CONTENTS

PREFACE
 Preface xi
 Audience xi
 Document Conventions xi
 Related Documentation for Cisco Nexus 9000 Series Switches xii
 Documentation Feedback xii
 Communications, Services, and Additional Information xii

CHAPTER 1
 New and Changed Information 1
 New and Changed Information 1

CHAPTER 2
 Overview 7
 VXLAN Overview 7
 VXLAN Encapsulation and Packet Format 7
 VXLAN Tunnel Endpoint 8
 VXLAN Packet Forwarding Flow 8
 Cisco Nexus 9000 as Hardware-Based VXLAN Gateway 8
 vPC Consistency Check for vPC VTEPs 9
 Static Ingress Replication 10
 Bud Node Topology 10
 VXLAN BGP EVPN Control Plane 11

CHAPTER 3
 Configuring VXLAN 13
 Information About VXLAN 13
 Guidelines and Limitations for VXLAN 13
 Considerations for VXLAN Deployment 19
 vPC Considerations for VXLAN Deployment 21
Network Considerations for VXLANDeployments 24
Considerations for the Transport Network 25
Configuring VXLAN 26
Enabling VXLANs 26
Mapping VLAN to VXLAN VNI 26
Guidelines and Limitations for Port VLAN Mapping 27
Configuring Port VLAN Mapping on a Trunk Port 27
Configuring Inner VLAN and Outer VLAN Mapping on a Trunk Port 31
Creating and Configuring an NVE Interface and Associate VNIs 33
Configuring Static MAC for VXLAN VTEP 33
Disabling VXLANs 34
Configuring BGP EVPN Ingress Replication 34
Configuring Static Ingress Replication 35
Guidelines and Limitations for Q-in-VNI 35
Configuring Q-in-VNI 36
Configuring Selective Q-in-VNI 39
Configuring Q-in-VNI with LACP Tunneling 41
Configuring QinQ-QinVNI 43
Overview for QinQ-QinVNI 43
Guidelines and Limitations for QinQ-QinVNI 44
Configuring QinQ-QinVNI 44
Removing a VNI 45
Configuring FHRP Over VXLAN 46
Overview for FHRP Over VXLAN 46
Guidelines and Limitations for FHRP Over VXLAN 46
Only Supported Deployments for FHRP Over VXLAN 47
New Supported Topology for Configuring FHRP Over VXLAN 48
Configuring IGMP Snooping Over VXLAN 50
Overview of IGMP Snooping Over VXLAN 50
Guidelines and Limitations for IGMP Snooping Over VXLAN 50
Configuring IGMP Snooping Over VXLAN 50
Configuring Line Cards for VXLAN 51
Centralized VRF Route Leaking using Default-Routes and Aggregates 52
Overview 52
CHAPTER 9

Configuring Tenant Routed Multicast 165

About Tenant Routed Multicast 165

About Tenant Routed Multicast Mixed Mode 167

Guidelines and Limitations for Tenant Routed Multicast 167

Guidelines and Limitations for Layer 3 Tenant Routed Multicast 168

Guidelines and Limitations for Layer 2/Layer 3 Tenant Routed Multicast (Mixed Mode) 168

Rendezvous Point for Tenant Routed Multicast 169

Configuring a Rendezvous Point for Tenant Routed Multicast 169

Configuring a Rendezvous Point Inside the VXLAN Fabric 170

Configuring an External Rendezvous Point 171

Configuring RP Everywhere with PIM Anycast 173

Configuring a TRM Leaf Node for RP Everywhere with PIM Anycast 173

Configuring a TRM Border Leaf Node for RP Everywhere with PIM Anycast 174

Configuring an External Router for RP Everywhere with PIM Anycast 176

Configuring RP Everywhere with MSDP Peering 178

Configuring a TRM Leaf Node for RP Everywhere with MSDP Peering 179

Configuring a TRM Border Leaf Node for RP Everywhere with MSDP Peering 180

Configuring an External Router for RP Everywhere with MSDP Peering 182

Configuring Layer 3 Tenant Routed Multicast 184

Configuring TRM on the VXLAN EVPN Spine 188

Configuring Tenant Routed Multicast in Layer 2/Layer 3 Mixed Mode 190

Configuring Layer 2 Tenant Routed Multicast 195

CHAPTER 10

Configuring VXLAN QoS 197

Information About VXLAN QoS 197

VXLAN QoS Terminology 197

VXLAN QoS Features 199

Trust Boundaries 199

Classification 199

Marking 199

Policing 199

Queuing and Scheduling 200

Traffic Shaping 200
Preface

This preface includes the following sections:

- Audience, on page xi
- Document Conventions, on page xi
- Related Documentation for Cisco Nexus 9000 Series Switches, on page xii
- Documentation Feedback, on page xii
- Communications, Services, and Additional Information, on page xii

Audience

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

Document Conventions

Command descriptions use the following conventions:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>bold</strong></td>
<td>Bold text indicates the commands and keywords that you enter literally as shown.</td>
</tr>
<tr>
<td><em>Italic</em></td>
<td>Italic text indicates arguments for which you supply the values.</td>
</tr>
<tr>
<td>[x]</td>
<td>Square brackets enclose an optional element (keyword or argument).</td>
</tr>
<tr>
<td>[x</td>
<td>y]</td>
</tr>
<tr>
<td>{x</td>
<td>y}</td>
</tr>
<tr>
<td>[x {y</td>
<td>z}]</td>
</tr>
<tr>
<td>Convention</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>variable</td>
<td>Indicates a variable for which you supply values, in context where italics can cannot be used.</td>
</tr>
<tr>
<td>string</td>
<td>A nonquoted set of characters. Do not use quotation marks around the string or the string includes the quotation marks.</td>
</tr>
</tbody>
</table>

Examples use the following conventions:

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

**Related Documentation for Cisco Nexus 9000 Series Switches**

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


**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 Marketplace.
- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.
Cisco Bug Search Tool

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

- New and Changed Information, on page 1

New and Changed Information

This table summarizes the new and changed features for the *Cisco Nexus 9000 Series NX-OS VXLAN Configuration Guide* and where they are documented.

**Table 1: New and Changed Features**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>Changed in Release</th>
<th>Where Documented</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port VLAN Routing</td>
<td>Added support for Cisco Nexus 9300-EX, 9300-FX, and 9300-FX2.</td>
<td>7.0(3)I7(5)</td>
<td>Configuring Port VLAN Mapping on a Trunk Port, on page 27</td>
</tr>
<tr>
<td>RP Everywhere</td>
<td></td>
<td>7.0(3)I7(5)</td>
<td>Rendezvous Point for Tenant Routed Multicast, on page 169</td>
</tr>
<tr>
<td>VXLAN Tunnel Egress QoS Policy</td>
<td>Added support for VXLAN Tunnel Egress QoS Policy</td>
<td>7.0(3)I7(5)</td>
<td>About VXLAN Tunnel Egress QoS Policy, on page 57</td>
</tr>
<tr>
<td>Switching and routing on overlapped VLAN interfaces</td>
<td>Added support for the Nexus 9300, and 9500 switches.</td>
<td>7.0(3)I7(4)</td>
<td>Configuring VXLAN, on page 13</td>
</tr>
<tr>
<td>NGOAM Support</td>
<td>Added support for the Cisco Nexus 9336C-FX, 93300YCYC-FX, and 93240YCYC-FX2Z switches.</td>
<td>7.0(3)I7(4)</td>
<td>Configuring VXLAN OAM, on page 115</td>
</tr>
<tr>
<td>Configuring VXLAN IGMP Snooping</td>
<td>Added support for configuring VXLAN IGMP snooping on Cisco Nexus 9508 switches with 9636-RX line cards.</td>
<td>7.0(3)I7(4)</td>
<td>Guidelines and Limitations for VXLAN, on page 13</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
<td>Changed in Release</td>
<td>Where Documented</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
<td>-------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>FC/FCoE NPV coexistence with VXLAN</td>
<td>FCoE N-port Virtualization (NPV) can co-exist with VXLAN.</td>
<td>7.0(3)I7(3)</td>
<td>Guidelines and Limitations for VXLAN, on page 13</td>
</tr>
<tr>
<td>VXLAN on Cisco Nexus 9348GC-FXP switch.</td>
<td>VXLAN supported added for the Cisco Nexus 9348GC-FXP switch.</td>
<td>7.0(3)I7(3)</td>
<td>Guidelines and Limitations for VXLAN, on page 13</td>
</tr>
<tr>
<td>QinQ-QinVNI</td>
<td>Tunneling feature that allows configuring a trunk port as a multi-tag port.</td>
<td>7.0(3)I7(3)</td>
<td>Overview for QinQ-QinVNI, on page 43</td>
</tr>
<tr>
<td>IP Unnumbered</td>
<td>IP Unnumbered support has been added for VXLAN.</td>
<td>7.0(3)I7(2)</td>
<td>IP Unnumbered</td>
</tr>
<tr>
<td>Tenant Routed Multicast + VXLAN EVPN Multi-Site</td>
<td>Both features can co-exist on the same physical switch.</td>
<td>7.0(3)I7(2)</td>
<td>Configuring Tenant Routed Multicast, on page 165</td>
</tr>
<tr>
<td>Tenant Routed Multicast + VXLAN EVPN Multi-Site</td>
<td>Both features can co-exist on the same physical switch.</td>
<td>7.0(3)F3(3)</td>
<td>Configuring Tenant Routed Multicast, on page 165</td>
</tr>
<tr>
<td>VXLAN L2/L3 GW</td>
<td>Support added for: · PIM/ASM · ARP Suppression · IPv6 in the Overlay</td>
<td>7.0(3)F3(3)</td>
<td>Configuring VXLAN, on page 13</td>
</tr>
<tr>
<td>PBR over VXLAN</td>
<td>Policy Based Routing over Virtual Extensible LAN</td>
<td>7.0(3)I7(1)</td>
<td>Configuring Policy-Based Routing</td>
</tr>
<tr>
<td>Centralized VRF Route Leaks</td>
<td>Centralized VRF Route Leaking using Default-Routes and Aggregates</td>
<td>7.0(3)I7(1)</td>
<td>Centralized VRF Route Leaking using Default-Routes and Aggregates, on page 52</td>
</tr>
<tr>
<td>Tenant Routed Multicast</td>
<td>Tenant Routing Multicast (TRM) enables multicast forwarding on the VXLAN fabric using BGP-based EVPN control pane. TRM supports Layer 2 and Layer 3 multicast for sender and receivers on the same or different VTEPs in a tenant VRF.</td>
<td>7.0(3)I7(1)</td>
<td>Configuring Tenant Routed Multicast, on page 165</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
<td>Changed in Release</td>
<td>Where Documented</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
<td>--------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>VIP/PIP</td>
<td>Advertises type-5 routes using the primary IP address of the VTEP interfaces as the next hop address in the VXLAN EVPN fabric.</td>
<td>7.0(3)I7(1)</td>
<td>Configuring VIP/PIP, on page 153</td>
</tr>
<tr>
<td>VXLAN EVPN Multi-Site</td>
<td>The VXLAN EVPN Multi-Site feature is a solution to interconnect two or more BGP-based Ethernet VPN (EVPN) site's fabrics in a scalable fashion over an IP-only network.</td>
<td>7.0(3)I7(1)</td>
<td>Configuring VXLAN EVPN Multi-Site, on page 157</td>
</tr>
<tr>
<td>Configuring new CLI command <code>lacp vpc-convergence</code></td>
<td>Added support for configuring CLI command <code>lacp vpc-convergence</code> for better convergence of Layer 2 EVPN VXLAN.</td>
<td>7.0(3)I6(1)</td>
<td>Guidelines and Limitations for VXLAN, on page 13</td>
</tr>
<tr>
<td>Configuring port-VLAN with VXLAN</td>
<td>Added support for configuring port-VLAN with VXLAN on Cisco Nexus 9300-EX and 9500 Series switches with 9700-EX line cards.</td>
<td>7.0(3)I6(1)</td>
<td>Guidelines and Limitations for VXLAN, on page 13</td>
</tr>
<tr>
<td>Configuring VXLAN on Cisco Nexus 3232C and 3264Q switches</td>
<td>Added support for configuring VXLAN on Cisco Nexus 3232C and 3264Q switches.</td>
<td>7.0(3)I6(1)</td>
<td>Guidelines and Limitations for VXLAN, on page 13</td>
</tr>
<tr>
<td>Configuring NGOAM Authentication</td>
<td>Added support for configuring NGOAM authentication.</td>
<td>7.0(3)I6(1)</td>
<td>NGOAM Authentication, on page 123</td>
</tr>
<tr>
<td>Configuring VXLAN EVPN Multihoming, VLAN Consistency Checking, and ESI ARP Suppression</td>
<td>Added support for VXLAN EVPN Multihoming, VLAN Consistency Checking, and ESI ARP suppression.</td>
<td>7.0(3)I5(2)</td>
<td>Introduction to Multihoming, on page 125, Overview of VLAN Consistency Checking, on page 147, and Overview of ESI ARP Suppression, on page 150</td>
</tr>
<tr>
<td>Configuring Selective Q-in-VNI</td>
<td>Added support for configuring selective Q-in-VNI.</td>
<td>7.0(3)I5(2)</td>
<td>Configuring Selective Q-in-VNI, on page 39</td>
</tr>
<tr>
<td>Configuring FHRP over VXLAN</td>
<td>Added support for configuring FHRP over VXLAN on the Cisco Nexus 9200, 9300, and 9300-EX Series switches.</td>
<td>7.0(3)I5(2)</td>
<td>Guidelines and Limitations for FHRP Over VXLAN, on page 46</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
<td>Changed in Release</td>
<td>Where Documented</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------------------</td>
<td>-------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Configuring VXLAN IGMP Snooping</td>
<td>Added support for configuring VXLAN IGMP snooping on Cisco Nexus 9300 Series switches and Cisco Nexus 9500 Series switches with N9K-X9732C-EX line cards.</td>
<td>7.0(3)I5(2)</td>
<td>Guidelines and Limitations for IGMP Snooping Over VXLAN, on page 50</td>
</tr>
<tr>
<td>Configuring VRRP (FHRP) over VXLAN</td>
<td>Added support for configuring VRRP (FHRP) over VXLAN.</td>
<td>7.0(3)I5(1)</td>
<td>Configuring FHRP Over VXLAN, on page 46</td>
</tr>
<tr>
<td>Configuring VXLAN OAM</td>
<td>Added support for configuring VXLAN OAM.</td>
<td>7.0(3)I5(1)</td>
<td>VXLAN OAM Overview, on page 115</td>
</tr>
<tr>
<td>Configuring IGMP Snooping over VXLAN</td>
<td>Added support for configuring IGMP Snooping over VXLAN.</td>
<td>7.0(3)I5(1)</td>
<td>Configuring IGMP Snooping Over VXLAN, on page 50</td>
</tr>
<tr>
<td>VXLAN</td>
<td>Added support for the Cisco Nexus 93108TC-EX and 93180YC-EX switches and the X9732C-EX line card.</td>
<td>7.0(3)I4(2)</td>
<td>Guidelines and Limitations for VXLAN, on page 13</td>
</tr>
<tr>
<td>Support for <code>suppress mac-route</code> command</td>
<td>Added support for <code>suppress mac-route</code> command.</td>
<td>7.0(3)I4(1)</td>
<td>Commands for BGP EVPN</td>
</tr>
</tbody>
</table>
| Support for In Service Software Upgrade (ISSU) | Added support for In Service Software Upgrade (ISSU).                                                                                                                                                                                                                                                                                        | 7.0(3)I4(1)        | Guidelines and Limitations for VXLAN                   
| Tracking route support          | Added support for displaying tracking route information.                                                                                                                                                                                                                                                                                     | 7.0(3)I2(2)        | Verifying the VXLAN Configuration                                                                   |
| LACP tunneling support for VXLAN | Added support for VXLAN with LACP tunneling.                                                                                                                                                                                                                                                                                                   | 7.0(3)I2(2)        | Configuring Q-in-VNI with LACP Tunneling                                                           |
| VXLAN FEX HIF support           | Added VXLAN support for FEX host interface port.                                                                                                                                                                                                                                                                                              | 7.0(3)I2(1)        | Guidelines and Limitations for VXLAN                   
<p>| Flood and Learn and centralized gateway support | Added recommendation for centralized gateway                                                                                                                                                                                                                                                                                                 | 7.0(3)I2(1)        | Considerations for VXLAN Deployment                   |
| PV Routing support              | Added support for PV routing.                                                                                                                                                                                                                                                                                                                 | 7.0(3)I2(1)        | Configuring Port VLAN Mapping on a Trunk Port                                                        |
| VXLAN Bud Node Over VPC         | Example configuration of VXLAN bud node over VPC.                                                                                                                                                                                                                                                                                            | 7.0(3)I2(1)        | VXLAN Bud Node Over VPC Overview                                                                   |</p>
<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>Changed in Release</th>
<th>Where Documented</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGP EVPN DHCP Relay support</td>
<td>Enables DHCP relay support in BGP EVPN environment.</td>
<td>7.0(3)I2(1)</td>
<td>DHCP Relay in VXLAN BGP EVPN</td>
</tr>
<tr>
<td>Q-in-VNI support</td>
<td>Added support for Q-in-VNI.</td>
<td>7.0(3)I2(1)</td>
<td>Configuring Q-in-VNI</td>
</tr>
<tr>
<td>Inner VLAN and Outer VLAN Mapping on a Trunk Port support</td>
<td>Added support for inner VLAN and outer VLAN mapping on a trunk port.</td>
<td>7.0(3)I2(1)</td>
<td>Configuring Inner VLAN and Outer VLAN Mapping on a Trunk Port</td>
</tr>
<tr>
<td>VXLAN EVPN ingress replication</td>
<td>Replicates BUM traffic to remote VTEP peers that are learned through the BGP EVPN control plane on Cisco Nexus 9300 Series switches.</td>
<td>7.0(3)I1(2)</td>
<td>Configuring VXLAN EVPN Ingress Replication</td>
</tr>
<tr>
<td>Port VLAN mapping on a trunk port</td>
<td>Enables VLAN translation between the ingress VLAN and a local VLAN on a port on Cisco Nexus 9300 Series switches.</td>
<td>7.0(3)I1(2)</td>
<td>Configuring Port VLAN Mapping on a Trunk Port</td>
</tr>
<tr>
<td>vPC Consistency Check for vPC VTEPs</td>
<td>Enables two switches configured as a vPC pair to exchange and verify their configuration compatibility.</td>
<td>7.0(3)I1(2)</td>
<td>vPC Consistency Check for vPC VTEPs</td>
</tr>
<tr>
<td>Static MAC for VXLAN VTEP support</td>
<td>Enables the configuration of static MAC addresses behind a peer VTEP on Cisco Nexus 9300 Series switches.</td>
<td>7.0(3)I1(2)</td>
<td>Configuring Static MAC for VXLAN VTEP</td>
</tr>
<tr>
<td>VXLAN BGP EVPN support</td>
<td>Added support for Cisco Nexus 9500 Series switches.</td>
<td>7.0(3)I1(2)</td>
<td>Configuring VXLAN BGP EVPN</td>
</tr>
<tr>
<td>Ingress replication support</td>
<td>Enables the replication of BUM traffic to remote peers.</td>
<td>7.0(3)I1(1)</td>
<td>Configuring Static Ingress Replication</td>
</tr>
<tr>
<td>Bud node topology support</td>
<td>Enables a VXLAN VTEP device to also be an IP transit device.</td>
<td>7.0(3)I1(1)</td>
<td>Bud Node Topology</td>
</tr>
<tr>
<td>VXLAN BGP EVPN support</td>
<td>Enables the learning of remote VTEPs, overlay MACs, and routes through the BGP EVPN control plane protocol on Cisco Nexus 9300 Series switches.</td>
<td>7.0(3)I1(1)</td>
<td>Configuring VXLAN BGP EVPN</td>
</tr>
</tbody>
</table>
Overview

