroup 239.1.1.1
 member vni 10020
  mcastgroup 239.1.1.1
 member vni 1001000 associate-vrf
 member vni 1002000 associate-vrf

router ospf 1
 router-id 192.168.1.1
  vrf INSIDE
  VRF OUTSIDE


VTEPA# show ip route ospf-1 vrf OUTSIDE
 IP Route Table for VRF "OUTSIDE"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

10.1.1.0/24, ubest/mbest: 1/0
  *via 20.1.1.1 Vlan20, [110/41], 1w5d, ospf-1, intra

VTEPA# show ip route ospf-1 vrf INSIDE
IP Route Table for VRF "INSIDE"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

20.1.1.0/24, ubest/mbest: 1/0
  *via 10.1.1.1 Vlan10, [110/41], 1w5d, ospf-1, intra
```

This route is then redistributed into BGP and advertised through the EVPN fabric so that all other VTEPs have all routes in each VRF pointing to VTEP A as the next hop.

Redistribute OSPF Routes into BGP and Advertise to the Rest of the Fabric

VTEP A and VTEP B:

```
router bgp 65000
 vrf OUTSIDE
  address-family ipv4 unicast
   redistribute ospf 1 route-map OUTSIDEOSPF-to-BGP
 vrf INSIDE
  address-family ipv4 unicast
   redistribute ospf 1 route-map INSIDEOSPF-to-BGP


VTEPA# show ip route 10.1.1.0/24 vrf OUTSIDE

10.1.1.0/24 ubest/mbest: 1/0
  *via 10.1.1.18%default, [200/41], 1w1d, bgp-65000, internal, tag 65000 (evpn) segid:
200100 tunnelid: 0xa010112 encap: VXLAN
```

Traffic is VXLAN encapsulated from VTEP to services VTEP and decapsulated and sent to the firewall. The firewall enforces the rules and sends the traffic to the services VTEP on the inside VRF. This traffic is then VXLAN encapsulated and sent to the destination VTEP where traffic is decapsulated and sent to the end client.

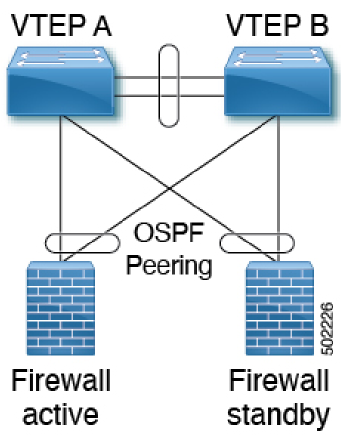
Firewall Failover

When the active firewall fails and the standby firewall takes over, routes are withdrawn from service VTEP A and advertised to the fabric by service VTEP B.

Dual-Attached Firewall with OSPF

Cisco NX-OS supports dynamic OSPF peering over vPC using Layer 3, which enables firewall connectivity using vPC and establishes OSPF peering over this link. The VLAN used to establish peering between the Cisco Nexus 9000 switches and the firewall must be a non-VXLAN-enabled VLAN.

Figure 52: Dual-Attached Firewall with OSPF



Note Do not use Anycast Gateway for OSPF adjacencies.

VTEP A:

```
Vlan 10
 Name inside

Vlan 20
 Name outside

Interface VLAN 10
 Description inside_vlan
 VRF member INSIDE
 IP address 10.1.1.253/24
 Ip router ospf 1 area 0

Interface VLAN 20
 Description outside_vlan
 VRF member OUTSIDE
 IP address 20.1.1.253/24
 Ip router ospf 1 area 0
```

```
vpc domain 100
 layer3 peer-router
 peer-gateway
 peer-switch
 peer-keepalive destination x.x.x.x source x.x.x.x peer-gateway
 ipv6 nd synchronize
 ip arp synchronize

router ospf 1
 vrf INSIDE VRF OUTSIDE
```

VTEP B:

```
Vlan 10
 Name inside

Vlan 20
 Name outside

Interface VLAN 10
 Description inside_vlan
 VRF member INSIDE
 IP address 10.1.1.254/24
 Ip router ospf 1 area 0

Interface VLAN 20
 Description outside_vlan
 VRF member OUTSIDE
 IP address 20.1.1.254/24
 Ip router ospf 1 area 0

vpc domain 100
 layer3 peer-router
 peer-gateway
 peer-switch
 peer-keepalive destination x.x.x.x source x.x.x.x peer-gateway
 ipv6 nd synchronize
 ip arp synchronize

router ospf 1
 vrf INSIDE VRF OUTSIDE


VTEPA# show ip route ospf-1 vrf OUTSIDE
IP Route Table for VRF "OUTSIDE"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

10.1.1.0/24, ubest/mbest: 1/0
  *via 20.1.1.1 Vlan20, [110/41], 1w5d, ospf-1, intra

VTEPA# show ip route ospf-1 vrf INSIDE
IP Route Table for VRF "INSIDE"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

20.1.1.0/24, ubest/mbest: 1/0
  *via 10.1.1.1 Vlan10, [110/41], 1w5d, ospf-1, intra
```

Redistribute OSPF Routes into BGP and Advertise to the Rest of the Fabric

VTEP A and VTEP B:

```
router bgp 65000
 vrf OUTSIDE
  address-family ipv4 unicast
   redistribute ospf 1 route-map OUTSIDEOSPF-to-BGP
 vrf INSIDE
  address-family ipv4 unicast
   redistribute ospf 1 route-map INSIDEOSPF-to-BGP
```

Firewall as Default Gateway

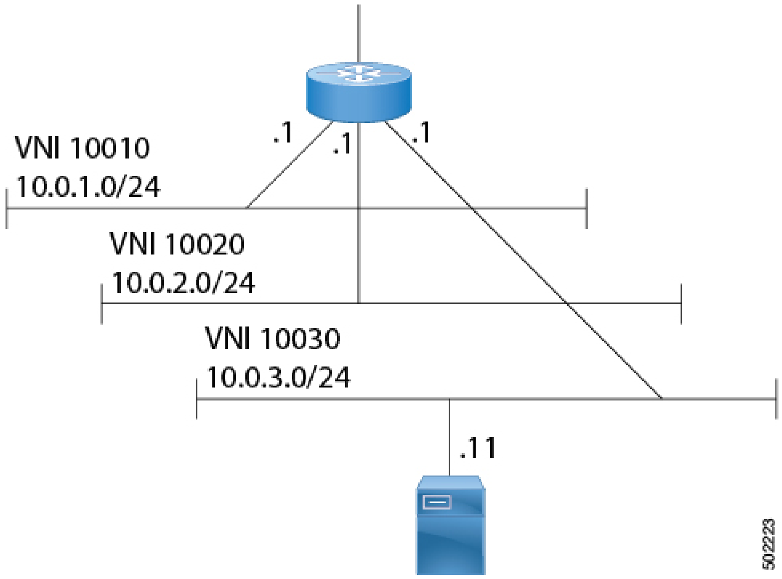
In this deployment model, the VXLAN fabric is a Layer 2 fabric, and the default gateway resides on the firewall.

For example:

```
vlan 10
  name WEB
  vn-segment 10010
vlan 20
  name APPLICATION
  vn-segment 10020
vlan 30
  name DATABASE
  vn-segment 10030

interface nve1
  no shutdown
  host-reachability protocol bgp
  source-interface loopback1
  member vni 10010
   mcastgroup 239.1.1.1
  member vni 10020
   mcastgroup 239.1.1.1
  member vni 10030
   mcastgroup 239.1.1.1
```

The firewall has a logical interface in each VNI and is the default gateway for all endpoints. Every inter-VNI communication flows through the firewall. Take special care with the sizing of the firewall so that it does not become a bottleneck. Therefore, use this design in environments with low-bandwidth requirements.

Figure 53: Firewall as Default Gateway with a Layer-2 VXLAN Fabric



Transparent Firewall Insertion

Transparent firewalls or Layer 2 firewalls (including IPS/IDS) typically bridge between an inside VLAN and outside VLAN and inspect traffic as it traverses through them. VLAN stitching is done by placing the default gateway for the service on the inside VLAN. The Layer 2 reachability to this gateway is done on the outside VLAN.

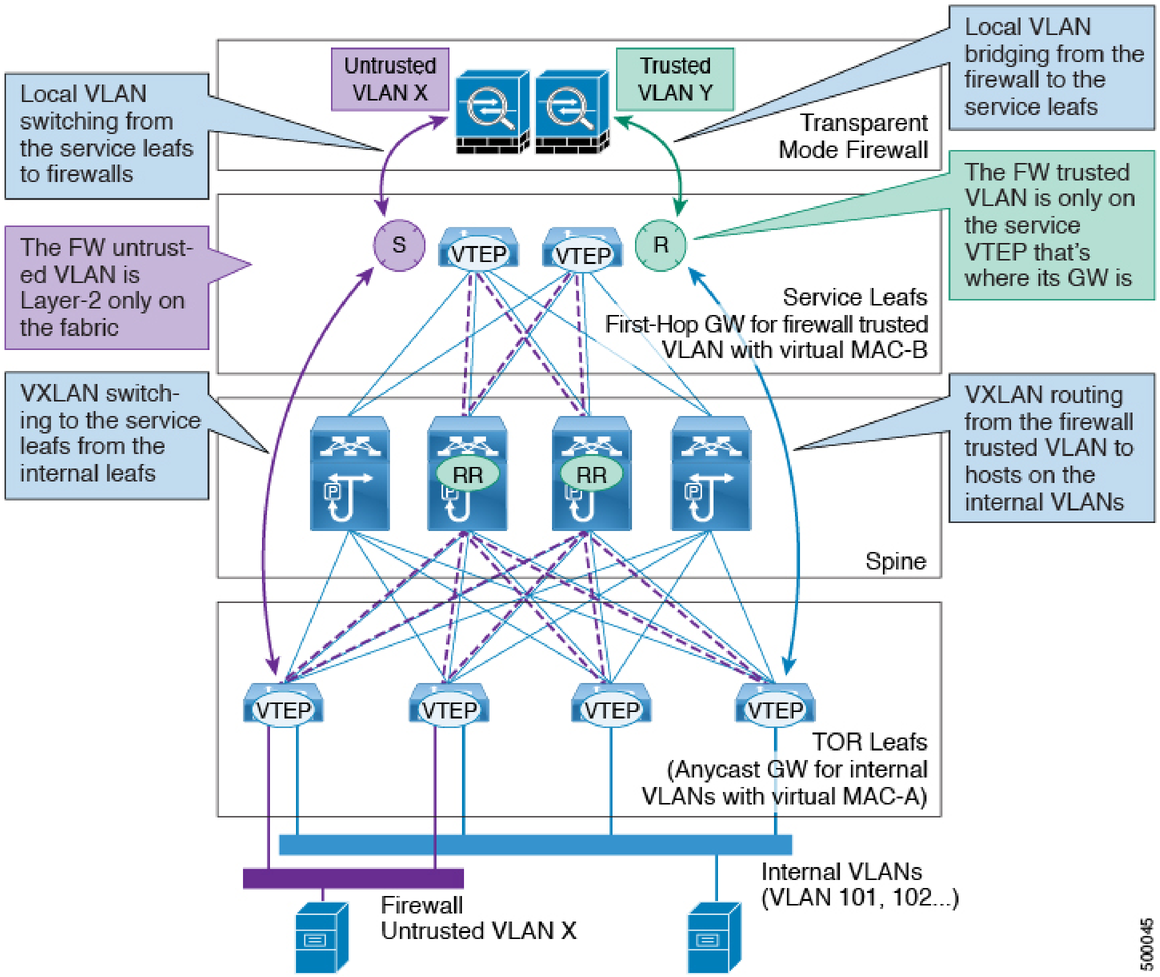
Overview of EVPN with Transparent Firewall Insertion

The topology contains the following types of VLANs:

- Internal VLAN (a regular VXLAN on ToR leafs with Anycast Gateway)
- Firewall untrusted VLAN X
- Firewall trusted VLAN Y

In this topology, the traffic that goes from VLAN X to other VLANs must go through a transparent Layer 2 firewall that is attached to the service leafs. This topology utilizes an approach of an untrusted VLAN X and a trusted VLAN Y. All ToR leafs have a Layer 2 VNI VLAN X. There is no SVI for VLAN X. The service leafs that are connected to the firewall have Layer 2 VNI VLAN X, non-VXLAN VLAN Y, and SVI Y with an HSRP gateway.

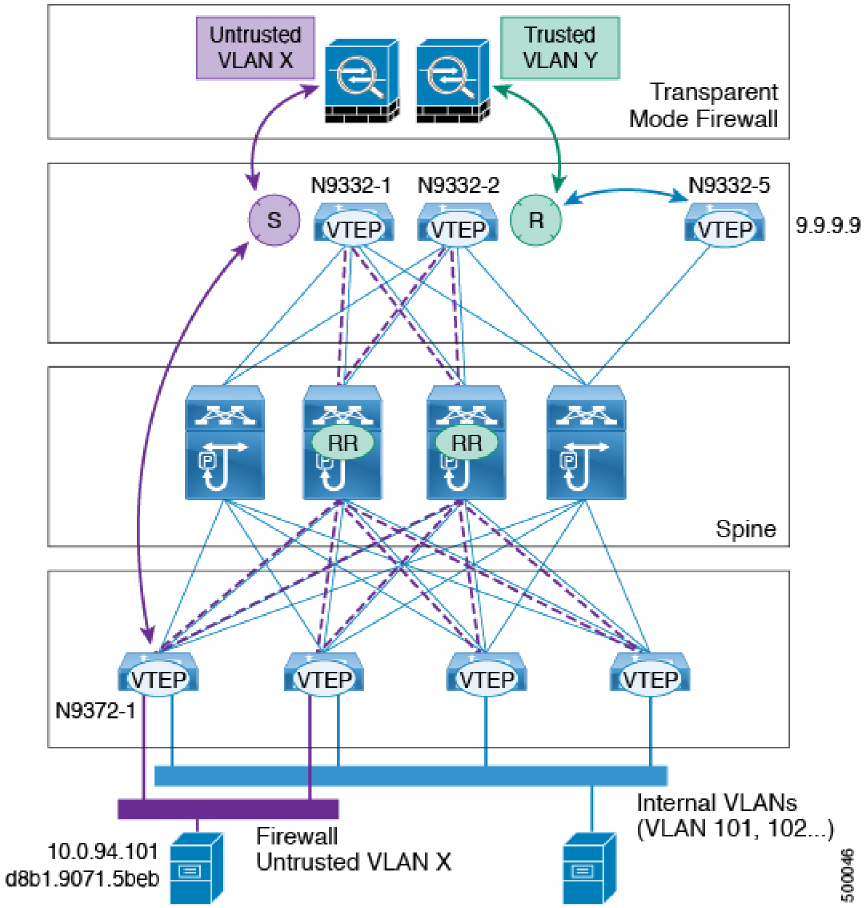
Overview of EVPN with Transparent Firewall Insertion



Note For VXLAN EVPN, we recommend using the distributed Anycast Gateway with transparent firewall insertion. Doing so allows all VLANs to be VXLAN enabled. When using an HSRP/VRRP-based First-Hop Gateway, the VLAN for the SVI can't be VXLAN enabled and should reside on a vPC pair for redundancy.

EVPN with Transparent Firewall Insertion Example

Example of EVPN with Transparent Firewall Insertion



- Host in VLAN X: 10.1.94.101
- ToR leaf: N9372-1
- Service leaf in vPC: N9332-1 and N9332-2
- Border leaf: N9332-5

ToR Leaf Configuration

```
vlan 94
vn-segment 100094

interface nve1
  member vni 100094
   mcastgroup 239.1.1.1

router bgp 64500
  routerid 1.1.2.1
  neighbor 1.1.1.1 remote-as 64500
  address-family l2vpn evpn
    send-community extended
```

```
  neighbor 1.1.1.2 remote-as 64500
  address-family l2vpn evpn
    send-community extended
  vrf Ten1
   address-family ipv4 unicast
     advertise l2vpn evpn

evpn
vni 100094 l2
  rd auto
  route-target import auto
  route-target export auto
```

Service Leaf 1 Configuration Using HSRP

```
vlan 94
description untrusted_vlan
  vn-segment 100094

vlan 95
  description trusted_vlan

vpc domain 10
  peer-switch
  peer-keepalive destination 10.1.59.160
  peer-gateway
  auto-recovery
  ip arp synchronize

interface Vlan2
description vpc_backup_svi_for_overlay
  no shutdown
  no ip redirects
  ip address 10.10.60.17/30
  no ipv6 redirects
  ip router ospf 100 area 0.0.0.0
  ip ospf bfd
  ip pim sparsemode

interface Vlan95
description SVI_for_trusted_vlan
  no shutdown
  mtu 9216
  vrf member Ten-1
  no ip redirects
  ip address 10.0.94.2/24
  hsrp 0
   preempt priority 255
   ip 10.0.94.1

interface nve1
  member vni 100094
   mcast-group 239.1.1.1

router bgp 64500
  routerid 1.1.2.1
  neighbor 1.1.1.1 remote-as 64500
  address-family l2vpn evpn
   send-community extended
  neighbor 1.1.1.2 remote-as 64500
   address-family l2vpn evpn
    send-community extended
  vrf Ten-1
   address-family ipv4 unicast
```

```
    network 10.0.94.0/24 /*advertise /24 for SVI 95 subnet; it is not VXLAN anymore*/
    advertise l2vpn evpn

evpn
vni 100094 l2
 rd auto
 route-target import auto
 route-target export auto
```

Service Leaf 2 Configuration Using HSRP

```
vlan 94
  description untrusted_vlan
  vnsegment 100094

vlan 95
  description trusted_vlan

vpc domain 10
  peer-switch
  peer-keepalive destination 10.1.59.159
  peer-gateway
  auto-recovery
  ip arp synchronize

interface Vlan2
description vpc_backup_svi_for_overlay
  no shutdown
  no ip redirects
  ip address 10.10.60.18/30
  no ipv6 redirects
  ip router ospf 100 area 0.0.0.0
  ip pim sparsemode

interface Vlan95
  description SVI_for_trusted_vlan
  no shutdown
  mtu 9216
  vrf member Ten-1
  no ip redirects
  ip address 10.0.94.3/24
  hsrp 0
   preempt priority 255
   ip 10.0.94.1

interface nve1
  member vni 100094
   mcastgroup 239.1.1.1

router bgp 64500
  router-id 1.1.2.1
  neighbor 1.1.1.1 remote-as 64500
  address-family l2vpn evpn
    send-community extended
  neighbor 1.1.1.2 remote-as 64500
  address-family l2vpn evpn
   send-community extended
  vrf Ten-1
    address-family ipv4 unicast
     network 10.0.94.0/24 /*advertise /24 for SVI 95 subnet; it is not VXLAN anymore*/
     advertise l2vpn evpn

evpn
vni 100094 l2
```

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

Show Command Examples

Display information about the ingress leaf learned local MAC from host:

```
switch# sh mac add vl 94 | i 5b|MAC
* primary entry, G - Gateway MAC, (R) Routed - MAC, O - Overlay MAC
VLAN MAC Address Type age Secure NTFY Ports
* 94 d8b1.9071.5beb dynamic 0 F F Eth1/1
```

Display information about the service leaf found MAC of host:

Note In VLAN 94, the service leaf learned the host MAC from the remote peer by BGP.

```
switch# sh mac add vl 94 | i VLAN|eb

VLAN MAC Address Type age Secure NTFY Ports
* 94 d8b1.9071.5beb dynamic 0 F F nve1(1.1.2.1)

switch#  sh mac add vl 94 | i VLAN|eb

VLAN MAC Address Type age Secure NTFY Ports
* 94 d8b1.9071.5beb dynamic 0 F F nve1(1.1.2.1)

switch#  sh mac add vl 95 | i VLAN|eb

VLAN MAC Address Type age Secure NTFY Ports
+ 95 d8b1.9071.5beb dynamic 0 F F Po300

switch#  sh mac add vl 95 | i VLAN|eb

VLAN MAC Address Type age Secure NTFY Ports
+ 95 d8b1.9071.5beb dynamic 0 F F Po300
```

Display information about service leaf learned ARP for host on VLAN 95:

```
switch# sh ip arp vrf ten-1
Address     Age      MAC Address     Interface
10.0.94.101 00:00:26 d8b1.9071.5beb  Vlan95
```

Service Leaf learns 9.9.9.9 from EVPN.

```
switch# sh ip route vrf ten-1 9.9.9.9
IP Route Table for VRF "Ten-1"
'*' denotes best ucast nexthop
'**' denotes best mcast nexthop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

9.9.9.9/32, ubest/mbest: 1/0
  *via 1.1.2.7%default, [200/0], 02:57:27, bgp64500,internal, tag 65000 (evpn) segid: 10011
tunnelid: 0x1
010207 encap: VXLAN
```

Display information about the border leaf learned host routes by BGP:

```
switch# sh ip route 10.0.94.101

IP Route Table for VRF "default"
'*' denotes best ucast nexthop
'**' denotes best mcast nexthop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