This chapter contains the following sections:

• VXLAN Overview, on page 7

VXLAN Overview

Cisco Nexus 9000 switches are designed for hardware-based VXLAN function. It provides Layer 2 connectivity extension across the Layer 3 boundary and integrates between VXLAN and non-VXLAN infrastructures. This can enable virtualized and multitenant data center designs over a shared common physical infrastructure.

VXLAN provides a way to extend Layer 2 networks across Layer 3 infrastructure using MAC-in-UDP encapsulation and tunneling. VXLAN enables flexible workload placements using the Layer 2 extension. It can also be an approach to building a multitenant data center by decoupling tenant Layer 2 segments from the shared transport network.

When deployed as a VXLAN gateway, Cisco Nexus 9000 switches can connect VXLAN and classic VLAN segments to create a common forwarding domain so that tenant devices can reside in both environments.

VXLAN has the following benefits:

• Flexible placement of multitenant segments throughout the data center.
  It provides a way to extend Layer 2 segments over the underlying shared network infrastructure so that tenant workloads can be placed across physical pods in the data center.

• Higher scalability to address more Layer 2 segments.
  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.)

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

VXLAN Encapsulation and Packet Format

VXLAN is a Layer 2 overlay scheme over a Layer 3 network. It uses 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 goes in 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 VNID, VXLAN can support 16 million LAN segments.

**VXLAN Tunnel Endpoint**

VXLAN uses VXLAN tunnel endpoint (VTEP) devices to map tenants’ end devices to VXLAN segments and to perform VXLAN encapsulation and de-encapsulation. Each VTEP function has two interfaces: One is a switch interface on the local LAN segment to support local endpoint communication through bridging, and the other is an IP interface to the transport IP network.

The IP interface has a unique IP address that identifies the VTEP device on the transport IP network known as the infrastructure VLAN. The VTEP device uses this IP address to encapsulate Ethernet frames and transmits the encapsulated packets to the transport network through the IP interface. A VTEP device also discovers the remote VTEPs for its VXLAN segments and learns remote MAC Address-to-VTEP mappings through its IP interface.

The VXLAN segments are independent of the underlying network topology; conversely, the underlying IP network between VTEPs is independent of the VXLAN overlay. It routes the encapsulated packets based on the outer IP address header, which has the initiating VTEP as the source IP address and the terminating VTEP as the destination IP address.

**VXLAN Packet Forwarding Flow**

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

**Cisco Nexus 9000 as Hardware-Based VXLAN Gateway**

VXLAN is a new technology for virtual data center overlays and is being adopted in data center networks more and more, especially for virtual networking in the hypervisor for virtual machine-to-virtual machine communication. However, data centers are likely to contain devices that are not capable of supporting VXLAN, such as legacy hypervisors, physical servers, and network services appliances, such as physical firewalls and load balancers, and storage devices, etc. Those devices need to continue to reside on classic VLAN segments. It is not uncommon that virtual machines in a VXLAN segment need to access services provided by devices in a classic VLAN segment. This type of VXLAN-to-VLAN connectivity is enabled by using a VXLAN gateway.

A VXLAN gateway is a VTEP device that combines a VXLAN segment and a classic VLAN segment into one common Layer 2 domain.

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 eliminates the need for an additional physical or virtual
device to be the gateway. The hardware-based encapsulation and de-encapsulation provides line-rate performance for all frame sizes.

**vPC Consistency Check for vPC VTEPs**

The vPC consistency check is a mechanism used by the two switches configured as a vPC pair to exchange and verify their configuration compatibility. Consistency checks are performed to ensure that NVE configurations and VN-Segment configurations are identical across vPC peers. This check is essential for the correct operation of vPC functions.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>vPC Check Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VLAN-VNI mapping</td>
<td>Type-1-nongraceful</td>
<td>Brings down the affected VLANs on vPC ports on both sides.</td>
</tr>
<tr>
<td>VTEP-Member-VNI</td>
<td>Type-1-nongraceful</td>
<td>Member VNIs must be the same on both nodes. VNIs that are not common bring down the corresponding VLANs on vPC ports on both sides. (The attributes considered are mcast group address, suppress-arp, and Layer 3 VRF VNI.)</td>
</tr>
<tr>
<td>VTEP-emulated IP</td>
<td>Type-1-nongraceful</td>
<td>If an emulated IP address is not the same on both nodes, all gateway vPC ports on one side (secondary) are brought down. Alternatively, one side of all vPC ports is brought down. The VTEP source loopback on the vPC secondary is also brought down if the emulated IP address is not the same on both sides.</td>
</tr>
<tr>
<td>NVE Oper State</td>
<td>Type-1-nongraceful</td>
<td>The NVE needs to be in the oper UP state on both sides for the vPC consistency check. If both VTEPs are not in the OPER_UP state, the secondary leg is brought down along with the VTEP source loopback on the vPC secondary.</td>
</tr>
<tr>
<td>NVE Host-Reachability Protocol</td>
<td>Type-1-nongraceful</td>
<td>The vPC on both sides must be configured with the same host-reachability protocol. Otherwise, the secondary leg is brought down along with the VTEP source loopback on the vPC secondary.</td>
</tr>
</tbody>
</table>

VLAN-to-VXLAN VN-segment mapping is a type-1 consistency check parameter. The two VTEP switches are required to have identical mappings. VLANs that have mismatched VN-segment mappings will be suspended. When the graceful consistency check is disabled and problematic VLANs arise, the primary vPC switch and the secondary vPC switch will suspend the VLANs.

The following situations are detected as inconsistencies:

- One switch has a VLAN mapped to a VN-segment (VXLAN VNI), and the other switch does not have a mapping for the same VLAN.

- The two switches have a VLAN mapped to different VN-segments.
Beginning with 7.0(3)I1(2), each VXLAN VNI must have the same configuration. However, when configuring with ingress-replication protocol static, the list of static peer IP addresses are not checked as part of the consistency check.

The following is an example of displaying vPC information:

```
sys06-tor3# sh vpc consistency-parameters global
```

Legend:

Type 1 : vPC will be suspended in case of mismatch

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Local Value</th>
<th>Peer Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vlan to Vn-segment Map</td>
<td>1</td>
<td>1024 Relevant Map(s)</td>
<td>1024 Relevant Map(s)</td>
</tr>
<tr>
<td>STP Mode</td>
<td>1</td>
<td>MST</td>
<td>MST</td>
</tr>
<tr>
<td>STP Disabled</td>
<td>1</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>STP MST Region Name</td>
<td>1</td>
<td>&quot;&quot;</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>STP MST Region Revision</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>VLAN Mapping</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STP Loopguard</td>
<td>1</td>
<td>Disabled</td>
<td>Disabled</td>
</tr>
<tr>
<td>STP Bridge Assurance</td>
<td>1</td>
<td>Enabled</td>
<td>Enabled</td>
</tr>
<tr>
<td>STP Port Type, Edge BPDUFilter, Edge BPDUGuard</td>
<td>1</td>
<td>Normal, Disabled, Disabled</td>
<td>Normal, Disabled, Disabled</td>
</tr>
<tr>
<td>STP MST Simulate PVST</td>
<td>1</td>
<td>Enabled</td>
<td>Enabled</td>
</tr>
<tr>
<td>Nve Oper State, Secondary</td>
<td>1</td>
<td>Up, 4.4.4.4</td>
<td>Up, 4.4.4.4</td>
</tr>
<tr>
<td>Nve Vni Configuration</td>
<td>1</td>
<td>10002-11025</td>
<td>10002-11025</td>
</tr>
<tr>
<td>Allowed VLANS</td>
<td>-</td>
<td>1-1025</td>
<td>1-1025</td>
</tr>
<tr>
<td>Local suspended VLANS</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

### Static Ingress Replication

VXLAN uses flooding and dynamic MAC address learning to transport broadcast, unknown unicast, and multicast traffic. VXLAN forwards these traffic types using a multicast forwarding tree or ingress replication.

With static ingress replication:

- Remote peers are statically configured.
- Multi-destination packets are unicast encapsulated and delivered to each of the statically configured remote peers.

### Note

Cisco NX-OS supports multiple remote peers in one segment and also allows the same remote peer in multiple segments.

### Bud Node Topology

A bud node is a device that is a VXLAN VTEP device and at the same time it is an IP transit device for the same multicast group used for VXLAN VNIs. In the figure, multicast group 239.0.0.1 is used for VXLAN VNIs. For VXLAN multicast encapsulated traffic from Host-1 to Host-2, VTEP-1 performs a multicast
reverse-path forwarding (RPF) check in group 239.0.0.1 and then VXLAN decapsulation. For VXLAN multicast encapsulated traffic from Host-1 to Host-3 using the same group 239.0.0.1, VTEP-1 is an IP transit device for the multicast packets. It performs an RPF check and IP forwarding based on the outer IP header that has 239.0.0.1 as the destination. When these two different roles collide on the same device, the device becomes a bud node.

The Cisco Nexus 9000 Series switches provide support for the bud node topology. The application leaf engine (ALE) of the device enables it to be a VXLAN VTEP device and an IP transit device at the same time so the device can become a bud node.

*Figure 1: VXLAN Bud-Node Topology*

---

**Note**

The bud node topology is not supported when SVI uplinks exist in the configuration.

---

**Note**

For bud-node topologies, the source IP of the VTEP behind VPC must be in the same subnet as the infra-VLAN.

## VXLAN BGP EVPN Control Plane

A Cisco Nexus Series Switch can be configured to provide a BGP ethernet VPN (EVPN) control plane using a distributed anycast gateway, with Layer 2 and Layer 3 VxLAN overlay networks.

For a data center network, a BGP EVPN control plane provides:

- Flexible workload placement that is not restricted with physical topology of the data center network.
  - Virtual machines may be placed anywhere in the data center, without considerations of physical boundaries of racks.

- Optimal east-west traffic between servers within and across data centers
  - East west traffic between servers/virtual machines is achieved by most specific routing at the first hop router, where the first hop routing is done at the access layer. Host routes must be exchanged to ensure most specific routing to and from servers/hosts. Virtual machine mobility is supported via detecting of virtual machine attachment and signaling new location to rest of the network.
• Eliminate or reduce flooding in the data center.
  • Flooding is reduced by distributing MAC reachability information via BGP EVPN to optimize flooding relating to L2 unknown unicast traffic. Optimization of reducing broadcasts associated with ARP/IPv6 Neighbor solicitation is achieved via distributing the necessary information via BGP EVPN and caching it at the access switches, address solicitation request can then locally responded without sending a broadcast.

• Standards based control plane that can be deployed independent of a specific fabric controller.
  • The BGP 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 (BGP 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.

**Note**

Distributed anycast gateway refers to the use of anycast gateway addressing and an overlay network to provide a distributed control plane that governs the forwarding of frames within and across a L3 core network. The distributed anycast gateway functionality will be used to facilitate flexible workload placement, and optimal traffic across the L3 core network. The overlay network that will be used is based on VXLAN.
CHAPTER 3

Configuring VXLAN

This chapter contains the following sections:

• Information About VXLAN, on page 13
• Configuring VXLAN, on page 26
• VXLAN Tunnel Egress QoS Policy, on page 57
• Verifying the VXLAN Configuration, on page 59
• Example of VXLAN Bridging Configuration, on page 61

Information About VXLAN

Guidelines and Limitations for VXLAN

VXLAN has the following guidelines and limitations:

Table 2: ACL Options That can be used for VXLAN Traffic, on Platforms That Include, Cisco Nexus 92300YC, 92160YC-X, 93120TX, 9332PQ, and 9348GC-FXP Switches

<table>
<thead>
<tr>
<th>ACL Direction</th>
<th>ACL Type</th>
<th>VTEP Type</th>
<th>Port Type</th>
<th>Flow Direction</th>
<th>Traffic Type</th>
<th>Supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACL Direction</td>
<td>ACL Type</td>
<td>VTEP Type</td>
<td>Port Type</td>
<td>Flow Direction</td>
<td>Traffic Type</td>
<td>Supported</td>
</tr>
<tr>
<td>---------------</td>
<td>----------</td>
<td>-----------</td>
<td>-----------</td>
<td>----------------</td>
<td>--------------</td>
<td>-----------</td>
</tr>
<tr>
<td>Egress</td>
<td>RACL</td>
<td>Ingress VTEP</td>
<td>Uplink L3/L3-PO/SVI</td>
<td>Access to Network [GROUP:encap direction]</td>
<td>VXLAN encap [GROUP:outer]</td>
<td>NO</td>
</tr>
<tr>
<td>Ingress</td>
<td>RACL</td>
<td>Egress VTEP</td>
<td>Uplink L3/L3-PO/SVI</td>
<td>Network to Access [GROUP:decap direction]</td>
<td>VXLAN encap [GROUP:outer]</td>
<td>NO</td>
</tr>
<tr>
<td>Egress</td>
<td>PACL</td>
<td>Egress VTEP</td>
<td>L2 port</td>
<td>Network to Access [GROUP:decap direction]</td>
<td>Native L2 traffic [GROUP:inner]</td>
<td>NO</td>
</tr>
<tr>
<td>VACL</td>
<td>Egress VTEP</td>
<td>VLAN</td>
<td>Network to Access [GROUP:decap direction]</td>
<td>Native L2 traffic [GROUP:inner]</td>
<td>NO</td>
<td></td>
</tr>
</tbody>
</table>

Table 3: ACL Options That can be used for VXLAN Traffic, on Platforms that Include, Cisco Nexus 92160YC-X, 93108TC-EX, 93180LC-EX, and 93180YC-EX Switches, Release 7.0(3)I6(1)

<table>
<thead>
<tr>
<th>ACL Direction</th>
<th>ACL Type</th>
<th>VTEP Type</th>
<th>Port Type</th>
<th>Flow Direction</th>
<th>Traffic Type</th>
<th>Supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ingress</td>
<td>PACL</td>
<td>Ingress VTEP</td>
<td>L2 port</td>
<td>Access to Network [GROUP:encap direction]</td>
<td>Native L2 traffic [GROUP:inner]</td>
<td>YES (works only for base port PO)</td>
</tr>
<tr>
<td>Egress</td>
<td>PACL</td>
<td>Egress VTEP</td>
<td>L2 port</td>
<td>Network to Access [GROUP:decap direction]</td>
<td>Native L2 traffic [GROUP:inner]</td>
<td>NO</td>
</tr>
<tr>
<td>ACL Direction</td>
<td>ACL Type</td>
<td>VTEP Type</td>
<td>Port Type</td>
<td>Flow Direction</td>
<td>Traffic Type</td>
<td>Supported</td>
</tr>
<tr>
<td>---------------</td>
<td>----------</td>
<td>-----------</td>
<td>-----------</td>
<td>----------------</td>
<td>--------------</td>
<td>-----------</td>
</tr>
<tr>
<td>Ingress</td>
<td>RACL</td>
<td>Ingress VTEP</td>
<td>Tenant L3 SVI</td>
<td>Access to Network [GROUP:encap direction]</td>
<td>Native L3 traffic [GROUP:inner]</td>
<td></td>
</tr>
<tr>
<td>Egress</td>
<td>RACL</td>
<td>Egress VTEP</td>
<td>Tenant L3 SVI</td>
<td>Network to Access [GROUP:decap direction]</td>
<td>Post-decap L3 traffic [GROUP:inner]</td>
<td></td>
</tr>
<tr>
<td>Ingress</td>
<td>RACL</td>
<td>Egress VTEP</td>
<td>Uplink L3/L3-PO/SVI</td>
<td>Network to Access [GROUP:decap direction]</td>
<td>VXLAN encap [GROUP:outer]</td>
<td>NO</td>
</tr>
<tr>
<td>Egress</td>
<td>RACL</td>
<td>Ingress VTEP</td>
<td>Uplink L3/L3-PO/SVI</td>
<td>Access to Network [GROUP:encap direction]</td>
<td>VXLAN encap [GROUP:outer]</td>
<td>NO</td>
</tr>
</tbody>
</table>

- The `lacp vpc-convergence` command can be configured in VXLAN and non-VXLAN environments that have vPC port channels to hosts that support LACP.