10.0.94.0/24, ubest/mbest: 1/0
  *via 10.100.5.0, [20/0], 03:14:27, bgp65000,external, tag 6450
```

Service Redirection in VXLAN EVPN Fabrics

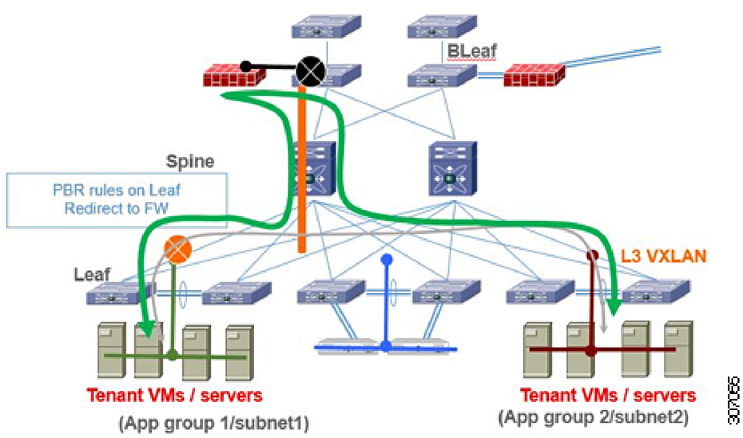
Today, insertion of service appliances (also referred to as service nodes or service endpoints) such as firewalls, load-balancers, etc are needed to secure and optimize applications within a data center. This section describes the Layer 4-Layer 7 service insertion and redirection features offered on VXLAN EVPN fabrics that provides sophisticated mechanisms to onboard and selectively redirect traffic to these services.

Use of Policy-Based Redirect for Services Insertion

Policy-based redirect (PBR) provides a mechanism to bypass a routing table lookup and redirect traffic to a next-hop IP reachable over VXLAN. The feature enables service redirection to Layer 4-Layer 7 devices such as firewalls and load balancers.

PBR involves configuring a route-map with rules that dictate where traffic must be forwarded. The route map is applied on the tenant SVI to influence traffic coming from the host-facing interfaces to a next hop reachable via the fabric.

In scenarios where traffic is coming to a VTEP from the overlay and needs to be redirected to another next hop, the PBR policy must be applied on the fabric facing Layer-3 VNI Interface.



In the previous figure, communication between App group 1 and App group 2 takes place via inter-VLAN/VNI routing in the tenant VRF by default. If there is a requirement where traffic from App group 1 to App group 2 must go through a firewall, a PBR policy can be used to redirect traffic. The example in section “Configuration Example for Policy-Based Redirect” provides the necessary configuration that redirects the traffic flow.

This VXLAN PBR functionality is very basic and lacks many of the required functionality for proper insertion of services in VXLAN fabric. Hence the recommendation is to instead look at ePBR for all the reasons explained in Enhanced-Policy Based Redirect (ePBR), on page 493 section.

Guidelines and Limitations for Policy-Based Redirect

The following guidelines and limitations apply to PBR over VXLAN.

- The following platforms support PBR over VXLAN:
  - Cisco Nexus 9332C and 9364C switches
  - Cisco Nexus 9300-EX switches
  - Cisco Nexus 9300-FX/FX2/FX3 switches
  - Cisco Nexus 9300-GX switches
  - Cisco Nexus 9504 and 9508 switches with -EX/FX line cards
- PBR over VXLAN doesn't support the following features:VTEP ECMP, and the **load-share** keyword in the **set {ip | ipv6} next-hop** *ip-address* command.
- When you configure **bestpath as-path multipath-relax**, BGP installs all the multi-paths for IPv4 and IPv6 as best-path in URIB with metric 0.

Enabling the Policy-Based Redirect Feature

To configure basic PBR, in cases where the advanced (and recommended) ePBR functions are not deployed, see the following sections:

- Enabling the Policy-Based Redirect Feature, on page 334
- Configuring a Route Policy, on page 335
- Verifying the Policy-Based Redirect Configuration, on page 336
- Configuration Example for Policy-Based Redirect, on page 336

Before you begin

Enable the policy-based redirect feature before you can configure a route policy.

SUMMARY STEPS

1. **configure terminal**
2. **[no] feature pbr**
3. (Optional) **show feature**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**[no] feature pbr** **Example:** `switch(config)# feature pbr`	Enables the policy-based routing feature.
Step 3	(Optional) **show feature** **Example:** `switch(config)# show feature`	Displays enabled and disabled features.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Saves this configuration change.

Configuring a Route Policy

You can use route maps in policy-based routing to assign routing policies to the inbound interface. Cisco NX-OS routes the packets when it finds a next hop and an interface.

Note The switch has a RACL TCAM region by default for IPv4 traffic.

Before you begin

Configure the RACL TCAM region (using TCAM carving) before you apply the policy-based routing policy. For instructions, see the "Configuring ACL TCAM Region Sizes" section in the Cisco Nexus 9000 Series NX-OS Security Configuration Guide, Release 9.2(x).

SUMMARY STEPS

1. **configure terminal**
2. **interface** *type slot/port*
3. **{ip | ipv6} policy route-map** *map-name*
4. **route-map** *map-name* **[permit | deny]** [*seq*]
5. **match {ip | ipv6} address access-list-name** *name* [*name*...]
6. **set ip next-hop** *address1*
7. **set ipv6 next-hop** *address1*
8. (Optional) **set interface null0**
9. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**interface** *type slot/port* **Example:** `switch(config)# interface ethernet 1/2`	Enters interface configuration mode.
Step 3	**{ip | ipv6} policy route-map** *map-name* **Example:** `switch(config-inf)# ip policy route-map Testmap`	Assigns a route map for IPv4 or IPv6 policy-based routing to the interface.
Step 4	**route-map** *map-name* **[permit | deny]** [*seq*] **Example:** `switch(config-inf)# route-map Testmap`	Creates a route map or enters route-map configuration mode for an existing route map. Use *seq* to order the entries in a route map.
Step 5	**match {ip | ipv6} address access-list-name** *name* [*name...*] **Example:** `switch(config-route-map)# match ip address access-list-name ACL1`	Matches an IPv4 or IPv6 address against one or more IPv4 or IPv6 access control lists (ACLs). This command is used for policy-based routing and is ignored by route filtering or redistribution.
Step 6	**set ip next-hop** *address1* **Example:** `switch(config-route-map)# set ip next-hop 192.0.2.1`	Sets the IPv4 next-hop address for policy-based routing.
Step 7	**set ipv6 next-hop** *address1* **Example:** `switch(config-route-map)# set ipv6 next-hop 2001:0DB8::1`	Sets the IPv6 next-hop address for policy-based routing.
Step 8	(Optional) **set interface null0** **Example:** `switch(config-route-map)# set interface null0`	Sets the interface that is used for routing. Use the **null0** interface to drop packets.
Step 9	(Optional) **copy running-config startup-config** **Example:** `switch(config-route-map)# copy running-config startup-config`	Saves this configuration change.

Verifying the Policy-Based Redirect Configuration

To display the policy-based redirect configuration information, perform one of the following tasks:

Command	Purpose
show [ip | ipv6] policy [*name*]	Displays information about an IPv4 or IPv6 policy.
show route-map [*name*] **pbr-statistics**	Displays policy statistics.

Use the **route-map** *map-name* **pbr-statistics** command to enable policy statistics. Use the **clear route-map** *map-name* **pbr-statistics** command to clear these policy statistics.

Configuration Example for Policy-Based Redirect

Perform the following configuration on all tenant VTEPs, excluding the service VTEP.

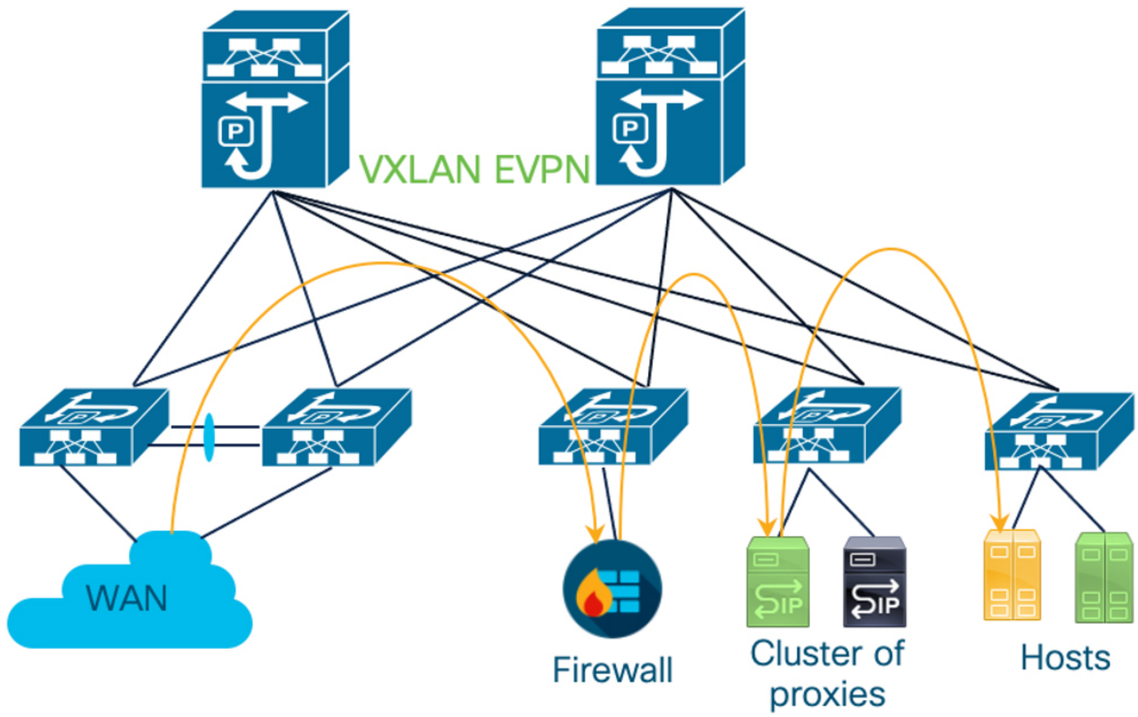
```
feature pbr

ipv6 access-list IPV6_App_group_1
10 permit ipv6 any 2001:10:1:1::0/64

ip access-list IPV4_App_group_1
10 permit ip any 10.1.1.0/24

ipv6 access-list IPV6_App_group_2
10 permit ipv6 any 2001:20:1:1::0/64

ip access-list IPV4_App_group_2
10 permit ip any 20.1.1.0/24

route-map IPV6_PBR_Appgroup1 permit 10
  match ipv6 address IPV6_App_group_2
  set ipv6 next-hop 2001:100:1:1::20  (next hop is that of the firewall)

route-map IPV4_ PBR_Appgroup1 permit 10
  match ip address IPV4_App_group_2
  set ip next-hop 10.100.1.20 (next hop is that of the firewall)

route-map IPV6_PBR_Appgroup2 permit 10
  match ipv6 address IPV6_App_group1
  set ipv6 next-hop 2001:100:1:1::20  (next hop is that of the firewall)

route-map IPV4_ PBR_Appgroup2 permit 10
  match ip address IPV4_App_group_1
  set ip next-hop 10.100.1.20 (next hop is that of the firewall)


interface Vlan10
! tenant SVI appgroup 1
vrf member appgroup
 ip address 10.1.1.1/24
 no ip redirect
 ipv6 address 2001:10:1:1::1/64
 no ipv6 redirects
 fabric forwarding mode anycast-gateway
ip policy route-map IPV4_ PBR_Appgroup1
ipv6 policy route-map IPV6_PBR_Appgroup1
interface Vlan20
! tenant SVI appgroup 2
vrf member appgroup
 ip address 20.1.1.1/24
 no ip redirect
 ipv6 address 2001:20:1:1::1/64
 no ipv6 redirects
 fabric forwarding mode anycast-gateway
```

```
ip policy route-map IPV4_ PBR_Appgroup2
ipv6 policy route-map IPV6_PBR_Appgroup2