- 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.

- Beginning with Cisco NX-OS Release 7.0(3)I7(3), Fibre Channel over Ethernet (FCoE) N-port virtualization (NPV) can co-exist with VXLAN on different fabric uplinks but on same or different front panel ports on the Cisco Nexus 93180YC-EX and 93180YC-FX switches.

  Fibre Channel N-port virtualization (NPV) can co-exist with VXLAN on different fabric uplinks but on same or different front panel ports on the Cisco Nexus 93180YC-FX switches. VXLAN can only exist on the Ethernet front panel ports, but not on the FC front panel ports.

- Beginning with Cisco NX-OS Release 7.0(3)I7(3), VXLAN is supported on the Cisco Nexus 9348GC-FXP switch.

- 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 is not applicable to the Cisco Nexus 9200 and 9300-EX platform switches and Cisco Nexus 9500 platform switches with 9700-EX line cards.

- For information about the `load-share` keyword usage for the PBR with VXLAN feature, see the Guidelines and Limitations section of the Configuring Policy-Based Routing chapter of the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide, Release 7.x.

- For Cisco NX-OS Release 7.0(3)F3(3) the following features are not supported:
  - VXLAN with vPC is not supported.
  - DHCP snooping, ACL, and QoS policies are not supported on VXLAN VLANs.
- IGMP snooping is not supported on VXLAN enabled VLANs.

- Beginning with Cisco NX-OS Release 7.0(3)F3(3), 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.

- Beginning with Cisco NX-OS Release 7.0(3)F3(3), if VXLAN is enabled, the Layer 2 Gateway cannot be enabled when there is any line card other than the 9636C-RX.

- Beginning with Cisco NX-OS Release 7.0(3)F3(3), PIM/ASM is supported in the underlay ports. PIM-BiDir is not supported. For more information, see the Cisco Nexus 9000 Series NX_OSMulticast Routing Configuration Guide, Release 7.x.

- Beginning with Cisco NX-OS Release 7.0(3)F3(3), IPv6 hosts routing in the overlay is supported.

- Beginning with Cisco NX-OS Release 7.0(3)F3(3), ARP suppression is supported.

- Beginning with Cisco NX-OS Release 7.0(3)I7(1), the 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_OUS Unicast Routing Configuration Guide, Release 7.x.

- Beginning with Cisco NX-OS Release 7.0(3)I6(1), a new CLI command `lacp vpc-convergence` is added for better convergence of Layer 2 EVPN VXLAN:

```plaintext
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
```

- Beginning with Cisco NX-OS Release 7.0(3)I6(1), 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.

- Beginning with Cisco NX-OS Release 7.0(3)I6(1), VXLAN is supported on Cisco Nexus 3232C and 3264Q switches. Cisco Nexus 3232C and 3264Q switches do not support inter-VNI routing.

  IGMP snooping on VXLAN enabled VLANs is not supported in Cisco Nexus 3232C and 3264Q switches. VXLAN with flood and learn and Layer 2 EVPN is supported in Cisco Nexus 3232C and 3264Q switches.

- 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.

• 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` CLI command. This is not applicable to Cisco 9200 and 9300-EX Series switches and Cisco 9500 Series switches with 9700-EX line cards.

• `show` commands with the `internal` keyword are not supported.

• FEX ports do not support IGMP snooping on VXLAN VLANs.

• Beginning with Cisco NX-OS Release 7.0(3)I4(2), 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.

• RA CLs are not supported on Layer 3 uplinks for VXLAN traffic. Egress VACLs support is not available for de-capsulated packets in the network to access direction on the inner payload.

As a best practice, use PACLs/VACLs for the access to the network direction.

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

• The QoS buffer-boost feature is not applicable for VXLAN traffic.

• For 7.0(3)I1(2), Cisco Nexus 9500 platform switches do not support VXLAN tunnel endpoint functionality, however they can be used as spines.

• SVI and subinterfaces as uplinks are not supported.

• VTEPs do not support VXLAN encapsulated traffic over Parent-Interfaces if subinterfaces are configured. This is regardless of VRF participation.

• VTEPs do not support VXLAN encapsulated traffic over subinterfaces. This is regardless of VRF participation or IEEE 802.1q encapsulation.

• Mixing Sub-Interfaces for VXLAN and non-VXLAN enabled VLANs is not supported.

• Point to multipoint Layer 3 and SVI uplinks are not supported.

• For 7.0(3)I2(1) and later, 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. This aids the traffic when the Layer 3 uplink (underlay) connectivity is lost for one of the vPC peers.

• 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.

• For 7.0(3)I2(1) and later, VXLAN is supported on Cisco Nexus 9500 Series switches with the following line cards:
  • 9500-R
  • 9564PX
- 9564TX
- 9536PQ
- 9700-EX
- 9700-FX

Cisco Nexus 9300 Series switches with 100G uplinks only support VXLAN switching/bridging. (7.0(3)I2(1) and later)

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

---

**Note**

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

---

- For 7.0(3)I2(1) and later, MDP is not supported for VXLAN configurations.
- For 7.0(3)I2(1) and later, bidirectional PIM is not supported for underlay multicast.
- Consistency checkers are not supported for VXLAN tables.
- 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.
- 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. (7.0(3)I3(1) and later).
- For the NX-OS 7.0(3)I2(3) release, the VXLAN network identifier (VNID) 16777215 is reserved and should not be configured explicitly.
- For 7.0(3)I4(1) and later, VXLAN supports In Service Software Upgrade (ISSU).
- VXLAN does not support co-existence with the GRE tunnel feature or the MPLS (static or segment-routing) feature on Cisco Nexus 9000 Series switches with a Network Forwarding Engine (NFE).
- VTEP connected to FEX host interface ports is not supported (7.0(3)I2(1) and later).
- In Cisco NX-OS Release 7.0(3)I4(1), 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.

---

- The following features are not supported:
  - Consistency checkers are not supported for VXLAN tables.
  - DHCP snooping and DAI features are not supported on VXLAN VLANs.
• IPv6 for VXLAN EVPN ESI MH is not supported.

• Native VLANs for VXLAN are not supported. All traffic on VXLAN Layer 2 trunks needs to be tagged. This limitation is applicable to Cisco Nexus 9300 and 9500 switches with 95xx line cards. This is not applicable to Cisco Nexus 9200, 9300-EX, 9300-FX, and 9500 platform switches with -EX or -FX line cards.

• QoS buffer-boost is not applicable for VXLAN traffic.

• QoS classification is not supported for VXLAN traffic in the network-to-host direction as ingress policy on uplink interface.

• Static MAC pointing to remote VTEP (VXLAN Tunnel End Point) is not supported with BGP EVPN (Ethernet VPN).

• TX SPAN (Switched Port Analyzer) for VXLAN traffic is not supported for the access-to-network direction.

• VXLAN routing and VXLAN Bud Nodes features on the 3164Q platform are not supported.

• The following ACL related features are not supported:

  • Egress RACL that is applied on an uplink Layer 3 interface that matches on the inner or outer payload in the access-to-network direction (encapsulated path).

  • Ingress RACL that is applied on an uplink Layer 3 interface that matches on the inner or outer payload in the network-to-access direction (decapsulated path).

Considerations for VXLAN Deployment

• When configuring VXLAN BGP EVPN, only the "System Routing Mode: Default" is applicable for the following hardware platforms:

  • Cisco Nexus 9200/9300-EX/FX/FX2

  • Cisco Nexus 9300 platform switches

  • Cisco Nexus 9500 platform switches with X9500 line cards

  • Cisco Nexus 9500 platform switches with X9700-EX/FX/FX2 line cards

• The “System Routing Mode: template-vxlan-scale” is not applicable to Cisco NX-OS Release 7.0(3)I5(2) and later.

• 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 (7.0(3)I2(2) and later), you can use the `source-interface hold-down-time` command to suppress advertisement of the NVE loopback address until the overlay has converged. The range for the `hold-down-time` is 0 - 2147483647 seconds. The default is 300 seconds.

• To establish IP multicast routing in the core, IP multicast configuration, PIM configuration, and RP configuration is required.

• VTEP to VTEP unicast reachability can be configured through any IGP protocol.

• In VXLAN flood and learn mode (7.0(3)I1(2) and earlier), the default gateway for VXLAN VLANs should be provisioned on external routing devices.

  In VXLAN flood and learn mode (7.0(3)I2(1) and later), 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 (7.0(3)I2(1) and later), 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, 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:

```plaintext
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 added or removed from a VTEP, the NVE interfaces on both the vPC primary and the vPC secondary should be shut before the change is made.

• 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, it is recommended 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, encaps routes are installed for multicast.

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

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

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

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

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

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

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

• When peer-link is no-shut, the NVE loopback address is brought up again and the route is advertised upstream, attracting traffic.

• For vPC, the loopback interface has 2 IP addresses: the primary IP address and the secondary IP address. The primary IP address is unique and is used by Layer 3 protocols.

The secondary IP address on loopback is necessary because the interface NVE uses it for the VTEP IP address. The secondary IP address must be same on both vPC peers.

• The vPC peer-gateway feature must be enabled on both peers.

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

In addition, increase the STP hello timer to 4 seconds to avoid unnecessary TCN generations when vPC role changes occur.
The following is an example (best practice) of a vPC configuration:

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

- **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.
- Enabling vpc peer-gateway configuration is mandatory. For peer-gateway functionality, at least one backup routing SVI is required to be enabled across peer-link and also configured with PIM. This provides a backup routing path in the case when VTEP loses complete connectivity to the spine. Remote peer reachability is re-routed over 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.

The following is an example of backup SVI with PIM enabled:

```
switch# sh ru int vlan 2
interface Vlan2
  description backup_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.
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 must be increased by 50 bytes. If the overlays use a 1500-byte MTU, the transport network must 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 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 number 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 multitenant 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.
Considerations for the Transport Network

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

- On the VTEP device:
  - Enable and configure IP multicast.*
  - Create and configure a loopback interface with a /32 IP address.
    (For vPC VTEPs, you must configure primary and secondary /32 IP addresses.)
  - Enable UP multicast on the loopback interface. *
  - Advertise the loopback interface /32 addresses through the routing protocol (static route) that runs in the transport network.
  - Enable IP multicast on the uplink outgoing physical interface. *

- Throughout the transport network:
  - Enable and configure IP multicast.*

With the Cisco Nexus 9200, 9300-EX, 9300-FX, and 9300-FX2, the use of the system nve infra-vlans command is required, as 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 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).
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.

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

### Configuring VXLAN

#### Enabling VXLANs

**Procedure**

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

#### Mapping VLAN to VXLAN VNI

**Procedure**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Step 2 vlan vlan-id</td>
<td>Specifies VLAN.</td>
</tr>
<tr>
<td>Step 3 vn-segment vnid</td>
<td>Specifies VXLAN VNID (Virtual Network Identifier)</td>
</tr>
<tr>
<td>Step 4 exit</td>
<td>Exit configuration mode.</td>
</tr>
</tbody>
</table>
Guidelines and Limitations for Port VLAN Mapping

Port VLAN mapping has the following guidelines and limitations:

- Before removing a port-channel which has VLAN mapping configured, VLAN mappings on the interface must be removed.
- CoS (QoS) marking is not applicable for the VLANs which are translated on a port.
- Do not configure translation on the native VLAN.
- When SPAN / Ethalyzer is used to capture the traffic on PV enabled ports, only the incoming 802.1q tag is seen in the captured traffic.
- On a port VLAN translation enabled port, traffic should not be received in translated VLAN. If traffic is received on a translated VLAN on a port VLAN translation-enabled port, traffic will fail.
- Overlapping VLAN mapping is supported. For example, `switchport vlan mapping 10 20`, `switchport vlan mapping 20 30`. Traffic can hit the port with VLAN 10 and VLAN 20, but not with VLAN 30 as it is a translated VLAN.
- Port VLAN mapping is not supported on FEX ports.
- Control packets support for translation are ARP, IPv6 neighbor discovery, IPv6 neighbor solicitation.

Configuring Port VLAN Mapping on a Trunk Port

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 egress 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 BGP EVPN mode for VXLAN.

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 an L2 VNI. There is an operational benefit of terminating different customer VLANs and mapping them to the fabric-managed-VLANs, L2 VNIs.

Notes for Port VLAN Mapping:

- Beginning with Cisco NX-OS Release 7.0(3)I7(5), routing is supported on translated VLANs with port VLAN mapping configured on trunk ports. This is supported on Cisco Nexus 9300-EX, 9300-EX, and 9300-FX2 platform switches.
- Port VLAN mapping is supported on Cisco Nexus 9300 platform switches. Beginning with Cisco NX-OS Release 7.0(3)I6(1), port VLAN mapping is supported on Cisco Nexus 9300-EX and 9500 platform switches with 9700-EX line cards with the following exceptions:
Only Layer 2 (no routing) is supported with port VLAN on these switches.

No inner VLAN mapping is supported.

Beginning with Release 7.0(3)I7(4), Cisco Nexus 9300, and 9500 switches support switching on overlapped VLAN interfaces; only VLAN-mapping switching is applicable for Cisco Nexus 9500 with EX/FX line cards.

Beginning with Cisco NX-OS 7.0(3)I7(3), port VLAN switching is supported on 9300-FX2 platform switches.

Beginning with Cisco NX-OS 7.0(3)I7(1), port VLAN switching is supported on 9300-FX platform switches.

Beginning with Cisco NX-OS Release 7.0(3)I2(1), Cisco Nexus 9300 platform switches with NFE ASIC Port VLAN switching is supported.

Beginning with Cisco NX-OS Release 7.0(3)I1(2), Cisco Nexus 9300 platform switches with NFRE ASIC Port VLAN routing is supported.

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.

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 mapping configuration with that VLAN as the translated VLAN, should be flapped to ensure correct behavior.

For example:

```plaintext
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 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.

```plaintext
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*/
```

Now `int po 101` has the "switchport vlan mapping enable" configuration, while eth 1/8 does not. If you want to add eth 1/8 to port channel 101, you first need to apply the "switchport vlan mapping enable" configuration on eth 1/8, and then use the `force` command.

```
int eth 1/8
switchport vlan mapping enable
channel-group 101 force
```

- Port VLAN mapping is not supported on Cisco Nexus 9200 Series switches.

**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.

**Procedure**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong> interface type port</td>
<td>Enters interface configuration mode.</td>
</tr>
<tr>
<td><strong>Step 3</strong> [no] switchport vlan mapping enable</td>
<td>Enables VLAN translation on the switch port. VLAN translation is disabled by default.</td>
</tr>
<tr>
<td><strong>Step 4</strong> [no] switchport vlan mapping <code>vlan-id translated-vlan-id</code></td>
<td>Translates a VLAN to another VLAN.</td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Command or Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>• The range for both the <code>vlan-id</code> and <code>translated-vlan-id</code> arguments is from 1 to 4094.</td>
<td></td>
</tr>
<tr>
<td>• 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.</td>
<td></td>
</tr>
<tr>
<td>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 egress out.</td>
<td></td>
</tr>
<tr>
<td>Note</td>
<td>Use the <code>no</code> form of this command to clear the mappings between a pair of VLANs.</td>
</tr>
</tbody>
</table>

### Step 5

**[no]** switchport vlan mapping all

Removes all VLAN mappings configured on the interface.

### 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 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# config t
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

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.
## Procedure

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Step 2</td>
<td>interface type port</td>
<td>Enters interface configuration mode.</td>
</tr>
<tr>
<td>Step 3</td>
<td>[no] switchport mode trunk</td>
<td>Enters trunk configuration mode.</td>
</tr>
<tr>
<td>Step 4</td>
<td>switchport vlan mapping enable</td>
<td>Enables VLAN translation on the switch port. VLAN translation is disabled by default.</td>
</tr>
<tr>
<td>Note</td>
<td>Use the no form of this command to disable VLAN translation.</td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td>switchport vlan mapping outer-vlan-id inner-vlan-id translated-vlan-id</td>
<td>Translates inner VLAN and outer VLAN to another VLAN.</td>
</tr>
<tr>
<td>Step 6</td>
<td>(Optional) copy running-config startup-config</td>
<td>Copies the running configuration to the startup configuration.</td>
</tr>
<tr>
<td>Note</td>
<td>The VLAN translation configuration does not become effective until the switch port becomes an operational trunk port.</td>
<td></td>
</tr>
<tr>
<td>Step 7</td>
<td>(Optional) show interface [if-identifier] vlan mapping</td>
<td>Displays VLAN mapping information for a range of interfaces or for a specific interface.</td>
</tr>
</tbody>
</table>

## Example

This example shows how to configure translation of double tag VLAN traffic (inner VLAN 12; outer VLAN 11) to VLAN 111.

```
switch# config t
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

<table>
<thead>
<tr>
<th>VLAN</th>
<th>MAC Address</th>
<th>Type</th>
<th>age</th>
<th>Secure NTFY Ports</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>* 111 0000.0092.0001</td>
<td>dynamic</td>
<td>0</td>
<td>F F nve1(100.100.100.254)</td>
</tr>
<tr>
<td></td>
<td>* 111 0000.0940.0001</td>
<td>dynamic</td>
<td>0</td>
<td>F F Eth1/1</td>
</tr>
</tbody>
</table>
```
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:

<table>
<thead>
<tr>
<th>Procedure</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Step 2</td>
<td>interface nve x</td>
<td>Creates a VXLAN overlay interface that terminates VXLAN tunnels.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note Only 1 NVE interface is allowed on the switch.</td>
</tr>
<tr>
<td>Step 3</td>
<td>source-interface src-if</td>
<td>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.</td>
</tr>
<tr>
<td>Step 4</td>
<td>member vni vni</td>
<td>Associate VXLAN VNIs (Virtual Network Identifiers) with the NVE interface.</td>
</tr>
<tr>
<td>Step 5</td>
<td>mcast-group start-address [end-address]</td>
<td>Assign a multicast group to the VNIs.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note used only for BUM traffic</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Procedure</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Step 2</td>
<td>mac address-table static mac-address vni vni-id interface nve x peer-ip ip-address</td>
<td>Specifies the MAC address pointing to the remote VTEP.</td>
</tr>
</tbody>
</table>

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

**Procedure**

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

**Configuring BGP EVPN Ingress Replication**

The following enables BGP EVPN with ingress replication for peers.

**Procedure**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
</tbody>
</table>
### Configuring VXLAN

**Purpose**

- Creates a VXLAN overlay interface that terminates VXLAN tunnels.

**Command or Action**

- `interface nve x`

**Step 2**

- Only 1 NVE interface is allowed on the switch.

**Note**

**Command or Action**

- `source-interface src-if`

**Step 3**

- 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.

**Command or Action**

- `member vni vni`

**Step 4**

- Associate VXLAN VNIs (Virtual Network Identifiers) with the NVE interface.

**Command or Action**

- `ingress-replication protocol bgp`

**Step 5**

- Enables BGP EVPN with ingress replication for the VNI.

---

### Configuring Static Ingress Replication

The following enables static ingress replication for peers.

**Procedure**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>configuration terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><code>interface nve x</code></td>
<td>Creates a VXLAN overlay interface that terminates VXLAN tunnels. Only 1 NVE interface is allowed on the switch.</td>
</tr>
<tr>
<td>`member vni [vni-id</td>
<td>vni-range]`</td>
</tr>
<tr>
<td><code>ingress-replication protocol static</code></td>
<td>Enables static ingress replication for the VNI.</td>
</tr>
<tr>
<td><code>peer-ip n.n.n.n</code></td>
<td>Enables peer IP.</td>
</tr>
</tbody>
</table>

---

### Guidelines and Limitations for Q-in-VNI

Q-in-VNI has the following limitations:

- Q-in-VNI and Selective Q-in-VNI are supported only with VXLAN Flood and Learn.
• Q-in-VNI, Selective Q-in-VNI, and QinQ-QinVNI features are not supported with Multicast underlay on Nexus 9000 EX platforms.

• It is recommended that you enter the system dot1q tunnel transit when running these features on vPC VTEPs.

• For proper operation during L3 uplink failure scenarios on vPC VTEPs configure 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.

• Single tag is supported on Cisco Nexus 9300 platform switches. It can be enabled by unconfiguring the overlay-encapsulation vxlan-with-tag command from an interface NVE:

```
switch(config)# int nve 1
switch (config-if-nve)# no overlay-encapsulation vxlan-with-tag
```

```
switch # sh run int nve 1
!Command: show running-config interface nve1

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
  suppress-arp
  mcast-group 225.4.0.1
```

• Single tag is not supported on Cisco Nexus 9500 platform switches; only double tag is supported.

• Double tag is not supported on Cisco Nexus 9300-EX platform switches, only single tag is supported.

• When upgrading from Cisco NX-OS Release 7.0(3)I3(1) or 7.0(3)I4(1) to Cisco NX-OS Release 7.0(3)I7(5) with Cisco Nexus 9300 platform switches without the overlay-encapsulation vxlan-with-tag command under interface NVE, you should add overlay-encapsulation vxlan-with-tag under the NVE interface in the older release before starting the ISSU upgrade. We were only supporting double tag in Cisco NX-OS Release 7.0(3)I3(1) and 7.0(3)I4(1). We now support single tag also in Cisco NX-OS Release 7.0(3)I7(5).

• We do not support traffic between ports that are configured for Q-in-VNI and ports that are configured for trunk on Cisco Nexus 9300-EX platform switches.

• Q-in-VNI is supported only with both flood and learn.

• The Q-in-VNI feature cannot coexist with a VTEP which has Layer 3 subinterfaces configured.

• The Q-in-VNI or selective Q-in-VNI feature is not supported with VXLAN or VXLAN EVPN on Cisco Nexus 9000-EX platform switches when Multicast is used for BUM replication (L2VNI).

## Configuring Q-in-VNI

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

Notes about configuring a Q-in-VNI:
• Q-in-VNI only supports VXLAN bridging. It does not support VXLAN routing.

• Q-in-VNI does not support FEX.

• When configuring access ports and trunk ports:
  • For NX-OS 7.0(3)I2(2) and earlier releases, when a switch is in dot1q mode, you cannot have access ports or trunk ports configured on any other interface on the switch.
  • For NX-OS 7.0(3)I3(1) and later releases running on a Network Forwarding Engine (NFE), you can have access ports, trunk ports and dot1q ports on different interfaces on the same switch.
  • For NX-OS 7.0(3)I5(1) and later releases running on a Leaf Spine Engine (LSE), you can have access ports, trunk ports and dot1q ports on different interfaces on the same switch.

• For NX-OS 7.0(3)I3(1) and later releases, you cannot have the same VLAN configured for both dot1q and trunk ports/access ports.

Before you begin

Configuring the Q-in-VNI feature requires:

• The base port mode must be a dot1q tunnel port with an access VLAN configured.

• VNI mapping is required for the access VLAN on the port.

• 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).

Procedure

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Step 2</td>
<td>interface type port</td>
<td>Enters interface configuration mode.</td>
</tr>
<tr>
<td>Step 3</td>
<td>switchport mode dot1q-tunnel</td>
<td>Creates a 802.1Q tunnel on the port.</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
<td></td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> switchport access vlan vlan-id</td>
<td>Specifies the port assigned to a VLAN.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> spanning-tree bpdufilter enable</td>
<td>Enables BPDU Filtering for the specified spanning tree edge interface. By default, BPDU Filtering is disabled.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> interface nve x</td>
<td>Creates a VXLAN overlay interface that terminates VXLAN tunnels.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Note: This step is required for NX-OS 7.0(3)I2(2) and earlier releases.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Note: This step is not required for NX-OS 7.0(3)I3(1) and later releases.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> overlay-encapsulation vxlan-with-tag</td>
<td>Enables Q-in-VNI.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Note: This step is required for NX-OS 7.0(3)I2(2) and earlier releases:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Note: This step is not required for NX-OS 7.0(3)I3(1) and later releases.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Note: Starting with Release 7.0(3)I5(1), this step is not required for Cisco Nexus 9000 Series switches with Application Spine Engine (ASE). Also, provider tagging (double tagging) is not applicable for Cisco Nexus 9000 Series switches with Application Spine Engine (ASE).</td>
<td></td>
</tr>
</tbody>
</table>

**Example**

- The following is an example of configuring a Q-in-VNI (NX-OS 7.0(3)I2(2) and earlier releases):

  ```
  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)# interface nve1
  switch(config-if)# overlay-encapsulation vxlan-with-tag
  ```

- The following is an example of configuring a Q-in-VNI (NX-OS 7.0(3)I3(1) and later releases):

  ```
  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
  ```
**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).

Beginning with Cisco NX-OS Release 7.0(3)I5(2), selective Q-in-VNI is supported on both vPC and non-vPC ports on Cisco Nexus 9300-EX Series switches. This feature is not supported on Cisco Nexus 9300 Series and 9200 Series switches.

This feature is also supported with flood and learn in IR mode.

See the following guidelines for selective Q-in-VNI:

- **Beginning with Cisco NX-OS Release 7.0(3)I5(2), configuring selective Q-in-VNI on one VXLAN 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.

- **Configure the `system dot1q-tunnel transit` CLI on the vPC switches with selective Q-in-VNI configurations. This CLI configuration 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.**

- **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>` CLI 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>` CLI command.

```plaintext
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
  no shutdown

• Only the native VLANs and the service provider VLANs are allowed on the selective Q-in-VNI port. No other VLANs are allowed on the selective Q-in-VNI port and even if they are allowed, the packets for those VLANs are not forwarded.

See the following configuration examples.

• See the following example for the provider VLAN configuration:

```plaintext
vlan 50
  vn-segment 10050

• See the following example for configuring VXLAN Flood and Learn with Ingress Replication:

```plaintext
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:

```plaintext
interface nve1
  no shutdown
  source-interface loopback0 member vni 10050
  mcast-group 230.1.1.1

• See the following example for the native VLAN configuration:

```plaintext
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 trunk vlan mapping 200-700 dot1q-tunnel 50
switch(config-if)# switchport trunk allowed vlan 50,150
switch(config-if)# no shutdown

## Configuring Q-in-VNI with LACP Tunneling

Q-in-VNI can be configured to tunnel LACP packets.

### Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure terminal</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>interface type port</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>switchport mode dot1q-tunnel</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>switchport access vlan vlan-id</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>interface nve x</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>overlay-encapsulation vxlan-with-tag tunnel-control-frames lacp</td>
</tr>
</tbody>
</table>

### Example

- The following is an example of configuring a Q-in-VNI for LACP tunneling (NX-OS 7.0(3)I2(2) and earlier releases):
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)# interface nve1
switch(config-if)# overlay-encapsulation vxlan-with-tag tunnel-control-frames

---

Note

- STP is disabled on VNI mapped VLANs.
- No spanning-tree VLAN <> on the VTEP.
- No MAC address-table notification for mac-move.
- As a best practice, configure a fast LACP rate on the interface where the LACP port is configured. Otherwise the convergence time is approximately 90 seconds.

---

• The following is an example of configuring a Q-in-VNI for LACP tunneling (NX-OS 7.0(3)I3(1) and later releases):

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)# interface nve1
switch(config-if)# overlay-encapsulation vxlan-with-tag tunnel-control-frames lacp

---

Note

- STP is disabled on VNI mapped VLANs.
- No spanning-tree VLAN <> on the VTEP.
- No MAC address-table notification for mac-move.
- As a best practice, configure a fast LACP rate on the interface where the LACP port is configured. Otherwise the convergence time is approximately 90 seconds.

---

• 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 2: LACP Tunneling Over VXLAN P2P Tunnels

- 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.

---

### 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 EVVPN.

• This feature is supported on the Cisco Nexus 9300-FX and Cisco Nexus 9300-FX2 switches.

**Guidelines and Limitations for QinQ-QinVNI**

QinQ-QinVNI has the following guidelines and limitations:

• 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 is supported with VXLAN bridging. It does not support VXLAN routing.

• Multicast data traffic with more than two Q-Tags in 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 the vPC peers. Otherwise, traffic traversing via the peer-link for these provider VLANs will not carry all inner C-Tags.

**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.

### Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
</tbody>
</table>

Example:

```
switch# configure terminal
```
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 2</strong> interface ethernet slot/port</td>
<td>Specifies the interface that you are configuring.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch(config)# interface ethernet1/7</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> switchport</td>
<td>Configures it as a Layer 2 port.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch(config-inf)# switchport</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> switchport mode trunk</td>
<td>Sets the interface as a Layer 2 trunk port.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>switch(config-inf)# switchport mode trunk</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> switchport trunk native vlan vlan-id</td>
<td>Sets the native VLAN for the 802.1Q trunk. Valid values are</td>
</tr>
<tr>
<td>Example:</td>
<td>from 1 to 4094. The default value is VLAN1.</td>
</tr>
<tr>
<td>switch(config-inf)# switchport trunk native vlan 30</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> switchport trunk allowed vlan vlan-list</td>
<td>Sets the allowed VLANs for the trunk interface. The default</td>
</tr>
<tr>
<td>Example:</td>
<td>to allow all VLANs on the trunk interface: 1 to 3967 and</td>
</tr>
<tr>
<td>switch(config-inf)# switchport trunk allowed vlan 10,20,30</td>
<td>4048 to 4094. VLANs 3968 to 4047 are the default VLANs</td>
</tr>
<tr>
<td></td>
<td>reserved for internal use by default.</td>
</tr>
<tr>
<td><strong>Step 7</strong> switchport trunk allow-multi-tag</td>
<td>Sets the allowed VLANs as the provider VLANs excluding the</td>
</tr>
<tr>
<td>Example:</td>
<td>native VLAN. In the following example, VLANs 10 and 20 are</td>
</tr>
<tr>
<td>switch(config-inf)# switchport trunk allow-multi-tag</td>
<td>provider VLANs and can carry multiple Inner Q-tags. Native</td>
</tr>
<tr>
<td></td>
<td>VLAN 30 will not carry inner Q-tags.</td>
</tr>
</tbody>
</table>

Example

```
interface Ethernet1/7
switchport
switchport mode trunk
switchport trunk native vlan 30
switchport trunk allowed vlan 10,20,30
no shutdown
```

**Removing a VNI**

Use this procedure to remove a VNI.
Procedure

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>Remove the VNI under NVE.</td>
</tr>
<tr>
<td>Step 2</td>
<td>Remove the VRF from BGP (applicable when decommissioning for Layer 3 VNI).</td>
</tr>
<tr>
<td>Step 3</td>
<td>Delete the SVI.</td>
</tr>
<tr>
<td>Step 4</td>
<td>Delete the VLAN and VNI.</td>
</tr>
</tbody>
</table>

Configuring FHRP Over VXLAN

Overview for FHRP Over VXLAN

Overview of FHRP

Starting with Release 7.0(3)I5(1), you can configure First Hop Redundancy Protocol (FHRP) over VXLAN on Cisco Nexus 9000 Series switches. The FHRP provides a redundant Layer 3 traffic path. It provides fast failure detection and transparent switching of the traffic flow. The FHRP avoids the use of the routing protocols on all the devices. It also avoids the traffic loss that is associated with the routing or the discovery protocol convergence. It provides an election mechanism to determine the next best gateway. Current FHRP supports HSRPv1, HSRPv2, VRRPv2, and VRRPv3.

FHRP over VXLAN

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

Guidelines and Limitations for FHRP Over VXLAN

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

- Configuring FHRP over VXLAN allows the FHRP protocols to peer using the hello packets that are flooded on the VXLAN overlay. The ACLs have been programmed into the Cisco Nexus 9500 Series switches that allow the HSRP packets that are flooded on the overlay to be punted to the supervisor module.

- When using FHRP with VXLAN, ARP-ETHER TCAM must be carved using the `hardware access-list tcam region arp-ether 256` CLI command.