On the service VTEP, the PBR policy is applied on the tenant VRF SVI. This ensures the
traffic post decapsulation will be redirected to firewall.
feature pbr

ipv6 access-list IPV6_App_group_1
10 permit ipv6 any 2001:10:1:1::0/64

ip access-list IPV4_App_group_1
10 permit ip any 10.1.1.0/24

ipv6 access-list IPV6_App_group_2
10 permit ipv6 any 2001:20:1:1::0/64

ip access-list IPV4_App_group_2
10 permit ip any 20.1.1.0/24

route-map IPV6_PBR_Appgroup1 permit 10
  match ipv6 address IPV6_App_group_2
  set ipv6 next-hop 2001:100:1:1::20  (next hop is that of the firewall)

route-map IPV6_PBR_Appgroup permit 20
  match ipv6 address IPV6_App_group1
  set ipv6 next-hop 2001:100:1:1::20  (next hop is that of the firewall)

route-map IPV4_ PBR_Appgroup permit 10
  match ip address IPV4_App_group_2
  set ip next-hop 10.100.1.20 (next hop is that of the firewall)

route-map IPV4_ PBR_Appgroup permit 20
  match ip address IPV4_App_group_1
  set ip next-hop 10.100.1.20 (next hop is that of the firewall)


interface vlan1000
!L3VNI SVI for Tenant VRF
vrf member appgroup
ip forward
ipv6 forward
ipv6 ipv6 address use-link-local-only
ip policy route-map IPV4_ PBR_Appgroup
ipv6 policy route-map IPV6_PBR_Appgroup
```

Enhanced-Policy Based Redirect (ePBR)

VXLAN PBR as a solution to selectively redirect traffic can only cater to simple traffic redirection requirements. For more complex use cases like service chaining, symmetric load-balancing, or tracking health of service appliances, usage of PBR becomes difficult. The challenge with service chaining using PBR is that it requires the user to create unique policies per node and manage the redirection rules manually across all the nodes in the chain. Also, given the stateful nature of the service nodes, the PBR rules must ensure symmetry for the reverse traffic, and this adds additional complexity to the configuration and management of the PBR policies.

Enhanced Policy-Based Redirect (ePBR) provides a comprehensive solution to insert service nodes, selectively redirect and load-balance traffic. ePBR provides a simplified workflow to create traffic chains and load-balancing rules along with providing options for probing/monitoring the health of service appliances and taking corrective action in the event of failure. ePBR is supported in both single and multi-site VXLAN EVPN deployments.



In this Figure, selective traffic originating from WAN is chained to a firewall and then the traffic is load-balanced across a cluster of proxies before forwarding toward the destination hosts. ePBR ensures symmetry is maintained for a given flow by making sure that traffic in both forward and reverse direction is redirected to the same service endpoint in the cluster of TCP proxies.

For more detailed information, guidelines and configuration examples on ePBR, see Cisco Nexus 9000 Series NX-OS ePBR Configuration Guide and Layer 4 to Layer 7 Service Redirection with Enhanced Policy-Based Redirect White Paper.

APPENDIX **D**

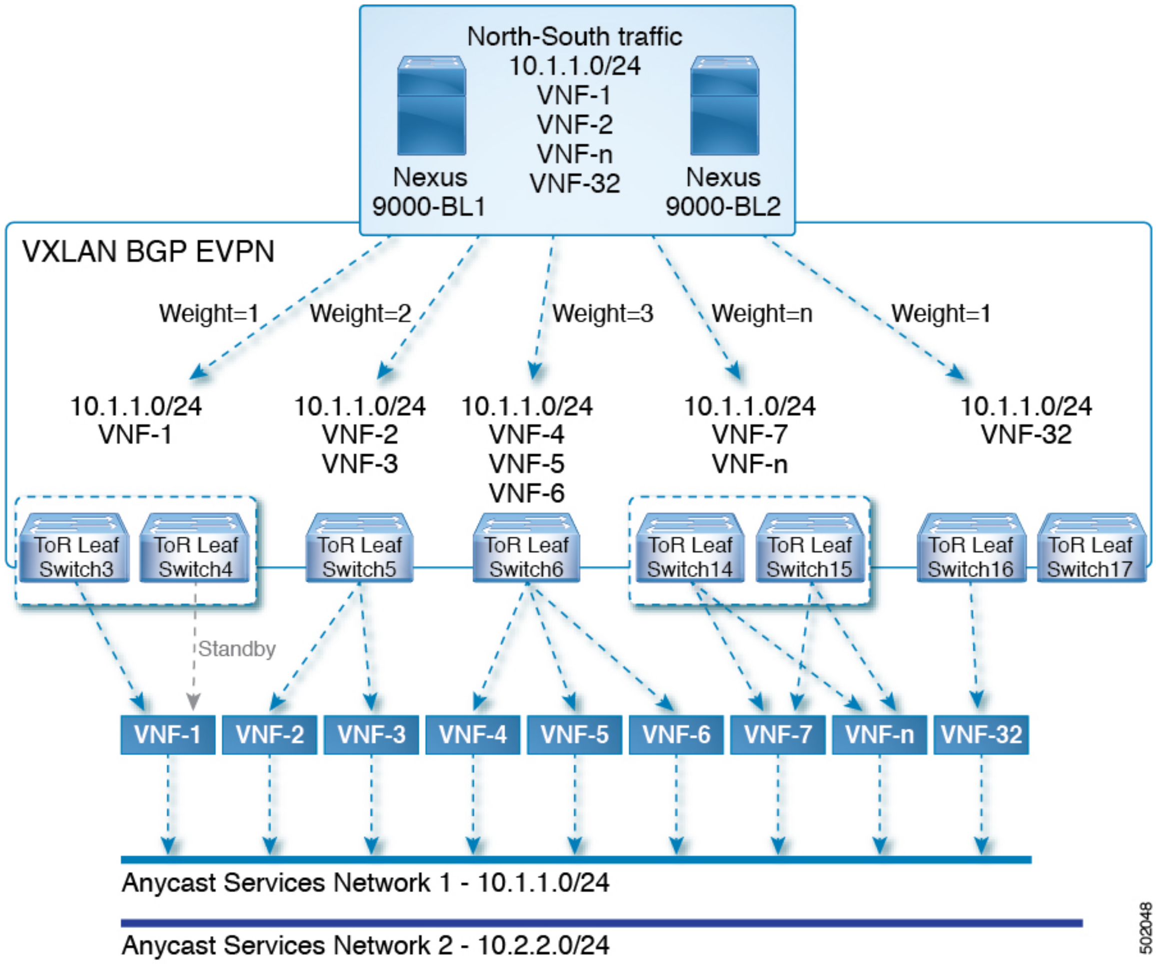
Configuring Proportional Multipath for VNF

This chapter contains the following sections:

About Proportional Multipath for VNF

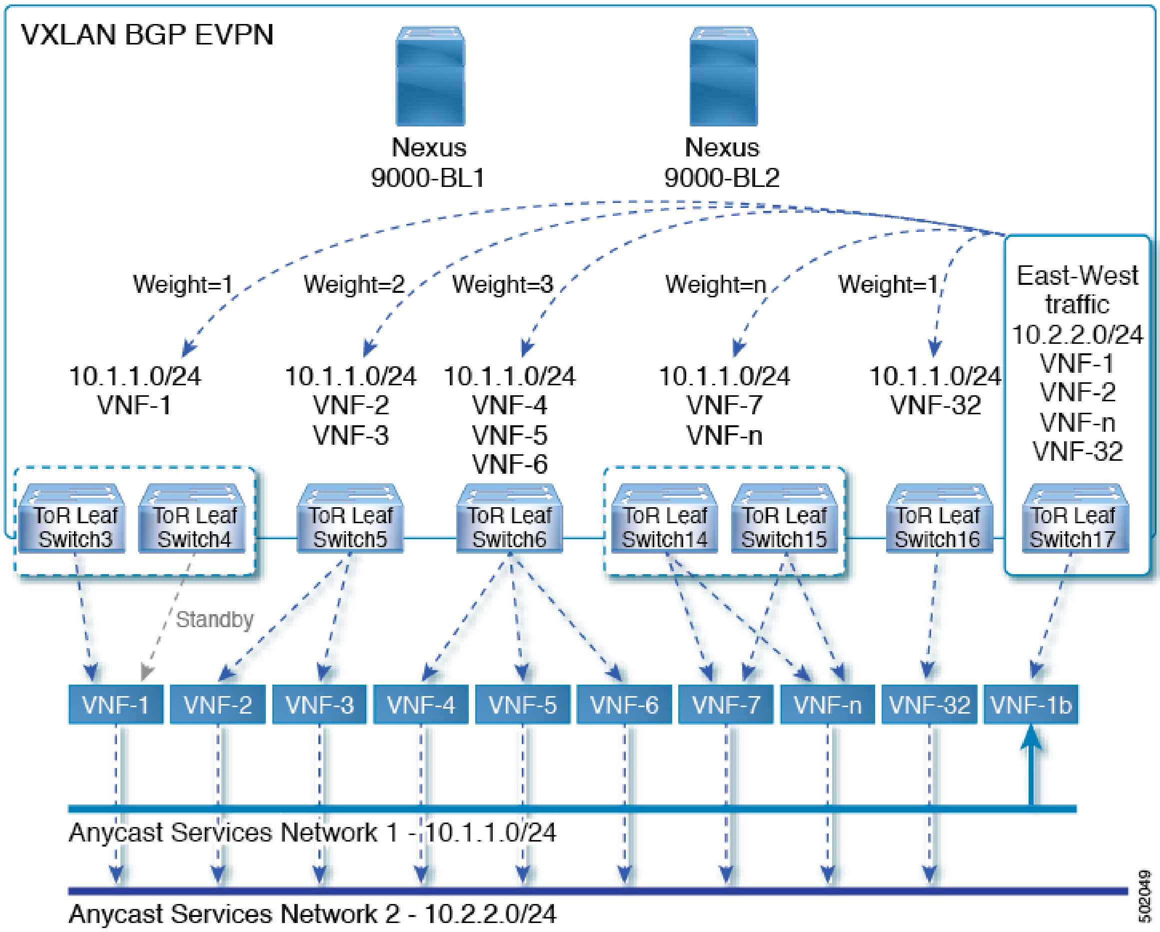
In Network Function Virtualization Infrastructures (NFVi), anycast services networks are advertised from multiple Virtual Network Functions (VNFs). The Proportional Multipath for VNF feature enables advertising of all the available next hops to a given destination network. This feature enables the switch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

Figure 54: Sample Topology (North-South Traffic)



In the preceding diagram, North-South traffic that enters the VXLAN fabric at a border leaf is sent across all egress endpoints with the traffic forwarded proportional to the number of links from the egress top of rack (ToR) to the destination network.

Figure 55: Sample Topology (East-West Traffic)



East-West traffic is forwarded between the VXLAN Tunnel Endpoints (VTEPs) proportional to the number of next hops advertised by each ToR switch to the destination network.

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

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

Advertisement of Customer Networks

Customer networks are configured statically or learned locally by using an interior gateway protocol, (IGP) or external BGP (eBGP), over a Provider Edge(PE)-Customer Edge(CE) link. These networks are redistributed into BGP and advertised to the VXLAN fabric.

The networks advertised to the ToRs by the virtual machines (VMs) attached to them are advertised to the VXLAN fabric as EVPN Type-5 routes with the following:

- The route distinguisher (RD) will be the Layer 3 VNI's configured RD.
- The gateway IP field will be populated with the next hop.
- The next hop of the EVPN route will continue to be the VTEP IP.
- The export route targets of the routes will be derived from the configured export route targets of the associated Layer 3 VNI.

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

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

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

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

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

Beginning with Cisco NX-OS Release 9.3(5), you don't need to use mixed paths. You can choose to have only eBGP or iBGP filter the ECMP paths.

When you enter the **maximum-paths mixed** command beginning with Cisco NX-OS Release 9.3(5), BGP checks for the AS-path length by default. If you want to ignore the AS-path length (for example, on nodes that participate in packet forwarding such as BGWs and VTEPs), you must enter the **bestpath as-path ignore** command. When the **maximum-paths mixed** command is enabled for earlier releases, BGP ignores the AS-path length, and URIB ignores the administrative distance when choosing ECMPs. To ensure that no impact is observed, we recommend upgrading to Cisco NX-OS Release 9.3(5) prior to entering this command.

Legacy Peer Support

Use the **advertise-gw-ip** command to advertise EVPN Type-5 routes with the gateway IP set. ToRs then advertise the gateway IP in the Type-5 NLRI. However, legacy peers running on NX-OS version older than Cisco NX-OS Release 9.2(1) can't process the gateway IP which might lead to unexpected behavior. To prevent this scenario from occurring, use the **no advertise-gw-ip** command to disable the Proportional

Multipath for VNF feature for a legacy peer. BGP sets the gateway IP field of the Type-5 NLRI to zero even if the path being advertised has a valid gateway IP.

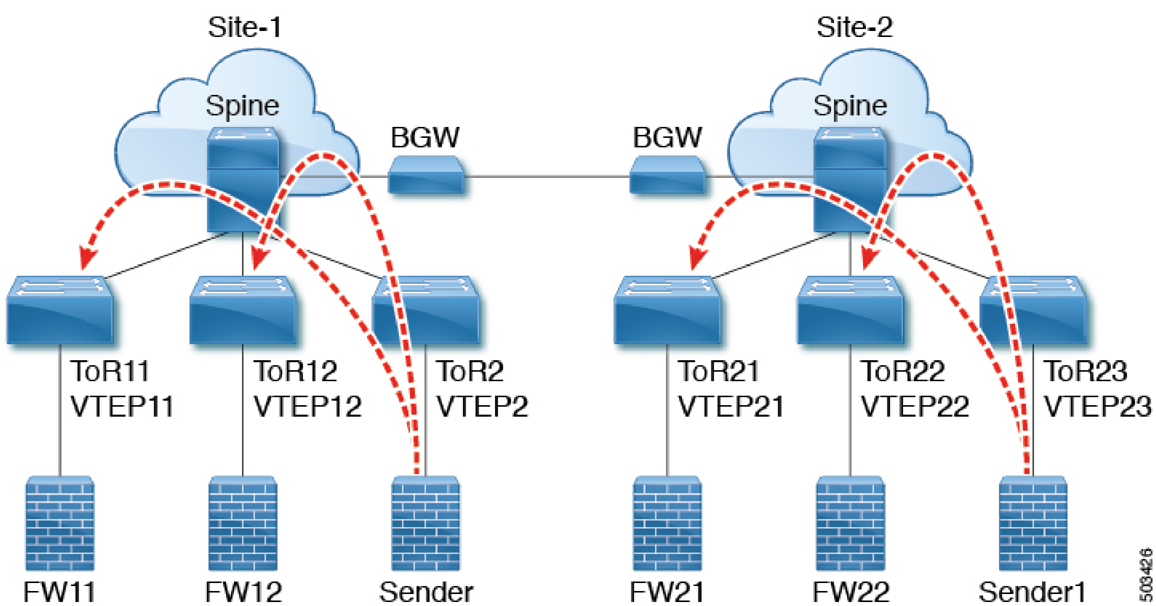
The **no advertise-gw-ip** command flaps the specified peer session as gracefully as possible. The remote peer triggers a graceful restart if the peer supports this capability. When the session is re-established, the local peer advertises EVPN Type-5 routes with the gateway IP set or with the gateway IP as zero depending on whether the **advertise-gw-ip** command has been used. By default, this knob is enabled and the gateway IP field is populated with the appropriate next hop value.

Proportional Multipath for VNF with Multi-Site

Cisco NX-OS Release 9.3(6) and later releases support Proportional Multipath for VNF with Multi-Site. This feature allows traffic to be sent across sites if a local VNF isn't available.

ToRs prefer to use local VNFs. However, if local VNFs aren't available, they can use VNFs in a different site. In the following topology, the ToRs in site 2 would use VNFs 21 and 22. However, if these VNFs aren't available, sender 1 in site 2 could send traffic to VNFs 11 and 12 in site 1.

Figure 56: VNFs in a Multi-Site Topology



To use this feature, simply configure Proportional Multipath for VNF and enable Multi-Site. For a sample configuration, see Configuration Example for Proportional Multipath for VNF with Multi-Site, on page 520.

Prerequisites for Proportional Multipath for VNF

If desired, take the following actions before upgrading to Cisco NX-OS Release 9.3(5):

- Configure a route map for redistributed paths and use the **set ip next-hop redist-unchanged** command when using locally redistributed paths to export the gateway IP address. This command preserves the next hop for locally redistributed paths. For example:

```
route-map redist-rtmap permit 10
match ip prefix-list vm-pfx-list
set ip next-hop redist-unchanged
```

- Enter the **bestpath as-path ignore** command on nodes that participate in packet forwarding, such as BGWs and VTEPs. This command causes BGP to ignore the AS-path length.

Guidelines and Limitations for Proportional Multipath for VNF

Proportional Multipath for VNF has the following guidelines and limitations:

- If the Proportional Multipath for VNF feature is enabled, maintenance mode isolation doesn't work because BGP installs all the paths in mixed multipath mode. Alternatively, a route-map is used to deny outbound BPG updates when a switch goes into maintenance mode by using user-defined profiles.
- This feature is supported for Cisco Nexus 9364C, 9300-EX, and 9300-FX/FX2/FX3 platform switches and Cisco Nexus 9500 platform switches with the N9K-C9508-FM-E2 fabric module and an -EX or -FX line card.
- This feature isn't supported on Cisco Nexus 9300-GX platform switches.
- Static and direct routes have to be redistributed into the BGP when the Proportional Multipath for VNF feature is enabled.
- If OSPF or EIGRP is being used as an IGP, routes can't be redistributed into BGP.
- If Proportional Multipath for VNF is enabled and routes aren't redistributed into BGP, asymmetric load balancing of traffic may occur as the local routes from URIB may not show up in BGP and on remote TORs as EVPN paths.
- Devices on which mixed-multipath is enabled must support the same load-balancing algorithm.
- If a VNF instance is multi-homed to multiple TORs, policies have to be configured or BGP routes have to be originated using a network command. As a result, each TOR connection to the VNF is displayed in the BGP routing table. Each TOR can now see the VNF's direct routes to the other TORs in which the VNF is multi-homed. Consequently, each TOR can advertise paths to the Gateway IPs through other TORs leading to a next hop resolution loop.

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

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

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

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

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

- The following guidelines and limitations apply to Proportional Multipath for VNF with Multi-Site:
  - This feature is supported for Cisco Nexus 9364C, 9300-EX, and 9300-FX/FX2/FX3 platform switches and Cisco Nexus 9500 platform switches with the N9K-C9508-FM-E2 fabric module and an -EX or -FX line card.
  - VNF moves across sites aren't supported.
- Proportional multipath with max-path mixed configuration is not supported for VNFs attached to vPC leaf switches. However, vPC is supported when the max-path mixed configuration is not used.
- Following guidelines and limitations are applied when a multisite Border Gateway is put into Maintenance Mode:
  - BUM Traffic from remote Fabrics will still be attracted to the Border gateway that is in maintenance mode
  - Border Gateway in maintenance mode still participates in Designated Forwarder Election
  - Default Maintenance mode profile applies the command "ip pim isolate" and so the Border gateway is isolated from S,G tree towards the fabric direction. This leads to BUM traffic loss and hence an appropriate maintenance mode profile should be used for Border Gateways than the default.

Configuring the Route Reflector

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *number*
3. **address-family l2vpn evpn**
4. **additional-paths send**
5. **additional-paths receive**
6. **additional-paths selection route-map passall**
7. **route-map passall permit** *seq-num*
8. **set path-selection all advertise**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enter global configuration mode.
Step 2	**router bgp** *number* **Example:** `switch(config)# router bgp 2`	Configure BGP.
Step 3	**address-family l2vpn evpn** **Example:** `switch(config-router)# address-family l2vpn evpn`	Configure address family Layer 2 VPN EVPN under **router bgp** context.
Step 4	**additional-paths send** **Example:** `switch(config-router-af)# additional-paths send`	The additional-paths configuration for sending..
Step 5	**additional-paths receive** **Example:** `switch(config-router-af)# additional-paths receive`	The additional-paths configuration for receiving.
Step 6	**additional-paths selection route-map passall** **Example:** `switch(config-router-af)# additional-paths selection route-map passall`	The additional-paths configuration applied the route map.
Step 7	**route-map passall permit** *seq-num* **Example:** `switch(config)# route-map passall permit 10`	Configure the route map.
Step 8	**set path-selection all advertise** **Example:** `switch(config-route-map)# set path-selection all advertise`	Sets the route-map related to the additional-paths feature.

Configuring the ToR

This procedure describes how to configure the ToR.

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *number*

3. **address-family l2vpn evpn**
4. [**no**] **maximum-paths** [*eBGP max-paths* |**mixed** | **ibgp** |**local** | **eibgp**] *mpath-count*
5. **additional-paths send**
6. **additional-paths receive**
7. **additional-paths selection route-map passall**
8. **exit**
9. **vrf evpn-tenant-1001**
10. **address-family ipv4 unicast**
11. **export-gateway-ip**
12. [**no**] **maximum-paths** [*eBGP max-paths* |**mixed** | **ibgp** |**local** | **eibgp**] *mpath-count*
13. **redistribute static route-map redist-rtmap**
14. **maximum-paths local** *number*
15. **exit**
16. **address-family ipv6 unicast**
17. **export-gateway-ip**
18. [**no**] **maximum-paths** [*eBGP max-paths* |**mixed** | **ibgp** |**local** | **eibgp**] *mpath-count*
19. **redistribute static route-map redist-rtmap**
20. **maximum-paths local** *number*
21. **exit**
22. **route-map passall permit** *seq-num*
23. **set path-selection all advertise**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enter global configuration mode.
Step 2	**router bgp** *number* **Example:** `switch(config)# router bgp 2`	Configure BGP.
Step 3	**address-family l2vpn evpn** **Example:** `switch(config-router)# address-family l2vpn evpn`	Configure address family Layer 2 VPN EVPN under **router bgp** context.

Step 4	[**no**] **maximum-paths** [*eBGP max-paths* |**mixed** | **ibgp** |**local** | **eibgp**] *mpath-count*
	Example:
	`switch(config-router-af)# maximum-paths ?`
	`  <1-64>  Number of parallel paths`
	`          *Default value is 1`
	`  eibgp   Configure multipath for both EBGP and IBGP paths`
	`  ibgp    Configure multipath for IBGP paths`
		The following options are available:
		• *eBGP max-path*–Enables the eBGP maximum paths. The range is from 1 to 64 parallel paths. The default value is 1.
		• **mixed**–Enables BGP and the Unicast Routing Information Base (URIB) to consider the following paths as Equal Cost Multi Path (ECMP):

	Command or Action	Purpose

	`local   Configure multipath for local paths`
	`mixed   Configure multipath for local and remote paths`
	`switch(config-router-af)# maximum-paths mixed 32`
	Example:
	`switch(config-router-af)# maximum-paths ibgp 32`
		• eBGP paths
		• eiBGP paths
		• iBGP paths
		• Paths from other protocols (such as static) that are redistributed or injected into BGP
		• **ibgp**–Uses iBGP to filter the ECMP paths.
		• **local**–Enables the multipath for local paths.
		• If you enter the command without the **mixed** or **ibgp** option, eBGP is used to filter the ECMP paths.
		Note Use the **no** form of this command if you want to use a single path instead of maximum paths.

Step 5	**additional-paths send** **Example:** `switch(config-router-af)# additional-paths send`	The additional-paths configuration for sending.
Step 6	**additional-paths receive** **Example:** `switch(config-router-af)# additional-paths receive`	The additional-paths configuration for receiving.
Step 7	**additional-paths selection route-map passall** **Example:** `switch(config-router-af)# additional-paths selection route-map passall`	The additional-paths configuration applied the route map.
Step 8	**exit** **Example:** `switch(config-router-af)# exit`	Exits command mode.
Step 9	**vrf evpn-tenant-1001** **Example:** `switch(config-router)# vrf evpn-tenant-1001`	Switch to the VRF configuration mode.
Step 10	**address-family ipv4 unicast** **Example:** `switch(config-router)# address-family ipv4 unicast`	Configure address family for IPv4.
Step 11	**export-gateway-ip** **Example:** `switch(config-router-vrf-af)# export-gateway-ip`	Enables BGP to advertise the gateway IP in the EVPN Type-5 routes. It exports the gateway IP for all prefixes in that VRF.

	Command or Action	Purpose

		Note If you want choose specific prefixes for which to export the gateway IP, use the following configuration instead of the **export-gateway-ip** command:
		
		route-map *name* permit *sequence*
		  match ip address prefix-list *name*
		  set evpn gateway-ip use-next-hop
		
		vrf context *vrf*
		  address-family ipv4 unicast
		    export map *name*

Step 12	[**no**] **maximum-paths** [*eBGP max-paths* |**mixed** | **ibgp** |**local** | **eibgp**] *mpath-count*
	
	Example:
	switch(config-router-vrf-af)# maximum-paths ?
	  <1-64> Number of parallel paths
	          *Default value is 1
	  eibgp Configure multipath for both EBGP and IBGP paths
	  ibgp Configure multipath for IBGP paths
	  local Configure multipath for local paths
	  mixed Configure multipath for local and remote paths
	
	switch(config-router-vrf-af)# **maximum-paths mixed 32**
	
	Example:
	switch(config-router-vrf-af)# **maximum-paths ibgp 32**
		The following options are available:
		
		• *eBGP max-path*–Enables the eBGP maximum paths. The range is from 1 to 64 parallel paths. The default value is 1.
		
		• **mixed**–Enables BGP and the Unicast Routing Information Base (URIB) to consider the following paths as Equal Cost Multi Path (ECMP):
		  • eBGP paths
		  • eiBGP paths
		  • iBGP paths
		  • Paths from other protocols (such as static) that are redistributed or injected into BGP
		
		• **ibgp**–Uses iBGP to filter the ECMP paths.
		
		• **local**–Enables the multipath for local paths.
		
		• If you enter the command without the **mixed** or **ibgp** option, eBGP is used to filter the ECMP paths.
		
		Note Use the **no** form of this command if you want to use a single path instead of maximum paths.

Step 13	**redistribute static route-map redist-rtmap** **Example:** switch(config-router-vrf-af)# **redistribute static route-map redist-rtmap**	Preserves the next-hop of the redistributed paths.
Step 14	**maximum-paths local** *number* **Example:** switch(config-router-vrf-af)# **maximum-paths local 32**	Specifies the number of local paths to be redistributed as the BGP best path for a route. The range is from 0 to 32. The default value is 1.

	Command or Action	Purpose
		Note This command isn't supported with the **maximum-paths mixed** *mpath-count* command. An error message appears if you try to configure them together. **Note** The **set ip next-hop redist-unchanged** command is required in order for the **maximum-paths local** command to work.
Step 15	**exit** **Example:** `switch(config-router-vrf-af)# exit`	Exits command mode.
Step 16	**address-family ipv6 unicast** **Example:** `switch(config-router-vrf)# address-family ipv6 unicast`	Configure address family for IPv6.

Step 17	**export-gateway-ip**
	
	Example:
	`switch(config-router-vrf-af)# export-gateway-ip`
		Enables BGP to advertise the gateway IP in the EVPN Type-5 routes. It exports the gateway IP for all prefixes in that VRF.
		
		Note If you want choose specific prefixes for which to export the gateway IP, use the following configuration instead of the **export-gateway-ip** command:
		
		`route-map name permit sequence`
		`  match ip address prefix-list name`
		`  set evpn gateway-ip use-next-hop`
		
		`vrf context vrf`
		`  address-family ipv4 unicast`
		`    export map name`

Step 18	[**no**] **maximum-paths** [*eBGP max-paths* |**mixed** | **ibgp** |**local** | **eibgp**] *mpath-count*
	
	Example:
	`switch(config-router-vrf-af)# maximum-paths ?`
	`  <1-64>  Number of parallel paths`
	`          *Default value is 1`
	`  eibgp   Configure multipath for both EBGP and IBGP paths`
	`  ibgp    Configure multipath for IBGP paths`
	`  local   Configure multipath for local paths`
	`  mixed   Configure multipath for local and remote paths`
	
	`switch(config-router-vrf-af)# maximum-paths mixed 32`
		The following options are available:
		
		• *eBGP max-path*–Enables the eBGP maximum paths. The range is from 1 to 64 parallel paths. The default value is 1.
		
		• **mixed**–Enables BGP and the Unicast Routing Information Base (URIB) to consider the following paths as Equal Cost Multi Path (ECMP):
		  • eBGP paths
		  • eiBGP paths
		  • iBGP paths

	Command or Action	Purpose
	Example: `switch(config-router-vrf-af)# maximum-paths ibgp 32`	• Paths from other protocols (such as static) that are redistributed or injected into BGP • **ibgp**–Uses iBGP to filter the ECMP paths. • **local**–Enables the multipath for local paths. • If you enter the command without the **mixed** or **ibgp** option, eBGP is used to filter the ECMP paths. **Note** Use the **no** form of this command if you want to use a single path instead of maximum paths.
Step 19	**redistribute static route-map redist-rtmap** **Example:** `switch(config-router-vrf-af)# redistribute static route-map redist-rtmap`	Preserves the next-hop of the redistributed paths.
Step 20	**maximum-paths local** *number* **Example:** `switch(config-router-vrf-af)# maximum-paths local 32`	Specifies the number of local paths to be redistributed as the BGP best path for a route. The range is from 0 to 32. The default value is 1. **Note** This command isn't supported with the **maximum-paths mixed** *mpath-count* command. An error message appears if you try to configure them together.
Step 21	**exit** **Example:** `switch(config-router-vrf-af)# exit`	Exits command mode.
Step 22	**route-map passall permit** *seq-num* **Example:** `switch(config)# route-map passall permit 10`	Configure the route map.
Step 23	**set path-selection all advertise** **Example:** `switch(config-route-map)# set path-selection all advertise`	Sets the route-map related to the additional-paths feature.

Configuring the Border Leaf

This procedure describes how to configure the border leaf.

SUMMARY STEPS

1. **configure terminal**

2. **router bgp** *number*
3. **address-family l2vpn evpn**
4. [**no**] **maximum-paths** [*eBGP max-paths* |**mixed** | **ibgp** |**local** | **eibgp**] *mpath-count*
5. **additional-paths send**
6. **additional-paths receive**
7. **additional-paths selection route-map passall**
8. **exit**
9. **vrf evpn-tenant-1001**
10. **address-family ipv4 unicast**
11. **export-gateway-ip**
12. [**no**] **maximum-paths** [*eBGP max-paths* |**mixed** | **ibgp** |**local** | **eibgp**] *mpath-count*
13. **redistribute static route-map redist-rtmap**