- Configuring FHRP over VXLAN is supported for both IR and multicast flooding of the FHRP packets. The FHRP protocol working does not change for configuring FHRP over VXLAN.

- The FHRP over VXLAN feature is supported for flood and learn only.

- For Layer 3 VTEPs in BGP EVPN, only anycast GW is supported.

- Beginning with Cisco NX-OS Release 7.0(3)I5(2), configuring FHRP over VXLAN is supported on the Cisco Nexus 9200, 9300, and 9300-EX Series switches.
Only Supported Deployments for FHRP Over VXLAN

See the following illustrations for only supported deployments for FHRP over VXLAN protocols.

**Figure 3: FHRP over VXLAN Leafs as Layer 3 Gateway**

**Figure 4: FHRP over VXLAN Spine as Layer 3 Gateway**
See the following configuration example for FHRP over VXLAN Leafs as Layer 3 Gateway (Figure 2) and FHRP over VXLAN Spine as Layer 3 Gateway (Figure 3):

```
BL1 / S1 FHRP configuration with HSRP
# VLAN with VNI
vlan 10
  vn-segment 10000

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

BL2 / S2 FHRP configuration with HSRP
# VLAN with VNI
vlan 10
  vn-segment 10000

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

The FHRP configuration can leverage HSRP or VRRP. The VLAN for FHRP has to be allowed on the vPC peer-link and as vPC is used, FHRP operates in active/active. The VNI mapped to the VLAN must be configured on the NVE interface and it is associated with the used BUM replication mode (Multicast or Ingress Replication).

**New Supported Topology for Configuring FHRP Over VXLAN**

Configuring FHRP over VXLAN is supported on the following Cisco Nexus 9000 Series switches and line cards:

- Cisco Nexus 9300 Series switches
- N9K-X9536PQ line cards
- N9k-X9564TX line cards
- N9K-X9564PX line cards

In the new topology for configuring FHRP over VXLAN, Bi-Directional Forwarding (BFD) is not supported with HSRP.

See the following new supported topology for configuring FHRP over VXLAN:
In the above topology, FHRP can be configured on the Spine Layer. The FHRP protocols synchronize its state with the hellos that get flooded on the Overlay without having a dedicated Layer 2 link in between the peers. The FHRP is operated in active (master)/standby (backup) as no vPC is being deployed.

See the following configuration example for the topology:

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

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

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

# Layer-3 Interface with FHRP (HSRP)
interface vlan 10
   ip address 192.168.1.3
   hsrp 10
      ip 192.168.1.1
The FHRP configuration can leverage HSRP or VRRP. No vPC peer-link is necessary and therefore no VLAN is allowed on the vPC peer-link. The VNI mapped to the VLAN must be configured on the NVE interface and it is associated with the used BUM replication mode (Multicast or Ingress Replication).

### Configuring IGMP Snooping Over VXLAN

#### Overview of IGMP Snooping Over VXLAN

Starting with Cisco NX-OS Release 7.0(3)f3(4), you can configure IGMP snooping over VXLAN. This feature is available on the Cisco Nexus 9508 switch with 9636-RX line cards.

Starting with Cisco NX-OS Release 7.0(3)i5(1), you can configure IGMP snooping over VXLAN. The configuration of IGMP snooping is same in VXLAN as in configuration of IGMP snooping in regular VLAN domain. For more information on IGMP snooping, see the Configuring IGMP Snooping section in 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:

- For IGMP snooping over VXLAN, all the guidelines and limitations of VXLAN apply.
- Beginning with Cisco NX-OS Release 7.0(3)i6(1), IGMP snooping on VXLAN VLANs is supported for Cisco Nexus 9300 and 9300-EX platform switches with multicast overlay networks and ingress replication underlay networks.
- Beginning with Cisco NX-OS Release 7.0(3)i5(1), IGMP snooping on VXLAN VLANs is supported for Cisco Nexus 9300 and 9300-EX platform switches and only with multicast underlay networks (not with ingress replication underlay networks).
- Beginning with Cisco NX-OS Release 7.0(3)i5(2), VXLAN IGMP snooping is supported on Cisco Nexus 9300 platform switches and Cisco Nexus 9500 platform switches with N9K-X9732C-EX line cards.
- By default, unknown multicast traffic gets flooded to the VLAN domains on Cisco Nexus 9300 platform switches.
- IGMP snooping over VXLAN is not supported on any FEX enabled platforms and FEX ports.

### Configuring IGMP Snooping Over VXLAN

#### Before you begin

For VXLAN IGMP snooping functionality, the ARP-ETHER TCAM must be configured in the double-wide mode using the `hardware access-list tcam region arp-ether 256 double wide` command for Cisco Nexus 9300 switches. This command is not required for Cisco Nexus 9300-EX switches.
### Procedure

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>switch(config)#ip igmp snooping vxlan</code></td>
<td>Enables IGMP snooping for VXLAN VLANs. You have to explicitly configure this command to enable snooping for VXLAN VLANs.</td>
</tr>
<tr>
<td>Step 2</td>
<td><code>switch(config)#ip igmp snooping disable-nve-static-router-port</code></td>
<td>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.</td>
</tr>
</tbody>
</table>
| Step 3 | `switch(config)#system nve ipmc global index-size ?`  
**Example:**  
`switch(config)# system nve ipmc global index-size ?`  
`<1000-7000> Ipmc allowed size` | Configures the VXLAN global IPMC index size. IGMP snooping over VXLAN uses the IPMC indexes from the NVE global range on the Cisco Nexus 9000 Series switches with Network Forwarding Engine (NFE). You need to reconfigure the VXLAN global IPMC index size according to the scale using this command. Cisco recommends to reserve 6000 IPMC indexes using this CLI command. The default IPMC index size is 3000.  
**Note** This command is not available on the Cisco Nexus 9508 platform switch. |
| Step 4 | `switch(config)# ip igmp snooping vxlan-umc drop vlan ?`  
**Example:**  
`switch(config)# ip igmp snooping vxlan-umc drop vlan ?`  
`<1-3863> VLAN IDs for which unknown multicast traffic is dropped` | Configures IGMP snooping over VXLAN to drop all the unknown multicast traffic on per VLAN basis using this global CLI command. On Cisco Nexus 9000 Series switches with Network Forwarding Engine (NFE), the default behavior of all unknown multicast traffic is to flood to the bridge domain.  
**Note** This command is not available on the Cisco Nexus 9508 platform switch. |

### Configuring Line Cards for VXLAN

This procedure applies only to the Cisco Nexus 9508 switch.

This procedure configures line cards for either VXLAN or MPLS. All line cards in the chassis must be either VXLAN or MPLS. They cannot be mixed.
Centralized VRF Route Leaking using Default-Routes and Aggregates

Overview

Centralizing VRF route leaks using default-routes facilitates installation and configuration of new hardware or software that must coexist with legacy systems, without any additional configuration overheads on the legacy nodes. However, enabling shared services and default-VRF access scenarios may require one additional configuration on a per-VRF-AF level in the Border Leaf (BL). Though the leaf nodes may not require configuration changes, the BLs must have the knowledge of all VRFs, as well as the fabric entry and exit points. EVPN enables multi-tenancy support by segregating traffic among the tenants. While segregation among different tenants is maintained in most cases, supporting the capability of cross-tenant traffic is also equally important for tenants to access common services. In order to achieve traffic segregation, the tenant’s routes are typically placed in different VRFs in an EVPN deployment case.

Deploying EVPN

When an EVPN solution is deployed in an existing datacenter, the legacy switches, that do not have EVPN support, co-exists with EVPN-capable VTEPs. The VTEPs supports tenant traffic segregation. Tenant routes are placed in the VRF while the legacy switches are typically placed in the global VRF. Existing servers remains connected to legacy switches. The hosts in the tenant’s VRF must have access to servers placed under the legacy switches in the global VRF. Access to the default-VRF is enabled by allowing routes, that are imported already, in a non-default-VRF, to be re-imported into the default-VRF. That in turn advertises the VPN learnt prefixes outside of the fabric. Because there is no support in EVPN similar to VPNv4 for advertising the default-routes directly via the VPN session, the default-route must be originated from the VRF AF. You must preferably use route-maps to control prefix leaking from the VRFs into the default-VRF.
Reachability between Leaves

EVPN Cross-VRF Connectivity between leaves is achieved by packet re-encapsulation on the BL, which will be the VTEP for all VNIs requiring cross-VRF reachability. Default routes provides cross-VRF reachability to the legacy nodes.

VPN to Default-VRF Reachability

Routes are not imported directly from VPN into the default-VRF. You must configure a VRF to import and hold those routes, which will then be evaluated for importing into the default-VRF after configuring the knob. Because all VRFs may be importing the other VRFs’ routes, only one VRF may be needed to leak its routes to the default-VRF for providing full VPN to default-VRF Reachability.

Guidelines and Limitations

- Centralized VRF Route Leaking is supported only on Cisco Nexus 9200 and 9300-EX platform switches
- Each prefix needs to be imported into each VRF for full EVPN Cross-VRF Reachability.
- Memory complexity of the deployment can be described by a O(NxM ) formula, where N is the number of prefixes, M is the number of EVPN VRFs.
- You must configure “feature bgp” to have access to “export vrf default” command. In order to achieve the full Centralized Route Leaking on EVPN, you must support downstream VNI assignment.

Note

Downstream VNI is not supported in the Release 7.0(3)I7(1)

- Centralized route leaking applies the longest prefix matching. A leaf with a less specific local route, may not be able to reach a more specific address of that route’s subnet from another VNI, unless you manually configure the border leaf switch to generate those advertisements.
- Hardware support for VXLAN packet re-encapsulation at BL is required for this functionality to work in EVPN.
Configuration Examples for Centralized VRF Route Leak

The following example shows how to leak routes from tenant VRF to default VRF.

```conf
vrf context vrf120
vni 300120
ip route 0.0.0.0/0 Null0 // static default route
ipv6 route ::/0 Null0 // static default route
rd auto
address-family ipv4 unicast
  route-target import 65535:120
  route-target import 65535:120 evpn
  route-target export 65535:120
  route-target export 65535:120 evpn
  import vrf default map permitall // Imports from default VRF to tenant VRF
  export vrf default 100 map block_default allow-vpn
address-family ipv6 unicast
  route-target import 65535:120
  route-target import 65535:120 evpn
  route-target export 65535:120 evpn
  import vrf default map permitall
  export vrf default 100 map block_default_v6 allow-vpn
```

The following example shows how to leak routes from default VRF to tenant VRF.

```conf
router bgp 1001
vrf vrf120
  address-family ipv4 unicast
    network 0.0.0.0/0 // advertises default route to host leaf VTEPs
    advertise 12vpn evpn
    redistribute hmm route-map permitall
    maximum-paths 64
    maximum-paths ibgp 64
  address-family ipv6 unicast
    network ::/0 // advertises default route to host leaf VTEPs
    advertise 12vpn evpn
    redistribute hmm route-map permitall
    maximum-paths 64
    maximum-paths ibgp 64
```

The following is an example configuration on a border-leaf switch to route leaks from one tenant VRF (VRF150) to another tenant VRF (VRF250). In these examples, BL-11 is used as the border-leaf switch. The aggregate-address is used for BL switches to advertise VRF250’s address to leaf switches so that leaf switch can send the routes destined to VRF250 to BL.

```conf
switch# sh run vrf vrf150
!Command: show running-config vrf vrf150
!Time: Thu Aug 3 16:54:57 2017
version 7.0(3)17(1)
interface Vlan150
  vrf member vrf150
vrf context vrf150
vni 300150
rd auto
  address-family ipv4 unicast
```
route-target import 65535:150
route-target import 65535:250 evpn //import VRF250 routes
route-target import 65535:150 evpn //import VRF250 routes
route-target export 65535:150 evpn
address-family ipv4 unicast
route-target import 65535:150
route-target import 65535:150 evpn
route-target import 65535:250 evpn //import VRF250 routes
route-target import 65535:250 evpn //import VRF250 routes
route-target export 65535:150 evpn
route-target export 65535:250 evpn

address-family ipv6 unicast
route-target import 65535:150
route-target import 65535:150 evpn
route-target import 65535:250
route-target import 65535:250 evpn
route-target export 65535:150 evpn
route-target export 65535:250 evpn

router bgp 1001
vrf vrf150
  address-family ipv4 unicast
  advertise 12vpn evpn
  redistribute hmm route-map permitall
  aggregate-address 12.50.0.0/15 //VRF250 has network 12.50.0.0/16
  aggregate-address 22.50.0.0/15 //VRF250 has network 22.50.0.0/16
  maximum-paths 64
  maximum-paths ibgp 64

  address-family ipv6 unicast
  advertise 12vpn evpn
  redistribute hmm route-map permitall
  aggregate-address 2001:0:12:50::/63 //VRF250 has network 2001:0:12:50::/64
  aggregate-address 2001:0:22:50::/63 //VRF250 has network 2001:0:12:50::/64
  maximum-paths 64
  maximum-paths ibgp 64

switch# sh run vrf vrf250
!Command: show running-config vrf vrf250
!Time: Thu Aug 3 17:21:22 2017
version 7.0(3)I7(1)
interface Vlan250
  vrf member vrf250
  vrf context vrf250
  vni 300250
  rd auto
  address-family ipv4 unicast
    route-target import 65535:150
    route-target import 65535:150 evpn
    route-target import 65535:250
    route-target import 65535:250 evpn
    route-target export 65535:150 evpn
    route-target export 65535:250 evpn

  address-family ipv6 unicast
    route-target import 65535:150
    route-target import 65535:150 evpn
    route-target import 65535:250
    route-target import 65535:250 evpn
    route-target export 65535:150 evpn
    route-target export 65535:250 evpn

go back to router 1001
vrf vrf250
  address-family ipv4 unicast
  advertise 12vpn evpn
  redistribute hmm route-map permitall
  aggregate-address 11.50.0.0/15
  aggregate-address 21.50.0.0/15
  maximum-paths 64
  maximum-paths ibgp 64
  address-family ipv6 unicast
VXLAN Tunnel Egress QoS Policy

About VXLAN Tunnel Egress QoS Policy

This feature applies the QoS policy for VXLAN tunnel terminated packets coming to this site. This configuration can be applied to the NVE interface. You can apply all input policies such as policing, scheduling, and marking for decapsulated packets coming from the VXLAN tunnel.

- The QoS policy is applied end-to-end. That is, the ingress QoS policy on access ports, as well as, the ingress NVE interface on the remote side.
- The uniform mode is the default. You have the ability to change the QoS mode by entering the `qos-mode pipe` command.

Guidelines and Limitations for VXLAN Tunnel Egress QoS Policy

VXLAN Tunnel Egress QoS Policy has the following guidelines and limitations:
• Beginning with Cisco NX-OS Release 7.0(3)I7(5), support is added for this feature.
• This feature is supported only on Cisco Nexus 9300-EX, 9300-FX, and 9300-FX2 platform switches.
• This feature is supported only in the EVPN fabric.

## Configuring VXLAN Tunnel Egress QoS Policy

This procedure configures the VXLAN Tunnel Egress QoS Policy.

### Before you begin

VXLAN configuration must be present.

Enter the `show running-config` command to determine the current state.

### Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: <code>switch# configure terminal</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> interface nve1</td>
<td>Creates a VXLAN overlay interface that terminates VXLAN tunnels.</td>
</tr>
<tr>
<td>Example: <code>switch(config)# interface nve1</code></td>
<td>Note: Only 1 NVE interface is allowed on the switch.</td>
</tr>
<tr>
<td><strong>Step 3</strong> service-policy type qos input policy-map-name</td>
<td>Input the service policy. Uniform mode is the default.</td>
</tr>
<tr>
<td>Example: <code>switch(config-if)# service-policy type qos input cos-decap-vlan</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> (Optional) qos-mode pipe</td>
<td>Defines the QoS mode as uniform or pipe. Default mode is uniform.</td>
</tr>
<tr>
<td>Example: <code>switch(config-if)# qos-mode pipe</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> no shutdown</td>
<td>Negate shutdown command.</td>
</tr>
<tr>
<td>Example: <code>switch(config-if)# no shutdown</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> host-reachability protocol bgp</td>
<td>Defines BGP as the mechanism for host reachability advertisement.</td>
</tr>
<tr>
<td>Example: <code>switch(config-if)# host-reachability protocol bgp</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> source-interface loopback1</td>
<td>The source interface must be a loopback interface that is configured on the switch with</td>
</tr>
</tbody>
</table>
### Configuring VXLAN

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>switch(config-if)# source-interface loopback1</code></td>
<td>Associate VXLAN VNIs (Virtual Network Identifiers) with the NVE interface.</td>
</tr>
</tbody>
</table>

**Step 8**