14. **maximum-paths local** *number*
15. **address-family ipv6 unicast**
16. **export-gateway-ip**
17. [**no**] **maximum-paths** [*eBGP max-paths* |**mixed** | **ibgp** |**local** | **eibgp**] *mpath-count*
18. **redistribute static route-map redist-rtmap**
19. **maximum-paths local** *number*
20. **exit**
21. **route-map passall permit** *seq-num*
22. **set path-selection all advertise**
23. **ip load-sharing address source-destination rotate** *rotate* **universal-id** *seed*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enter global configuration mode.
Step 2	**router bgp** *number* **Example:** `switch(config)# router bgp 2`	Configure BGP.
Step 3	**address-family l2vpn evpn** **Example:** `switch(config-router)# address-family l2vpn evpn`	Configure address family Layer 2 VPN EVPN under **router bgp** context.

Step 4	[**no**] **maximum-paths** [*eBGP max-paths* |**mixed**	**ibgp** |**local**	**eibgp**] *mpath-count*
			Example:
			`switch(config-router-af)# maximum-paths ?`
			`  <1-64>  Number of parallel paths`
				The following options are available:
				• *eBGP max-path*–Enables the eBGP maximum paths. The range is from 1 to 64 parallel paths. The default value is 1.

	Command or Action	Purpose

	`*Default value is 1`
	`eibgp   Configure multipath for both EBGP and IBGP paths`
	`ibgp    Configure multipath for IBGP paths`
	`local   Configure multipath for local paths`
	`mixed   Configure multipath for local and remote paths`
	
	`switch(config-router-af)#` **`maximum-paths mixed 32`**
	
	Example:
	
	`switch(config-router-af)#` **`maximum-paths ibgp 32`**
		• **mixed**–Enables BGP and the Unicast Routing Information Base (URIB) to consider the following paths as Equal Cost Multi Path (ECMP):
		  • eBGP paths
		  • eiBGP paths
		  • iBGP paths
		  • Paths from other protocols (such as static) that are redistributed or injected into BGP
		• **ibgp**–Uses iBGP to filter the ECMP paths.
		• **local**–Enables the multipath for local paths.
		• If you enter the command without the **mixed** or **ibgp** option, eBGP is used to filter the ECMP paths.
		
		Note Use the **no** form of this command if you want to use a single path instead of maximum paths.

Step 5	**additional-paths send** **Example:** `switch(config-router-af)#` **`additional-paths send`**	The additional-paths configuration for sending.
Step 6	**additional-paths receive** **Example:** `switch(config-router-af)#` **`additional-paths receive`**	The additional-paths configuration for receiving.
Step 7	**additional-paths selection route-map passall** **Example:** `switch(config-router-af)#` **`additional-paths selection route-map passall`**	The additional-paths configuration enables the additional-paths feature.
Step 8	**exit** **Example:** `switch(config-router-af)#` **`exit`**	Exits command mode.
Step 9	**vrf evpn-tenant-1001** **Example:** `switch(config-router)#` **`vrf evpn-tenant-1001`**	Switch to the VRF configuration mode.
Step 10	**address-family ipv4 unicast** **Example:** `switch(config-router)#` **`address-family ipv4 unicast`**	Configure address family for IPv4.

<table>
<tr><th></th><th>Command or Action</th><th>Purpose</th></tr>
<tr><td>Step 11</td><td>export-gateway-ip
Example:
<code>switch(config-router-vrf-af)# export-gateway-ip</code></td><td>Enables BGP to advertise the gateway IP in the EVPN Type-5 routes. It exports the gateway IP for all prefixes in that VRF.
Note If you want choose specific prefixes for which to export the gateway IP, use the following configuration instead of the export-gateway-ip command:<pre>route-map name permit sequence
  match ip address prefix-list name
  set evpn gateway-ip use-next-hop

vrf context vrf
  address-family ipv4 unicast
    export map name</pre></td></tr>
<tr><td>Step 12</td><td>[no] maximum-paths [eBGP max-paths |mixed | ibgp |local | eibgp] mpath-count
Example:<pre>switch(config-router-af)# maximum-paths ?
  <1-64> Number of parallel paths
          *Default value is 1
  eibgp   Configure multipath for both EBGP and
IBGP paths
  ibgp   Configure multipath for IBGP paths
  local   Configure multipath for local paths
  mixed   Configure multipath for local and remote
paths</pre><code>switch(config-router-vrf-af)# maximum-paths mixed 32</code>
Example:
<code>switch(config-router-vrf-af)# maximum-paths ibgp 32</code></td><td>The following options are available:
eBGP max-path–Enables the eBGP maximum paths. The range is from 1 to 64 parallel paths. The default value is 1.
mixed–Enables BGP and the Unicast Routing Information Base (URIB) to consider the following paths as Equal Cost Multi Path (ECMP):
eBGP paths
eiBGP paths
iBGP paths
Paths from other protocols (such as static) that are redistributed or injected into BGP
ibgp–Uses iBGP to filter the ECMP paths.
local–Enables the multipath for local paths.
If you enter the command without the mixed or ibgp option, eBGP is used to filter the ECMP paths.
Note Use the no form of this command if you want to use a single path instead of maximum paths.</td></tr>
<tr><td>Step 13</td><td>redistribute static route-map redist-rtmap
Example:
<code>switch(config-router-vrf-af)# redistribute static route-map redist-rtmap</code></td><td>Preserves the next-hop of the redistributed paths.</td></tr>
</table>

<table>
<tr><th></th><th>Command or Action</th><th>Purpose</th></tr>
<tr><td>Step 14</td><td>maximum-paths local number
Example:
<code>switch(config-router-vrf-af)# maximum-paths local 32</code></td><td>Specifies the number of local paths to be redistributed as the BGP best path for a route. The range is from 0 to 32. The default value is 1.
Note This command isn't supported with the maximum-paths mixed mpath-count command. An error message appears if you try to configure them together.</td></tr>
<tr><td>Step 15</td><td>address-family ipv6 unicast
Example:
<code>switch(config-router-vrf)# address-family ipv6 unicast</code></td><td>Configure address family for IPv6.</td></tr>
<tr><td>Step 16</td><td>export-gateway-ip
Example:
<code>switch(config-router-vrf-af)# export-gateway-ip</code></td><td>Enables BGP to advertise the gateway IP in the EVPN Type-5 routes. It exports the gateway IP for all prefixes in that VRF.
Note If you want choose specific prefixes for which to export the gateway IP, use the following configuration instead of the export-gateway-ip command:
<pre>route-map name permit sequence
  match ip address prefix-list name
  set evpn gateway-ip use-next-hop

vrf context vrf
  address-family ipv4 unicast
    export map name</pre></td></tr>
<tr><td>Step 17</td><td>[no] maximum-paths [eBGP max-paths |mixed | ibgp |local | eibgp] mpath-count
Example:
<pre>switch(config-router-vrf-af)# maximum-paths ?
  <1-64> Number of parallel paths
          *Default value is 1
  eibgp   Configure multipath for both EBGP and
IBGP paths
  ibgp   Configure multipath for IBGP paths
  local   Configure multipath for local paths
 mixed   Configure multipath for local and remote
 paths</pre><code>switch(config-router-vrf-af)# maximum-paths mixed 32</code>
Example:
<code>switch(config-router-vrf-af)# maximum-paths ibgp 32</code></td><td>The following options are available:
eBGP max-path–Enables the eBGP maximum paths. The range is from 1 to 64 parallel paths. The default value is 1.
mixed–Enables BGP and the Unicast Routing Information Base (URIB) to consider the following paths as Equal Cost Multi Path (ECMP):
eBGP paths
eiBGP paths
iBGP paths
Paths from other protocols (such as static) that are redistributed or injected into BGP
ibgp–Uses iBGP to filter the ECMP paths.
local–Enables the multipath for local paths.</td></tr>
</table>

	Command or Action	Purpose
		• If you enter the command without the **mixed** or **ibgp** option, eBGP is used to filter the ECMP paths. **Note** Use the **no** form of this command if you want to use a single path instead of maximum paths.
Step 18	**redistribute static route-map redist-rtmap** **Example:** `switch(config-router-vrf-af)# redistribute static route-map redist-rtmap`	Preserves the next-hop of the redistributed paths.
Step 19	**maximum-paths local** *number* **Example:** `switch(config-router-vrf-af)# maximum-paths local 32`	Specifies the number of local paths to be redistributed as the BGP best path for a route. The range is from 0 to 32. The default value is 1. **Note** This command isn't supported with the **maximum-paths mixed** *mpath-count* command. An error message appears if you try to configure them together.
Step 20	**exit** **Example:** `switch(config-router-vrf-af)# exit`	Exits command mode.
Step 21	**route-map passall permit** *seq-num* **Example:** `switch(config)# route-map passall permit 10`	Configure the route map.
Step 22	**set path-selection all advertise** **Example:** `switch(config-route-map)# set path-selection all advertise`	Sets the route-map related to the additional-paths feature.
Step 23	**ip load-sharing address source-destination rotate** *rotate* **universal-id** *seed* **Example:** `ip load-sharing address source-destination rotate 32 universal-id 1`	Configures the unicast FIB load-sharing algorithm for data traffic. • The **universal-id** option sets the random seed for the hash algorithm and shifts the flow from one link to another. You do not need to configure the universal ID. Cisco NX-OS chooses the Universal ID if you do not configure it. The *seed* range is from 1 to 4294967295. • The **rotate** option causes the hash algorithm to rotate the link picking selection so that it does not continually choose the same link across all nodes in the network. It does so by influencing the bit pattern for the hash algorithm. This option shifts the flow

	Command or Action	Purpose
		from one link to another and load balances the already load-balanced (polarized) traffic from the first ECMP level across multiple links. If you specify a **rotate** value, the 64-bit stream is interpreted starting from that bit position in a cyclic rotation. The **rotate** range is from 1 to 63, and the default is 32. **Note** With multi-tier Layer 3 topology, polarization is possible. To avoid polarization, use a different rotate bit at each tier of the topology. **Note** To configure a rotation value for port channels, use the **port-channel load-balance src-dst ip-l4port rotate** *rotate* command. For more information on this command, see the *Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide, Release 9.x*.

Configuring the BGP Legacy Peer

If you are running a Cisco Nexus Release prior to 9.2(1), follow this procedure to disable sending the gateway IP address to that peer.

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *number*
3. **neighbor** *address* **remote-as** *number*
4. **address-family l2vpn evpn**
5. **no advertise-gw-ip**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enter global configuration mode.
Step 2	**router bgp** *number* **Example:** `switch(config)# router bgp 2000000`	Configure BGP.

	Command or Action	Purpose
Step 3	**neighbor** *address* **remote-as** *number* **Example:** `switch(config-router)# neighbor 8.8.8.8 remote-as 2000000`	Define neighbor.
Step 4	**address-family l2vpn evpn** **Example:** `switch(config-router-neighbor)# address-family l2vpn evpn`	Configure address family Layer 2 VPN EVPN.
Step 5	**no advertise-gw-ip** **Example:** `switch(config-router-neighbor-af)# no advertise-gw-ip`	Disables the BGP EVPN Mixed-path and Proportional Layer-3 Multipath feature for a legacy peer.

Configuring a User-Defined Profile for Maintenance Mode

SUMMARY STEPS

1. **configure terminal**
2. **configure maintenance profile maintenance-mode**
3. **route-map** *name* **deny** *sequence*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enter global configuration mode.
Step 2	**configure maintenance profile maintenance-mode** **Example:** `switch(config)# configure maintenance profile maintenance-mode`	Configure maintenance mode profile.
Step 3	**route-map** *name* **deny** *sequence* **Example:** `switch(config-mm-profile)# route-map GIR deny 5`	Configure route map. The value of *sequence* is from 0 to 65535. Default is 10.

Configuring a User-Defined Profile for Normal Mode

SUMMARY STEPS

1. **configure terminal**
2. **configure maintenance profile normal-mode**
3. **route-map** *name* **permit** *sequence*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enter global configuration mode.
Step 2	**configure maintenance profile normal-mode** **Example:** `switch(config)# configure maintenance profile normal-mode`	Configure maintenance mode.
Step 3	**route-map** *name* **permit** *sequence* **Example:** `switch(config-mm-profile)# route-map GIR permit 5`	Configure route map. The value of *sequence* is from 0 to 65535. Default is 10.

Configuring a Default Route Map

SUMMARY STEPS

1. **configure terminal**
2. **route-map** *name* **permit** *sequence*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enter global configuration mode.
Step 2	**route-map** *name* **permit** *sequence* **Example:** `switch(config-mm-profile)# route-map GIR permit 5`	Configure route map. The value of *sequence* is from 0 to 65535. Default is 10.

Applying a Route Map to a Route Reflector

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *number*
3. **neighbor** *ip-address*
4. **address-family l2vpn evpn**
5. **route-map** *name* **out**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enter global configuration mode.
Step 2	**router bgp** *number* **Example:** `switch(config)# router bgp 2`	Configure BGP.
Step 3	**neighbor** *ip-address* **Example:** `switch(config-router)# neighbor 10.1.1.1`	Configure the IP address of a BGP neighbor which is the route reflector. *ip-address* can be an IPv4 or IPv6 address or prefix.
Step 4	**address-family l2vpn evpn** **Example:** `switch(config-router-neighbor)# address-family l2vpn evpn`	Configure a Layer 2 VPN EVPN address family.
Step 5	**route-map** *name* **out** **Example:** `switch(config-router-neighbor-af)# route-map GIR out`	Apply the route map to the neighbor route reflector.

Verifying Proportional Multipath for VNF

Command	Purpose
show bgp ipv4 unicast	Displays Border Gateway Protocol (BGP) information for the IPv4 unicast address family.

Command	Purpose
show bgp l2vpn evpn	Displays BGP information for the Layer-2 Virtual Private Network (L2VPN) Ethernet Virtual Private Network (EVPN) address family.
show ip route	Displays routes from the unicast RIB.
show maintenance profile maintenance-mode	Displays the GIR user-defined profile for the maintenance mode.
show maintenance profile normal-mode	Displays the GIR user-defined profile for the normal mode.

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

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

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

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

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

  Path-id 1 not advertised to any peer

Route Distinguisher: 4.4.4.4:3    (L3VNI 22001)       //   Local L3VNI
```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

The following example shows how to display the GIR user-defined profile for the maintenance mode:

```
switch# show maintenance profile maintenance-mode
  [Maintenance Mode]
  ip pim isolate
  router bgp 2
   isolate
  router isis 1
```

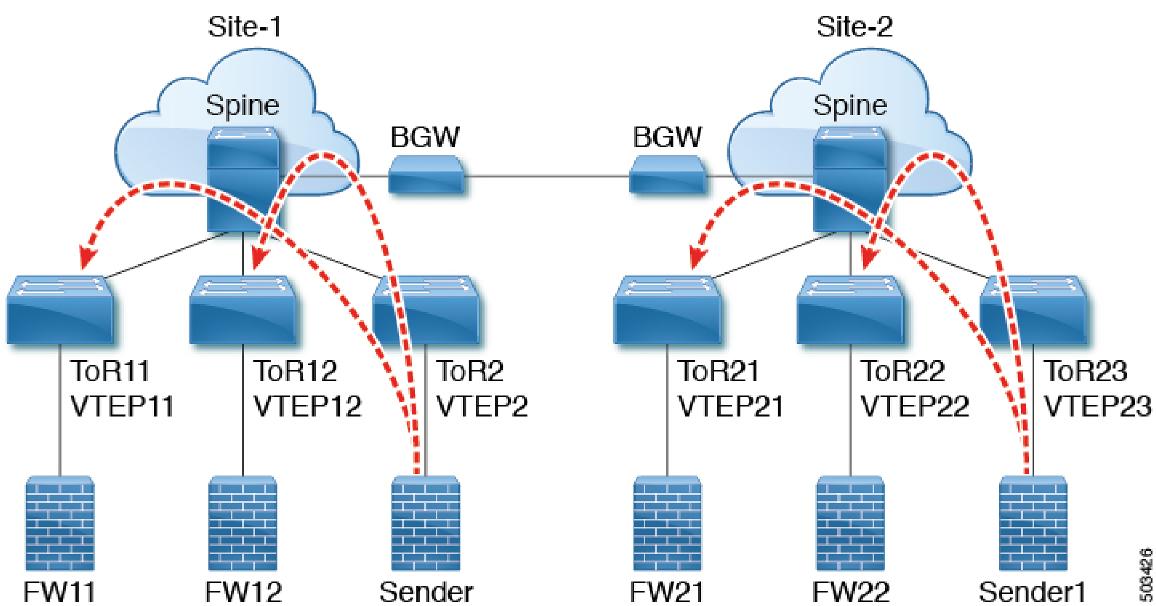
```
   isolate
  route-map GIR deny 5
```