```plaintext
Example:
switch(config-if)# member vni 10101-10102
```

**Step 9**

```plaintext
Example:
switch(config-if)# suppress-arp
```

Verifying the VXLAN Configuration

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

**Table 4: Display VXLAN configuration information (Release 7.0(3)I1(1))**

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>show tech-support vxlan [platform]</code></td>
<td>Displays related VXLAN tech-support information.</td>
</tr>
<tr>
<td><code>show logging level nve</code></td>
<td>Displays logging level.</td>
</tr>
<tr>
<td><code>show tech-support nve</code></td>
<td>Displays related NVE tech-support information.</td>
</tr>
<tr>
<td><code>show run interface nve x</code></td>
<td>Displays NVE overlay interface configuration.</td>
</tr>
<tr>
<td><code>show nve interface</code></td>
<td>Displays NVE overlay interface status.</td>
</tr>
<tr>
<td><code>show nve peers</code></td>
<td>Displays NVE peer status.</td>
</tr>
<tr>
<td><code>show nve peers peer_IP_address interface_ID counters</code></td>
<td>Displays per NVE peer statistics.</td>
</tr>
<tr>
<td><code>clear nve peers peer_IP_address interface_ID counters</code></td>
<td>Clears per NVE peer statistics.</td>
</tr>
<tr>
<td><code>clear nve peer-ip peer-ip-address</code></td>
<td>Clears stale NVE peers. Stale NVE peers are peers that do not have MAC addresses learnt behind them.</td>
</tr>
<tr>
<td><code>show nve vni</code></td>
<td>Displays VXLAN VNI status.</td>
</tr>
</tbody>
</table>
### Configuring VXLAN

#### Verifying the VXLAN Configuration

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>show nve vni ingress-replication</code></td>
<td>Displays the mapping of VNI to ingress-replication peer list and uptime for each peer.</td>
</tr>
<tr>
<td><code>show nve vni vni_number counters</code></td>
<td>Displays per VNI statistics.</td>
</tr>
<tr>
<td><code>clear nve vni vni_number counters</code></td>
<td>Clears per VNI statistics.</td>
</tr>
<tr>
<td><code>show nve vxlan-params</code></td>
<td>Displays VXLAN parameters, such as VXLAN destination or UDP port.</td>
</tr>
</tbody>
</table>

**Table 5: Display VXLAN configuration information (Release 7.0(3)I1(2) and later)**

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>show tech-support vxlan [platform]</code></td>
<td>Displays related VXLAN tech-support information.</td>
</tr>
<tr>
<td>`show interface {ethernet slot/port</td>
<td>port-channel port} vlan mapping`</td>
</tr>
<tr>
<td><code>show logging level nve</code></td>
<td>Displays logging level.</td>
</tr>
<tr>
<td><code>show tech-support nve</code></td>
<td>Displays related NVE tech-support information.</td>
</tr>
<tr>
<td><code>show run interface nve x</code></td>
<td>Displays NVE overlay interface configuration.</td>
</tr>
<tr>
<td><code>show nve interface</code></td>
<td>Displays NVE overlay interface status.</td>
</tr>
<tr>
<td><code>show nve peers</code></td>
<td>Displays NVE peer status.</td>
</tr>
<tr>
<td><code>show nve peers peer_IP_address interface interface_ID counters</code></td>
<td>Displays per NVE peer statistics.</td>
</tr>
<tr>
<td><code>clear nve peers peer_IP_address interface interface_ID counters</code></td>
<td>Clears per NVE peer statistics.</td>
</tr>
<tr>
<td><code>clear nve peer-ip peer-ip-address</code></td>
<td>Clears stale NVE peers. Stale NVE peers are peers that do not have MAC addresses learnt behind them.</td>
</tr>
<tr>
<td><code>show nve vni</code></td>
<td>Displays VXLAN VNI status.</td>
</tr>
<tr>
<td><code>show nve vni ingress-replication</code></td>
<td>Displays the mapping of VNI to ingress-replication peer list and uptime for each peer.</td>
</tr>
<tr>
<td><code>show nve vni vni_number counters</code></td>
<td>Displays per VNI statistics.</td>
</tr>
<tr>
<td><code>clear nve vni vni_number counters</code></td>
<td>Clears per VNI statistics.</td>
</tr>
<tr>
<td><code>show nve vxlan-params</code></td>
<td>Displays VXLAN parameters, such as VXLAN destination or UDP port.</td>
</tr>
<tr>
<td><code>show mac address-table static interface nve 1</code></td>
<td>Displays static MAC information.</td>
</tr>
</tbody>
</table>
### Table 6: Display VXLAN configuration information (Release 7.0(3)I2(2) and later)

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>show vxlan interface</td>
<td>Displays VXLAN interface status for 9200 platform switches.</td>
</tr>
<tr>
<td>show vxlan interface</td>
<td>count</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
</tr>
</tbody>
</table>

### Example of VXLAN Bridging Configuration

- An example of a loopback interface configuration and routing protocol configuration:
• Nexus 9000 VTEP-1 configuration:

```
switch-vtep-1(config)# feature ospf
switch-vtep-1(config)# feature pim
switch-vtep-1(config)# router ospf 1
switch-vtep-1(config-router)# router-id 100.100.100.1
switch-vtep-1(config)# ip pim rp-address 10.1.1.1 group-list 224.0.0.0/4
switch-vtep-1(config)# interface loopback0
switch-vtep-1(config-if)# ip address 100.100.100.1/32
switch-vtep-1(config-if)# ip router ospf 1 area 0.0.0.0
switch-vtep-1(config-if)# ip pim sparse-mode
switch-vtep-1(config)# interface e2/1
switch-vtep-1(config-if)# ip address 20.1.1.1/30
switch-vtep-1(config-if)# ip router ospf 1 area 0.0.0.0
switch-vtep-1(config-if)# ip pim sparse-mode
switch-vtep-1(config)# feature nv overlay
switch-vtep-1(config)# feature vn-segment-vlan-based
switch-vtep-1(config)# interface e1/1
switch-vtep-1(config-if)# switchport
switch-vtep-1(config-if)# switchport access vlan 10
switch-vtep-1(config-if)# no shutdown
switch-vtep-1(config)# interface loopback0
switch-vtep-1(config-if)# no shutdown
switch-vtep-1(config-if)# source-interface loopback0
switch-vtep-1(config-if)# member vni 10000 mcast-group 230.1.1.1
switch-vtep-1(config)# vlan 10
switch-vtep-1(config-vlan)# vn-segment 10000
switch-vtep-1(config-vlan)# exit
```

• Nexus 9000 VTEP-2 configuration:

```
switch-vtep-2(config)# feature ospf
switch-vtep-2(config)# feature pim
switch-vtep-2(config)# router ospf 1
switch-vtep-2(config-router)# router-id 100.100.100.2
switch-vtep-2(config)# ip pim rp-address 10.1.1.1 group-list 224.0.0.0/4
switch-vtep-2(config)# interface loopback0
switch-vtep-2(config-if)# ip address 100.100.100.2/32
```
**An example of an ingress replication topology:**

*Figure 11: Ingress Replication topology*

- Nexus 9000 VTEP-1 configuration:

```bash
switch-vtep-1(config)# feature ospf
switch-vtep-1(config)# router ospf 1
switch-vtep-1(config-router)# router-id 200.200.8.8
switch-vtep-1(config-router)# interface loopback0
switch-vtep-1(config-if)# ip address 200.200.8.8/32
switch-vtep-1(config-if)# ip router ospf 1 area 0.0.0.0
switch-vtep-1(config)# interface e2/1
switch-vtep-1(config-if)# ip address 20.1.1.1/30
switch-vtep-1(config-if)# ip router ospf 1 area 0.0.0.0
switch-vtep-1(config-if)# ip pim sparse-mode
switch-vtep-1(config)# feature nv overlay
switch-vtep-1(config)# feature vn-segment-vlan-based
```
Example of VXLAN Bridging Configuration

```bash
switch-vtep-1(config)# interface e1/1
switch-vtep-1(config-if)# switchport
switch-vtep-1(config-if)# switch port mode trunk
switch-vtep-1(config-if)# switch port allowed vlan 11-12
switch-vtep-1(config-if)# no shutdown
switch-vtep-1(config)# vlan 11
switch-vtep-1(config-vlan)# vn-segment 10011
switch-vtep-1(config)# vlan 12
switch-vtep-1(config-vlan)# vn-segment 10012
switch-vtep-1(config)# interface nve1
switch-vtep-1(config-if)# no shutdown
switch-vtep-1(config-if)# source-interface loopback0
switch-vtep-1(config-if)# member vni 10011
switch-vtep-1(config-if)# ingress-replication protocol static
switch-vtep-1(config-if)# peer_ip 200.200.9.9
switch-vtep-1(config-if)# member vni 10012
switch-vtep-1(config-if)# ingress-replication protocol static
switch-vtep-1(config-if)# peer_ip 200.200.9.9
switch-vtep-1(config-vlan)# exit

switch-vtep-1# show nve vni ingress-replication
Interface VNI show nve vni ingress-replication
Interface VNI Replication List Up Time
---------- ------- ----------------- -------
nve1 10011 200.200.9.9 07:39:51
nve1 10012 200.200.9.9 07:39:40
```

**Nexus 9000 VTEP-2 configuration:**