The following example shows how to display the GIR user-defined profile for the normal mode:

```
switch# show maintenance profile normal-mode
  [Normal Mode]
  no ip pim isolate
  router bgp 2
   no isolate
  router isis 1
   no isolate
  route-map GIR permit 5
```

Configuration Example for Proportional Multipath for VNF with Multi-Site

Figure 57: VNFs in a Multi-Site Topology



The following configuration example allows traffic to be sent across sites if a local VNF isn't available.

```
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 bfd
feature nv overlay

no password strength-check
```

```
username admin password 5 password role network-admin
ip domain-lookup
copp profile strict
evpn multisite border-gateway 1
  delay-restore time 30
snmp-server user admin network-admin auth md5 0x66a8185ad28d9df13d9214f6e19aad37 priv
0x66a8185ad28d9df13d9214f6e19aad37 localizedkey

fabric forwarding anycast-gateway-mac 0000.2222.3333
ip pim ssm range 232.0.0.0/8
vlan 1,14,24,100-110,120-150,1000-1010,1100-1110,2000-2010,2100-2110,3000-3010
vlan 100
  name l2-vni-vlan-0-for-vrf100
  vn-segment 2000100
vlan 101
  name l2-vni-vlan-0-for-vrf101
  vn-segment 2000101
vlan 1100
  name l2-vni-vlan-1-for-vrf100
  vn-segment 2001100
vlan 1101
  name l2-vni-vlan-1-for-vrf101
  vn-segment 2001101
vlan 2100
  name l3-vni-vlan-for-vrf100
  vn-segment 3000100
vlan 2101
  name l3-vni-vlan-for-vrf101
  vn-segment 3000101

route-map passall permit 10
  set path-selection all advertise
route-map permit-all permit 10
  set path-selection all advertise
route-map permit-all-v6 permit 10

vrf context vrf100
  vni 3000100
  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
vrf context vrf101
 vni 3000101
  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

interface Vlan14
  no shutdown
  vrf member vrf100
  ip address 192.14.0.1/24
  ipv6 address 192:14::1/64

interface Vlan24
  no shutdown
  vrf member vrf101
```

```
  ip address 192.24.0.1/24
  ipv6 address 192:24::1/64

interface Vlan100
  description "L3VRF.VLANNUM.0.222"
  no shutdown
  vrf member vrf100
  ip address 100.0.0.222/24
  ipv6 address 100::222/64
  fabric forwarding mode anycast-gateway

interface Vlan101
  description "L3VRF.VLANNUM.0.222"
  no shutdown
  vrf member vrf101
  ip address 101.0.0.222/24
  ipv6 address 101::222/64
  fabric forwarding mode anycast-gateway

interface Vlan1100
  description "L3VRF.VLANNUM.0.222"
  no shutdown
  vrf member vrf100
  ip address 100.1.0.222/16
  ipv6 address 100:1::222/64
  fabric forwarding mode anycast-gateway

interface Vlan1101
  description "L3VRF.VLANNUM.0.222"
  no shutdown
  vrf member vrf101
  ip address 101.1.0.222/16
  ipv6 address 101:1::222/64
  fabric forwarding mode anycast-gateway

interface Vlan2100
  no shutdown
  vrf member vrf100
  ip forward
  ipv6 address use-link-local-only

interface Vlan2101
  no shutdown
  vrf member vrf101
  ip forward
  ipv6 address use-link-local-only

interface nve1
  no shutdown
  host-reachability protocol bgp
  source-interface loopback1
  multisite border-gateway interface loopback2
  member vni 2000100-2000110
    suppress-arp
    mcast-group 227.1.1.1
  member vni 2000120-2000150
    suppress-arp
    mcast-group 227.1.1.1
  member vni 2001100-2001110
    suppress-arp
    mcast-group 227.1.1.1
  member vni 3000100-3000110 associate-vrf
  member vni 3100100-3100110 associate-vrf
```

```
interface Ethernet1/22
  description "BGW11 to BGW2"
  medium p2p
  ip unnumbered loopback0
  ip ospf cost 40
  ip ospf network point-to-point
  ip router ospf 12 area 0.0.0.0
  no shutdown
  evpn multisite dci-tracking

interface Ethernet1/25
  description "BGW11 to Spine11"
  medium p2p
  ip unnumbered loopback0
  ip ospf cost 40
  ip ospf network point-to-point
  ip router ospf 1 area 0.0.0.0
  no shutdown
  evpn multisite fabric-tracking

interface Ethernet1/27
  description "BGW11 to Spine12"
  medium p2p
  ip unnumbered loopback0
  ip ospf cost 40
  ip ospf network point-to-point
  ip router ospf 1 area 0.0.0.0
  no shutdown
  evpn multisite fabric-tracking

interface Ethernet1/34
  switchport
  switchport mode trunk
  switchport trunk allowed vlan 14,24
  no shutdown

interface loopback0
  ip address 1.1.11.0/32
  ip router ospf 1 area 0.0.0.0
  ip pim sparse-mode

interface loopback1
  ip address 1.1.11.1/32
  ip router ospf 1 area 0.0.0.0
  ip pim sparse-mode

interface loopback2
  ip address 11.11.11.11/32
  ip router ospf 12 area 0.0.0.0
  ip pim sparse-mode

router ospf 1
  redistribute direct route-map permit-all
router ospf 12
  redistribute direct route-map permit-all
ip load-sharing address source-destination rotate 32 universal-id 1

router bgp 1
  log-neighbor-changes
  address-family l2vpn evpn
    maximum-paths 8
    maximum-paths ibgp 8
    additional-paths send
    additional-paths receive
```

```
      additional-paths selection route-map passall
    neighbor 1.2.11.1
      remote-as 1
      description "SPINE-11"
      update-source loopback1
      address-family l2vpn evpn
        send-community extended
    neighbor 1.2.12.1
      remote-as 1
      description "SPINE-12"
      update-source loopback1
      address-family l2vpn evpn
        send-community extended
    neighbor 2.1.2.1
      remote-as 2
      description "BGW-2"
      update-source loopback1
      ebgp-multihop 3
      peer-type fabric-external
      address-family ipv4 unicast
      address-family l2vpn evpn
        send-community extended
        rewrite-evpn-rt-asn
    vrf vrf100
      address-family ipv4 unicast
        redistribute direct route-map permit-all
        maximum-paths 8
        maximum-paths ibgp 8
        export-gateway-ip
      address-family ipv6 unicast
        redistribute direct route-map permit-all
        maximum-paths 8
        maximum-paths ibgp 8
        export-gateway-ip
    vrf vrf101
      address-family ipv4 unicast
        redistribute direct route-map permit-all
        maximum-paths 8
        maximum-paths ibgp 8
        export-gateway-ip
      address-family ipv6 unicast
        redistribute direct route-map permit-all
        maximum-paths 8
        maximum-paths ibgp 8
        export-gateway-ip
  evpn
    vni 2000100 l2
      rd auto
      route-target import auto
      route-target export auto
    vni 2000101 l2
      rd auto
      route-target import auto
      route-target export auto
    vni 2001100 l2
      rd auto
      route-target import auto
      route-target export auto
    vni 2001101 l2
      rd auto
      route-target import auto
      route-target export auto
```

The following example shows that the VTEP in site 1 prefers the local VNF (FW).

```
leaf1# show bgp l2vpn evpn 200.100.1.1
BGP routing table information for VRF default, address family L2VPN EVPN
Route Distinguisher: 1.3.12.0:3
BGP routing table entry for [5]:[0]:[0]:[32]:[200.100.1.1]/224, version 77902
Paths: (4 available, best #2)
Flags: (0x000002) (high32 00000000) on xmit-list, is not in l2rib/evpn, is not in HW
Multipath: eBGP iBGP Local

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

  Gateway IP: 100.0.0.12
  AS-Path: 99 100 , path sourced external to AS
    1.3.12.1 (metric 81) from 1.2.12.1 (1.2.12.0)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 3000100
      Received path-id 2
      Extcommunity: RT:1:3000100 ENCAP:8 Router MAC:00be.7547.13bf
      Originator: 1.3.12.0 Cluster list: 1.2.12.0

  Advertised path-id 2
  Path type: local, path is valid, not best reason: Locally originated, multipath, no labeled
 nexthop
  Gateway IP: 100.0.0.11
  AS-Path: 99 100 , path sourced external to AS
    1.3.11.1 (metric 0) from 0.0.0.0 (1.3.11.0)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 3000100
      Received path-id 1
      Extcommunity: RT:1:3000100 ENCAP:8 Router MAC:d478.9bb3.c1a1
```

The following example shows how the local VNF is disabled so that the VNF from site 2 is used. The BGP adjacency is shut down between site 1's VTEP11 to FW11 and between VTEP12 to FW12.

```
leaf1(config-router)# vrf vrf100
leaf1(config-router-vrf)# neighbor 100::11
leaf1(config-router-vrf-neighbor)# shut
leaf1(config-router-vrf-neighbor)# neighbor 100::12
leaf1(config-router-vrf-neighbor)# shut
leaf1(config-router-vrf-neighbor)# neighbor 100:1::11
leaf1(config-router-vrf-neighbor)# shut
leaf1(config-router-vrf-neighbor)# neighbor 100:1::12
leaf1(config-router-vrf-neighbor)# shut
leaf1(config-router-vrf-neighbor)# neighbor 100.0.0.11
leaf1(config-router-vrf-neighbor)# shut
leaf1(config-router-vrf-neighbor)# neighbor 100.0.0.12
leaf1(config-router-vrf-neighbor)# shut
leaf1(config-router-vrf-neighbor)# neighbor 100.1.0.11
leaf1(config-router-vrf-neighbor)# shut
leaf1(config-router-vrf-neighbor)# neighbor 100.1.0.12
leaf1(config-router-vrf-neighbor)# shut
leaf1(config-router-vrf-neighbor)# end
```

The following example shows that the prefix now uses the VNF (FW) from site 2.

```
leaf1# show bgp l2vpn evpn 200.100.1.1
BGP routing table information for VRF default, address family L2VPN EVPN
Route Distinguisher: 1:3000100
BGP routing table entry for [5]:[0]:[0]:[32]:[200.100.1.1]/224, version 97269
Paths: (3 available, best #3)
Flags: (0x000002) (high32 00000000) on xmit-list, is not in l2rib/evpn, is not in HW
Multipath: eBGP iBGP Local
```

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

Gateway IP: 100.1.0.21
AS-Path: 2 99 100 , path sourced external to AS
  11.11.11.11 (metric 20) from 1.2.12.1 (1.2.12.0)
    Origin IGP, MED 2000, localpref 100, weight 0
    Received label 3000100
    Received path-id 2
    Extcommunity: RT:1:3000100 SOO:03030100:00000000 ENCAP:8
        Router MAC:0200.0b0b.0b0b
    Originator: 1.1.12.0 Cluster list: 1.2.12.0
```

INDEX

P

Q

R

S

T

U

V

W