```bash
switch-vtep-2(config)# feature ospf
switch-vtep-2(config)# router ospf 1
switch-vtep-2(config-router)# router-id 200.200.9.9
switch-vtep-2(config)# interface loopback0
switch-vtep-2(config-if)# ip address 200.200.9.9/32
switch-vtep-2(config-if)# ip router ospf 1 area 0.0.0.0
switch-vtep-2(config)# interface e2/1
switch-vtep-2(config-if)# ip address 30.1.1.1/30
switch-vtep-2(config-if)# ip router ospf 1 area 0.0.0.0
switch-vtep-2(config-if)# ip pim sparse-mode
switch-vtep-2(config)# feature nv overlay
switch-vtep-2(config)# feature vn-segment-vlan-based
switch-vtep-2(config)# interface e1/1
switch-vtep-2(config-if)# switchport
switch-vtep-2(config-if)# switch port mode trunk
switch-vtep-2(config-if)# switch port allowed vlan 11-12
switch-vtep-2(config-if)# no shutdown
switch-vtep-2(config)# vlan 11
switch-vtep-2(config-vlan)# vn-segment 10011
switch-vtep-2(config)# vlan 12
switch-vtep-2(config-vlan)# vn-segment 10012
switch-vtep-2(config)# interface nve1
switch-vtep-2(config-if)# no shutdown
switch-vtep-2(config-if)# source-interface loopback0
switch-vtep-2(config-if)# member vni 10011
switch-vtep-2(config-if)# ingress-replication protocol static
switch-vtep-2(config-if)# peer_ip 200.200.8.8
```
switch-vtep-2(config-if)# member vni 10012
switch-vtep-2(config-if)# ingress-replication protocol static
switch-vtep-2(config-if)# peer_ip 200.200.8.8
switch-vtep-2(config-vlan)# exit

switch-vtep-2# show nve vni ingress-replication
Interface VNI Replication List Up Time
----------------- --------------- ---------------
nve1 10011 200.200.8.8 07:42:23
200.200.10.10 07:42:23
nve1 10012 200.200.8.8 07:42:23

• For a vPC VTEP configuration, the loopback address requires a secondary IP.

An example of a vPC VTEP configuration:

Figure 12: VXLAN topology for vPC VTEP

• Nexus 9000 VTEP-1 configuration:

  switch-vtep-1(config)# feature nv overlay
  switch-vtep-1(config)# feature vn-segment-vlan-based
  switch-vtep-1(config)# feature ospf
  switch-vtep-1(config)# feature pim
  switch-vtep-1(config)# router ospf 1
  switch-vtep-1(config-router)# router-id 200.200.200.1
  switch-vtep-1(config)# ip pim rp-address 10.1.1.1 group-list 224.0.0.0/4
  switch-vtep-1(config)# interface loopback0
  switch-vtep-1(config-if)# ip address 200.200.200.1/32
  switch-vtep-1(config-if)# ip address 100.100.100.1/32 secondary
  switch-vtep-1(config-if)# ip pim ospf 1 area 0.0.0.0
  switch-vtep-1(config-if)# ip pim sparse-mode
  switch-vtep-1(config)# interface e2/1
  switch-vtep-1(config-if)# ip address 20.1.1.1/30
  switch-vtep-1(config-if)# ip address 100.100.100.1/32 secondary
  switch-vtep-1(config-if)# ip router ospf 1 area 0.0.0.0
  switch-vtep-1(config-if)# ip pim sparse-mode
switch-vtep-1(config)# interface port-channel 10
switch-vtep-1(config-if)# vpc 10
switch-vtep-1(config-if)# switchport
switch-vtep-1(config-if)# switchport mode access
switch-vtep-1(config-if)# switchport access vlan 10
switch-vtep-1(config-if)# no shutdown
switch-vtep-1(config-if)# interface e1/1
switch-vtep-1(config-if)# channel-group 10 mode active
switch-vtep-1(config-if)# no shutdown
switch-vtep-1(config)# interface nve1
switch-vtep-1(config-if)# no shutdown
switch-vtep-1(config-if)# source-interface loopback0

switch-vtep-1(config-if)# member vni 10000 mcast-group 230.1.1.1
switch-vtep-1(config)# vlan 10
switch-vtep-1(config-vlan)# vn-segment 10000
switch-vtep-1(config)# exit

• Nexus 9000 VTEP-2 configuration:

switch-vtep-2(config)# feature nv overlay
switch-vtep-2(config)# feature vn-segment-vlan-based
switch-vtep-2(config)# feature ospf
switch-vtep-2(config)# feature pim
switch-vtep-2(config)# router ospf 1
switch-vtep-2(config-router)# router-id 200.200.200.2
switch-vtep-2(config)# ip pim rp-address 10.1.1.1 group-list 224.0.0.0/4
switch-vtep-2(config)# interface loopback0
switch-vtep-2(config-if)# ip address 200.200.200.2/32
switch-vtep-2(config-if)# ip address 100.100.100.1/32 secondary
switch-vtep-2(config-if)# ip router ospf 1 area 0.0.0.0
switch-vtep-2(config-if)# ip pim sparse-mode
switch-vtep-2(config)# interface e2/1
switch-vtep-2(config-if)# ip address 20.1.1.5/30
switch-vtep-2(config-if)# ip router ospf 1 area 0.0.0.0
switch-vtep-2(config-if)# ip pim sparse-mode
switch-vtep-2(config-if)# interface port-channel 10
switch-vtep-2(config-if)# vpc 10
switch-vtep-2(config-if)# switchport
switch-vtep-2(config-if)# switchport mode access
switch-vtep-2(config-if)# switchport access vlan 10
switch-vtep-2(config-if)# no shutdown
switch-vtep-2(config-if)# interface e1/1
switch-vtep-2(config-if)# channel-group 10 mode active
switch-vtep-2(config-if)# no shutdown
switch-vtep-2(config-if)# interface nve1
switch-vtep-2(config-if)# no shutdown
switch-vtep-2(config-if)# source-interface loopback0

switch-vtep-2(config-if)# member vni 10000 mcast-group 230.1.1.1
switch-vtep-2(config)# vlan 10
switch-vtep-2(config-vlan)# vn-segment 10000
switch-vtep-2(config)# exit

• Nexus 9000 VTEP-3 configuration:

switch-vtep-3(config)# feature nv overlay
switch-vtep-3(config)# feature vn-segment-vlan-based
switch-vtep-3(config)# feature ospf
switch-vtep-3(config)# feature pim
switch-vtep-3(config)# router ospf 1
switch-vtep-3(config-router)# router-id 100.100.100.2
switch-vtep-3(config)# ip pim rp-address 10.1.1.1 group-list 224.0.0.0/4
switch-vtep-3(config)# interface loopback0
switch-vtep-3(config-if)# ip address 100.100.100.2/32
switch-vtep-3(config-if)# ip router ospf 1 area 0.0.0.0
switch-vtep-3(config-if)# ip pim sparse-mode
switch-vtep-3(config)# interface e2/1
switch-vtep-3(config-if)# ip address 30.1.1.1/30
switch-vtep-3(config-if)# ip router ospf 1 area 0.0.0.0
switch-vtep-3(config-if)# ip pim sparse-mode

switch-vtep-3(config)# interface e1/1
switch-vtep-3(config-if)# switchport
switch-vtep-3(config-if)# switchport access vlan 10
switch-vtep-3(config-if)# no shutdown
switch-vtep-3(config)# interface nve1
switch-vtep-3(config-if)# no shutdown
switch-vtep-3(config-if)# source-interface loopback0

switch-vtep-3(config-if)# member vni 10000 mcast-group 230.1.1.1
switch-vtep-3(config)# vlan 10
switch-vtep-3(config-vlan)# vn-segment 10000
switch-vtep-3(config-vlan)# exit

Note The secondary IP is used by the emulated VTEP for VXLAN.

Note Ensure that all configurations are identical between the VPC primary and VPC secondary.
Configuring VXLAN BGP EVPN

This chapter contains the following sections:

- Information About VXLAN BGP EVPN, on page 69
- Configuring VXLAN BGP EVPN, on page 80
- Verifying the VXLAN BGP EVPN Configuration, on page 93
- Example of VXLAN BGP EVPN (EBGP), on page 94
- Example of VXLAN BGP EVPN (IBGP), on page 103
- Example Show Commands, on page 111

Information About VXLAN BGP EVPN

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:
  - 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 7.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 is not applicable to the Cisco Nexus 9200 and 9300-EX platform switches and Cisco Nexus 9500 platform switches with 9700-EX line cards.

- Beginning with Cisco NX-OS Release 7.0(3)F3(3), VXLAN Layer 2 Gateway is supported only on the 9636C-RX line card. VXLAN and MPLS cannot be enabled on the Cisco Nexus 9508 switch at the same time.

- Beginning with Cisco NX-OS Release 7.0(3)F3(3), if VXLAN is enabled, the Layer 2 Gateway cannot be enabled when there is any line card other than the 9636C-RX.

- Beginning with Cisco NX-OS Release 7.0(3)I6(1), you can configure EVPN over segment routing or MPLS. See the Cisco Nexus 9000 Series NX-OS Label Switching Configuration Guide, Release 7.x for more information.
• Beginning with Cisco NX-OS Release 7.0(3)I6(1), 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 7.x for more information.

• In VXLAN EVPN setup that has 2K VNI scale configuration, the control plane down time takes more than 200 seconds. To avoid BGP flap, configure the graceful restart time to 300 seconds.

• SVI and subinterfaces as uplinks are not supported.

• In a VXLAN EVPN setup, border leaves must use unique route distinguishers, preferably using auto rd command. It is not supported to have same route distinguishers in different border leaves.

• ARP suppression is only supported for a VNI if the VTEP hosts the First-Hop Gateway (Distributed Anycast Gateway) for this VNI. The VTEP and the SVI for this VLAN have to be properly configured for the distributed Anycast Gateway operation, for example, global Anycast Gateway MAC address configured and Anycast Gateway feature with the virtual IP address on the SVI.

• 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 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 for other guidelines and limitations for the VXLAN QoS feature.

• The QoS buffer-boost feature is not applicable for VXLAN traffic.

• VTEP does not support Layer 3 subinterface uplinks that carry VXLAN encapsulated traffic.

• Layer 3 interface uplinks that carry VXLAN encapsulated traffic do not support subinterfaces for non-VxLAN encapsulated traffic.

• For 7.0(3)I2(1) and later, a FEX HIF (FEX host interface port) is supported for a VLAN that is extended with VXLAN.

• For eBGP, it is recommended to use a single overlay eBGP EVPN session between loopbacks.

• EBGP peering from a VXLAN host to local VTEP is supported with loopback in tenant VRF as BGP update-source.

• Bind NVE to a loopback address that is separate from other loopback addresses that are required by Layer 3 protocols. A best practice is to use a dedicated loopback address for VXLAN.

• VXLAN BGP EVPN does not support an NVE interface in a non-default VRF.

• It is recommended to configure a single BGP session over the loopback for an overlay BGP session.

• When configuring VXLAN BGP EVPN, only the "System Routing Mode: Default" is applicable for the following hardware platforms:
- Cisco Nexus 9200, 9300-EX, 9300-FX, and 9300-FX2
- Cisco Nexus 9300 platform switches
- Cisco Nexus 9500 platform switches with X9500 line cards
- Cisco Nexus 9500 platform switches with 9700-EX, 9700-FX, and 9700-FX2 line cards

- The “System Routing Mode: template-vxlan-scale” is not applicable to Cisco NX-OS Release 7.0(3)I5(2) and later.
- When using VXLAN BGP EVPN 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.
- For 7.0(3)I2(1) and later, VXLAN is supported on Cisco Nexus 9500 platform switches with the following line cards:
  - 9500-R
  - 9564PX
  - 9564TX
  - 9536PQ
  - 9700-EX
  - 9700-FX

- When Cisco Nexus 9500 platform switches are used as VTEPs(7.0(3)I2(1) and later), 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. (7.0(3)I2(1) and later)

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

---

**Note** For VXLAN routing support, a 40G 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 7.0(3)I1(2) and earlier, a static route with next-hop reachable over the VXLAN BGP EVPN route is not supported.
- 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. (7.0(3)I3(1) and later).
• For the NX-OS 7.0(3)I2(3) release, the VXLAN network identifier (VNID) 16777215 is reserved and should not be configured explicitly.

• For 7.0(3)I4(1) and later, VXLAN supports In Service Software Upgrade (ISSU).

• VXLAN does not support co-existence with the GRE tunnel feature or the MPLS (static or segment-routing) feature on Cisco Nexus 9000 Series switches with a Network Forwarding Engine (NFE).

• VTEP connected to FEX host interface ports is not supported (7.0(3)I2(1) and later).

• In Cisco NX-OS Release 7.0(3)I4(1), 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.

---

**Considerations for VXLAN BGP EVPN Deployment**

• A loopback address is required when using the `source-interface config` command. The loopback address represents the local VTEP IP.

• During boot-up of a switch (7.0(3)I2(2) and later), you can use the `source-interface hold-down-time` command to suppress advertisement of the NVE loopback address until the overlay has converged. The range for the `hold-down-time` is 0 - 1000 seconds. The default is 180 seconds.

• To establish IP multicast routing in the core, IP multicast configuration, PIM configuration, and RP configuration is required.

• VTEP to VTEP unicast reachability can be configured through any IGP/BGP protocol.

• If the anycast gateway feature is enabled for a specific VNI, then the anyway gateway feature must be enabled on all VTEPs that have that VNI configured. Having the anycast gateway feature configured on only some of the VTEPs enabled for a specific VNI is not supported.

• It is a requirement when changing the primary or secondary IP address of the NVE source interfaces to shut the NVE interface before changing the IP address.

• As a best practice, the RP for the multicast group should be configured only on the spine layer. Use the anycast RP for RP load balancing and redundancy.

• Every tenant VRF needs a VRF overlay VLAN and SVI for VXLAN routing.

• 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 9200 platform switches beginning with the Cisco NX-OS Release 7.0(3)I1(2) through 7.0(3)I5(2)
• Cisco Nexus 9300-EX, 9300-FX, and 9300-FX2 platforms switches beginning with the Cisco NX-OS Release 7.0(3)I1(2) through 7.0(3)I5(2)

• Cisco Nexus 9300 platforms switches beginning with the Cisco NX-OS Release 7.0(3)I1(2)

**Note**
Beginning with the Cisco NX-OS Release 7.0(3)I5(2), the Cisco Nexus 9200, Cisco Nexus 9300-EX, 9300-FX, and 9300-FX2 do not require this command. Beginning with Cisco NX-OS Release 7.0(3)I5(2), it is strongly recommended to remove the command on Cisco Nexus 9200, 9300-EX, 9300-FX, and 9300-FX2 platform switches as it would disable Tenant Routed Multicast functionality on the VRF.

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

**vPC Considerations for VXLAN BGP EVPN Deployment**

• The loopback address used by NVE needs to be configured to have a primary IP address and a secondary IP address.
  The secondary IP address is used for all VxLAN traffic that includes multicast and unicast encapsulated traffic.

• Each vPC peer needs to have separate BGP sessions to the spine.

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

• Consistent VNI to group mapping.

• The VRF overlay VLAN should be a member of the peer-link port-channel.

• For multicast, the vPC node that receives the (S, G) join from the RP (rendezvous point) becomes the DF (designated forwarder). On the DF node, encap routes are installed for multicast.

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

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

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

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

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

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

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

• When peer-link is no-shut, the NVE loopback address is brought up again and the route is advertised upstream, attracting traffic.

• For vPC, the loopback interface has 2 IP addresses: the primary IP address and the secondary IP address.

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

The secondary IP address on loopback is necessary because the interface NVE uses it for the VTEP IP address. The secondary IP address must be same on both vPC peers.

• The vPC peer-gateway feature must be enabled on both peers.

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

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

The following is an example (best practice) of a vPC configuration:
On a vPC pair, shutting down NVE or NVE loopback on one of the vPC nodes is not a supported configuration. This means that traffic failover on one-side NVE shut or one-side loopback shut is not supported.

- Redundant anycast RPs configured in the network for multicast load-balancing and RP redundancy are supported on vPC VTEP topologies.

- Enabling vpc peer-gateway configuration is mandatory. For peer-gateway functionality, at least one backup routing SVI is required to be enabled across peer-link and also configured with PIM. This provides a backup routing path in the case when VTEP loses complete connectivity to the spine. Remote peer reachability is re-routed over the peer-link in this case.

The following is an example of SVI with PIM enabled:

```
switch# sh ru int vlan 2
interface Vlan2
description special_svi_over_peer-link
no shutdown
ip address 30.2.1.1/30
ip pim sparse-mode
```

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

- As a best practice when changing the secondary IP address of an anycast vPC VTEP, the NVE interfaces on both the vPC primary and the vPC secondary should be shut before the IP changes are made.

- To provide redundancy and failover of VXLAN traffic when a VTEP loses all of its uplinks to the spine, it is recommended to run a Layer 3 link or an SVI link over the peer-link between vPC peers.

- If DHCP Relay is required in VRF for DHCP clients or if loopback in VRF is required for reachability test on a vPC pair, it is necessary to create a backup SVI per VRF with PIM enabled.

```
switch# sh ru int vlan 20
interface Vlan20
description backup routing svi for VRF Green
vrf member GREEN
no shutdown
ip address 30.2.10.1/30
```
Network Considerations for VXLAN Deployments

- **MTU Size in the Transport Network**
  
  Due to the MAC-to-UDP encapsulation, VXLAN introduces 50-byte overhead to the original frames. Therefore, the maximum transmission unit (MTU) in the transport network must be increased by 50 bytes. If the overlays use a 1500-byte MTU, the transport network must 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 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 number 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 multitenant VXLAN networks to share a multicast group does not bring any implications to the Layer 2 isolation between the tenant networks. After receiving an encapsulated packet from the multicast group, a VTEP checks and validates the VNID in the VXLAN header of the packet. The VTEP discards the packet if the VNID is unknown to it. Only when the VNID matches one of the VTEP’s local VXLAN VNIDs, does it forward the packet to that VXLAN segment. Other tenant networks will not receive the packet. Thus, the segregation between VXLAN segments is not compromised.

Considerations for the Transport Network

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

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

  (For vPC VTEPs, you must configure primary and secondary /32 IP addresses.)

  - Enable UP multicast on the loopback interface. *
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, and 9300-FX2 platform switches.

Figure 13: 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 (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 of a Layer 3 port-channel interface

## BGP EVPN Considerations for VXLAN Deployment

### Commands for BGP EVPN

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

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>member vni range [associate-vrf]</td>
<td>Associate VXLAN VNI (Virtual Network Identifiers) with the NVE interface. The attribute <code>associate-vrf</code> is used to identify and separate processing VNI that are associated with a VRF and used for routing. <strong>Note</strong> The VRF and VNI specified with this command must match the configuration of the VNI under the VRF.</td>
</tr>
<tr>
<td>show nve vni</td>
<td>Displays information that determine if the VNI is configured for peer and host learning via the control plane or data plane.</td>
</tr>
<tr>
<td>show nve vni summary</td>
<td>Displays the Layer 2 VNE VPN address family.</td>
</tr>
<tr>
<td>show bgp l2vpn evpn</td>
<td>Specifies BGP as the mechanism for host reachability advertisement.</td>
</tr>
<tr>
<td>show bgp l2vpn evpn summary</td>
<td>Suppresses ARP under Layer 2 VNI.</td>
</tr>
<tr>
<td>host-reachability protocol bgp</td>
<td>Configures anycast gateway MAC of the switch.</td>
</tr>
<tr>
<td>suppress-arp</td>
<td>Creates the VRF and enter the VRF mode.</td>
</tr>
<tr>
<td>fabric forwarding anycast-gateway-mac</td>
<td>Enables/Disables the Ethernet VPN (EVPN).</td>
</tr>
<tr>
<td>vrf context</td>
<td>Configures the Border Gateway Protocol (BGP).</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| `system vlan nve-overlay id range` | 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 9200 platform switches beginning with the Cisco NX-OS Release 7.0(3)I1(2) through 7.0(3)I5(2)
- Cisco Nexus 9300-EX, 9300-FX, and 9300-FX2 platforms switches beginning with the Cisco NX-OS Release 7.0(3)I1(2) through 7.0(3)I5(2)
- Cisco Nexus 9300 platforms switches beginning with the Cisco NX-OS Release 7.0(3)I1(2)
- Cisco Nexus 9500 platforms switches with -EX and -FX line cards. |

**Note** Beginning with the Cisco NX-OS Release 7.0(3)I5(2), the Cisco Nexus 9200, Cisco Nexus 9300-EX, 9300-FX, and 9300-FX2 do not require this command. Beginning with Cisco NX-OS Release 7.0(3)I5(2), it is strongly recommended to remove the command on Cisco Nexus 9200, 9300-EX, 9300-FX, and 9300-FX2 platform switches as it would disable Tenant Routed Multicast functionality on the VRF. |

**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). |
### Configuring VXLAN BGP EVPN

#### Enabling VXLAN

Enable VXLAN and the EVPN.

**Procedure**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>feature vn-segment</td>
<td>Enable VLAN-based VXLAN</td>
</tr>
<tr>
<td>2</td>
<td>feature nv overlay</td>
<td>Enable VXLAN</td>
</tr>
<tr>
<td>3</td>
<td>nv overlay evpn</td>
<td>Enable the EVPN control plane for VXLAN</td>
</tr>
</tbody>
</table>

#### Command and Description

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>suppress mac-route</td>
<td>Suppresses the BGP MAC route so that BGP only sends the MAC/IP route for a host. Under NVE, the MAC updates for all VNIs are suppressed.</td>
</tr>
</tbody>
</table>

**Note**

- **Receive-side** — Suppressing the MAC route depends upon the capability of the remote EVPN peer to derive a MAC route from the MAC/IP route (7.0(3)I2(2) and later). Avoid using the “suppress mac-route” command if devices in the network are running an earlier NX-OS release.

- **Send-side** — Suppressing the MAC route means that the sender has a MAC/IP route. If your configuration has pure-Layer 2 VNIs (such as no corresponding VRF or Layer3-VNI), then there is no corresponding MAC/IP and you should avoid using the “suppress mac-route” command.
### Configuring VLAN and VXLAN VNI

**Procedure**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> <code>vlan number</code></td>
<td>Specify VLAN.</td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>vn-segment number</code></td>
<td>Map VLAN to VXLAN VNI to configure Layer 2 VNI under VXLAN VLAN.</td>
</tr>
</tbody>
</table>

### Configuring VRF for VXLAN Routing

Configure the tenant VRF.

**Procedure**

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

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.
## Configuring SVI for Hosts for VXLAN Routing

Configure the SVI for hosts.

### Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 vlan number</td>
<td>Specify VLAN</td>
</tr>
<tr>
<td>Step 2 interface vlan-number</td>
<td>Specify VLAN interface.</td>
</tr>
<tr>
<td>Step 3 vrf member vxlan-number</td>
<td>Configure SVI for host.</td>
</tr>
<tr>
<td>Step 4 ip address address</td>
<td>Specify IP address.</td>
</tr>
</tbody>
</table>

## Configuring VRF Overlay VLAN for VXLAN Routing

### Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 vlan number</td>
<td>Specify VLAN.</td>
</tr>
<tr>
<td>Step 2 vn-segment number</td>
<td>Specify vn-segment.</td>
</tr>
</tbody>
</table>

## Configuring VNI Under VRF for VXLAN Routing

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

### Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 vrf context vxlan</td>
<td>Create a VXLAN Tenant VRF</td>
</tr>
<tr>
<td>Step 2 vni number</td>
<td>Configure Layer 3 VNI under VRF.</td>
</tr>
</tbody>
</table>

---

Auto derived Route-Targets for a 4-byte ASN are not supported.
## Configuring Anycast Gateway for VXLAN Routing

**Procedure**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Configure distributed gateway virtual MAC address.</td>
</tr>
<tr>
<td>fabric forwarding anycast-gateway-mac address</td>
<td></td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>One virtual MAC per VTEP</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>All VTEPs must have the same virtual MAC address.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Associate SVI with Anycast Gateway under VLAN configuration mode.</td>
</tr>
<tr>
<td>fabric forwarding mode anycast-gateway</td>
<td></td>
</tr>
</tbody>
</table>

## Configuring the NVE Interface and VNIs

**Procedure**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Configure the NVE interface.</td>
</tr>
<tr>
<td>interface nve-interface</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>This defines BGP as the mechanism for host reachability advertisement</td>
</tr>
<tr>
<td>host-reachability protocol bgp</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Add Layer-3 VNIs, one per tenant VRF, to the overlay.</td>
</tr>
<tr>
<td>member vni vni associate-vrf</td>
<td></td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>Required for VXLAN routing only.</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Add Layer 2 VNIs to the tunnel interface.</td>
</tr>
<tr>
<td>member vni vni</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>Configure the mcast group on a per-VNI basis</td>
</tr>
<tr>
<td>mcast-group address</td>
<td></td>
</tr>
</tbody>
</table>

## Configuring BGP on the VTEP

**Procedure**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Configure BGP.</td>
</tr>
<tr>
<td>router bgp number</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Specify router address.</td>
</tr>
<tr>
<td>router-id address</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Define MP-BGP neighbors. Under each neighbor define l2vpn evpn.</td>
</tr>
<tr>
<td>neighbor address remote-as number</td>
<td></td>
</tr>
</tbody>
</table>
### Configuring RD and Route Targets for VXLAN Bridging

#### Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 evpn</td>
<td>Configure VRF.</td>
</tr>
<tr>
<td>Step 2 vni number 12</td>
<td><strong>Note</strong> Only Layer 2 VNIs need to be specified.</td>
</tr>
<tr>
<td>Step 3 rd auto</td>
<td>Define VRF RD (route distinguisher) to configure VRF context.</td>
</tr>
<tr>
<td>Step 4 route-target import auto</td>
<td>Define VRF Route Target and import policies.</td>
</tr>
<tr>
<td>Step 5 route-target export auto</td>
<td>Define VRF Route Target and export policies.</td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 4 address-family ipv4 unicast</td>
<td>Configure address family for IPv4.</td>
</tr>
<tr>
<td>Step 5 address-family l2vpn evpn</td>
<td>Configure address family Layer 2 VPN EVPN under the BGP neighbor. <strong>Note</strong> Address-family ipv4 evpn for vxlan host-based routing</td>
</tr>
<tr>
<td>Step 6 (Optional) Allowas-in</td>
<td>Allows duplicate AS numbers in the AS path. Configure this parameter on the leaf for eBGP when all leaves are using the same AS, but the spines have a different AS than leaves.</td>
</tr>
<tr>
<td>Step 7 send-community extended</td>
<td>Configures community for BGP neighbors.</td>
</tr>
<tr>
<td>Step 8 vrf vrf-name</td>
<td>Specify VRF.</td>
</tr>
<tr>
<td>Step 9 address-family ipv4 unicast</td>
<td>Configure address family for IPv4.</td>
</tr>
<tr>
<td>Step 10 advertise l2vpn evpn</td>
<td>Enable advertising EVPN routes.</td>
</tr>
<tr>
<td>Step 11 address-family ipv6 unicast</td>
<td>Configure address family for IPv6.</td>
</tr>
<tr>
<td>Step 12 advertise l2vpn evpn</td>
<td>Enable advertising EVPN routes. <strong>Note</strong> To disable advertisement for a VRF toward the EVPN, disable the VNI in NVE by entering the <strong>no member vni vni associate-vrf</strong> command in interface nve1. The vni is the VNI associated with that particular VRF.</td>
</tr>
</tbody>
</table>
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. 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. 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.

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

Note

Auto derived Route-Targets for a 4-byte ASN are not supported.
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.

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

Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> interface nve-interface</td>
<td>Configure the NVE interface.</td>
</tr>
<tr>
<td><strong>Step 2</strong> host-reachability protocol bgp</td>
<td>This defines BGP as the mechanism for host reachability advertisement</td>
</tr>
<tr>
<td><strong>Step 3</strong> member vni vni associate-vrf</td>
<td>Add Layer-3 VNIs, one per tenant VRF, to the overlay.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> Required for VXLAN routing only.</td>
</tr>
<tr>
<td><strong>Step 4</strong> member vni vni</td>
<td>Add Layer 2 VNIs to the tunnel interface.</td>
</tr>
<tr>
<td><strong>Step 5</strong> ingress-replication protocol bgp</td>
<td>Enables 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.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> Using <code>ingress-replication protocol bgp</code> avoids the need for any multicast configurations that might have been required for configuring the underlay.</td>
</tr>
</tbody>
</table>
## Configuring BGP for EVPN on the Spine

### Procedure

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>route-map permitall permit 10</td>
<td>Configure route-map.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
<td>The route-map keeps the next-hop unchanged for EVPN routes.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Required for eBGP.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Optional for iBGP.</td>
</tr>
<tr>
<td>2</td>
<td>set ip next-hop unchanged</td>
<td>Set next-hop address.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
<td>The route-map keeps the next-hop unchanged for EVPN routes.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Required for eBGP.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Optional for iBGP.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
<td>When two next hops are enabled, next hop ordering is not maintained.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>3</td>
<td>router bgp autonomous system number</td>
<td>Specify BGP.</td>
</tr>
<tr>
<td>4</td>
<td>address-family l2vpn evpn</td>
<td>Configure address family Layer 2 VPN EVPN under the BGP neighbor.</td>
</tr>
<tr>
<td>5</td>
<td>retain route-target all</td>
<td>Configure retain route-target all under address-family Layer 2 VPN EVPN [global].</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
<td>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.</td>
</tr>
<tr>
<td>6</td>
<td>neighbor address remote-as number</td>
<td>Define neighbor.</td>
</tr>
</tbody>
</table>
Configure address family Layer 2 VPN EVPN under the BGP neighbor.

### Step 8
`disable-peer-as-check`
Disables checking the peer AS number during route advertisement. Configure this parameter on the spine for eBGP when all leafs are using the same AS but the spines have a different AS than leafs.

**Note** Required for eBGP.

### Step 9
`send-community extended`
Configures community for BGP neighbors.

### Step 10
`route-map permit all out`
Applies route-map to keep the next-hop unchanged.

**Note** Required for eBGP.

### Suppressing ARP

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

**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.

#### Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>Purpose</strong></td>
</tr>
<tr>
<td><code>hardware access-list tcam region arp-ether</code></td>
<td>Configure TCAM region to suppress ARP.</td>
</tr>
<tr>
<td><code>size double-wide</code></td>
<td><code>tcam-size</code>—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.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> Reload is required for the TCAM configuration to be in effect.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> Configuring the <code>hardware access-list tcam region arp-ether</code></td>
</tr>
<tr>
<td></td>
<td><code>size double-wide</code> is not required on Cisco Nexus 9200 Series switches.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><strong>Purpose</strong></td>
</tr>
<tr>
<td><code>interface nve 1</code></td>
<td>Create the network virtualization endpoint (NVE) interface.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><strong>Purpose</strong></td>
</tr>
<tr>
<td><code>member vni vni-id</code></td>
<td>Specify VNI ID.</td>
</tr>
</tbody>
</table>
### Disabling VXLANs

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 4</td>
<td>suppress-arp</td>
<td>Configure to suppress ARP under Layer 2 VNI.</td>
</tr>
<tr>
<td>Step 5</td>
<td>copy running-config start-up-config</td>
<td>Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.</td>
</tr>
</tbody>
</table>

### Duplicate Detection for IP and MAC Addresses

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

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

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

- For MAC addresses:
  - After the 5th move within 180 seconds, the switch starts a 30 second lock (hold down timer) before checking to see if the duplication still exists (an effort to prevent an increment of the sequence bit). This 30 second lock can occur 3 times within 24 hours (this means 5 moves in 180 seconds for 3 times) before the switch permanently locks or freezes the duplicate entry. (*show l2rib internal permanently-frozen-list*)

- Wherever a MAC address is permanently frozen, a syslog message with written by L2RIB.
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:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>switch(config)# fabric forwarding ?</td>
<td>Available sub-commands:</td>
</tr>
<tr>
<td>switch(config)# fabric forwarding</td>
<td>• Anycast gateway MAC of the</td>
</tr>
<tr>
<td>switch(config)# fabric forwarding</td>
<td>• To detect duplicate host addresses</td>
</tr>
<tr>
<td>switch(config)# fabric forwarding</td>
<td>in n seconds.</td>
</tr>
<tr>
<td>switch(config)# fabric forwarding</td>
<td>The number of host moves allowed in n</td>
</tr>
<tr>
<td>switch(config)# fabric forwarding</td>
<td>seconds. The range is 1 to 1000 moves;</td>
</tr>
<tr>
<td>switch(config)# fabric forwarding</td>
<td>default is 5 moves.</td>
</tr>
<tr>
<td>switch(config)# fabric forwarding</td>
<td>The duplicate detection timeout in</td>
</tr>
<tr>
<td>switch(config)# fabric forwarding</td>
<td>seconds for the number of host moves.</td>
</tr>
<tr>
<td>switch(config)# fabric forwarding</td>
<td>The range is 2 to 36000 seconds; default</td>
</tr>
<tr>
<td>switch(config)# fabric forwarding</td>
<td>is 180 seconds.</td>
</tr>
<tr>
<td>switch(config)# fabric forwarding</td>
<td>Detects duplicate host addresses (limited</td>
</tr>
<tr>
<td>switch(config)# fabric forwarding</td>
<td>to 100 moves) in a period of 10 seconds.</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>switch(config)# l2rib dup-host-mac-detection ?</td>
<td>Available sub-commands for L2RIB:</td>
</tr>
<tr>
<td>switch(config)# l2rib dup-host-mac-detection</td>
<td>• The number of host moves allowed</td>
</tr>
<tr>
<td>switch(config)# l2rib dup-host-mac-detection</td>
<td>in n seconds. The range is 1 to 1000</td>
</tr>
<tr>
<td>switch(config)# l2rib dup-host-mac-detection</td>
<td>moves.</td>
</tr>
<tr>
<td>switch(config)# l2rib dup-host-mac-detection</td>
<td>• Default setting (5 moves in 180 in</td>
</tr>
<tr>
<td>switch(config)# l2rib dup-host-mac-detection</td>
<td>seconds).</td>
</tr>
<tr>
<td>switch(config)# l2rib dup-host-mac-detection 100 ?</td>
<td>The duplicate detection timeout in</td>
</tr>
<tr>
<td>switch(config)# l2rib dup-host-mac-detection 100</td>
<td>seconds for the number of host moves.</td>
</tr>
<tr>
<td>switch(config)# l2rib dup-host-mac-detection 100</td>
<td>The range is 2 to 36000 seconds; default</td>
</tr>
<tr>
<td>switch(config)# l2rib dup-host-mac-detection 100</td>
<td>is 180 seconds.</td>
</tr>
</tbody>
</table>
Enabling Nuage Controller Interoperability

The following steps enable Nuage controller interoperability.

**Procedure**

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

**Example**

The following is an example to enable Nuage controller interoperability:

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

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

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

Cisco Nexus 9000 Series NX-OS VXLAN Configuration Guide, Release 7.x
Verifying the VXLAN BGP EVPN Configuration

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

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>show nve vrf</code></td>
<td>Displays VRFs and associated VNIs</td>
</tr>
<tr>
<td><code>show bgp l2vpn evpn</code></td>
<td>Displays routing table information.</td>
</tr>
<tr>
<td>`show ip arp suppression-cache [detail</td>
<td>summary</td>
</tr>
<tr>
<td><code>show vxlan interface</code></td>
<td>Displays VXLAN interface status.</td>
</tr>
<tr>
<td>`show vxlan interface</td>
<td>count`</td>
</tr>
<tr>
<td>`show l2route evpn mac [all</td>
<td>evi evi [bgp</td>
</tr>
<tr>
<td><code>show l2route evpn fl all</code></td>
<td>Displays all fl routes.</td>
</tr>
<tr>
<td><code>show l2route evpn imet all</code></td>
<td>Displays all imet routes.</td>
</tr>
<tr>
<td><code>show l2route evpn mac-ip all</code></td>
<td>Displays all MAC IP routes.</td>
</tr>
<tr>
<td><code>show l2route evpn mac-ip all detail</code></td>
<td>Displays all MAC IP routes.</td>
</tr>
<tr>
<td><code>show l2route topology</code></td>
<td>Displays Layer 2 route topology.</td>
</tr>
</tbody>
</table>

---

**Note** Although the `show ip bgp` command is available for verifying a BGP configuration, as a best practice, it is preferable to use the `show bgp` command instead.
Example of VXLAN BGP EVPN (EBGP)

An example of a VXLAN BGP EVPN (EBGP):

*Figure 14: 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 BGP
    ```
    interface loopback0
    ip address 10.1.1.1/32
    ip pim sparse-mode
    ```
- Configure Loopback for Anycast RP
  ```
  interface loopback1
  ```
ip address 100.1.1.1/32
ip pim sparse-mode

• Configure Anycast RP

ip pim rp-address 100.1.1.1 group-list 224.0.0.0/4
ip pim ssm range 232.0.0.0/8
ip pim anycast-rp 100.1.1.1 10.1.1.1
ip pim anycast-rp 100.1.1.1 20.1.1.1

• Configure route-map used by EBGP for Spine

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

• Configure interfaces for Spine-leaf interconnect

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

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

• Configure the BGP overlay for the EVPN address family.

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

• Configure the BGP underlay.

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

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

```bash
nv overlay evpn
feature bgp
feature pim
```

• Configure Anycast RP

```bash
ip pim rp-address 100.1.1.1 group-list 224.0.0.0/4
ip pim ssm range 232.0.0.0/8
ip pim anycast-rp 100.1.1.1 10.1.1.1
ip pim anycast-rp 100.1.1.1 20.1.1.1
```

• Configure route-map used by EBGP for Spine

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

• Configure interfaces for Spine-leaf interconnect

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

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

• Configure Loopback for BGP

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

• Configure Loopback for Anycast RP

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

• Configure the BGP overlay for the EVPN address family.

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

• Configure the BGP underlay.

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

• Leaf (9396-A)

• Enable the EVPN control plane
  nv overlay evpn

• Enable the relevant protocols

  feature bgp
  feature pim
  feature interface-vlan

• Enable VXLAN with distributed anycast-gateway using BGP EVPN

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

• Enable PIM RP

  ip pim rp-address 100.1.1.1 group-list 224.0.0.0/4
  ip pim ssm range 232.0.0.0/8

• Create VLANs

  vlan 1-1002

• Configure Loopback for BGP

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

• Configure Loopback for local VTEP IP

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

• Configure interfaces for Spine-leaf interconnect
interface Ethernet2/2
  ip address 192.168.1.22/24
  ip pim sparse-mode
  no shutdown

interface Ethernet2/3
  ip address 192.168.3.23/24
  ip pim sparse-mode
  no shutdown

- Create the VRF overlay VLAN and configure the vn-segment.
  vlan 101
  vn-segment 900001

- Configure VRF overlay VLAN/SVI for the VRF
  interface Vlan101
    no shutdown
    vrf member vxlan-900001
    ip forward

- Create VLAN and provide mapping to VXLAN
  vlan 1001
    vn-segment 2001001
  vlan 1002
    vn-segment 2001002

- Create VRF and configure VNI
  vrf context vxlan-900001
  vni 900001
  rd auto
  address-family ipv4 unicast
    route-target import 65535:101 evpn
    route-target export 65535:101 evpn
    route-target import 65535:101
    route-target export 65535:101
  address-family ipv6 unicast
    route-target import 65535:101 evpn
    route-target export 65535:101 evpn
    route-target import 65535:101
    route-target export 65535:101

- Create server facing SVI and enable distributed anycast-gateway
  interface Vlan1001
    no shutdown
    vrf member vxlan-900001
    ip address 4.1.1.1/24
    ipv6 address 4::1/64
    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, 9300-FX, and 9300-FX2.

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

• Create the network virtualization endpoint (NVE) interface

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

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

evpn
  vni 2001001 12
  vni 2001002 12
rd auto
route-target import auto
route-target export auto

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

    nv overlay evpn
    feature bgp
    feature pim
    feature interface-vlan
    feature vn-segment-vlan-based
    feature nv overlay

• Enable PIM RP

    ip pim rp-address 100.1.1.1 group-list 224.0.0.0/4
    ip pim ssm range 232.0.0.0/8

• Enable VXLAN with distributed anycast-gateway using BGP EVPN

    fabric forwarding anycast-gateway-mac 0000.2222.3333

• Create VLANs

    vlan 1-1002

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

    vlan 101
      vn-segment 900001

• Create VLAN and provide mapping to VXLAN

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

• Create VRF and configure VNI

    vrf context vxlan-900001
      vni 900001

    rd auto
address-family ipv4 unicast
  route-target import 65535:101 evpn
  route-target export 65535:101 evpn
  route-target import 65535:101
  route-target export 65535:101
address-family ipv6 unicast
  route-target import 65535:101 evpn
  route-target export 65535:101 evpn
  route-target import 65535:101
  route-target export 65535:101

• 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, 9300-FX, and 9300-FX2 switches.

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

• Configure internal control VLAN/SVI for the VRF

interface Vlan101
  no shutdown
  vrf member vxlan-900001
  ip forward

• Create server facing SVI and enable distributed anycast-gateway

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

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

• Create the network virtualization endpoint (NVE) interface

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

• Configure interfaces for hosts/servers
interface Ethernet1/47
  switchport
  switchport access vlan 1002

interface Ethernet1/48
  switchport
  switchport access vlan 1001

• Configure interfaces for Spine-leaf interconnect

interface Ethernet2/2
  ip address 192.168.4.22/24
  ip pim sparse-mode
  no shutdown

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

• Configure Loopback for BGP

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

• Configure Loopback for local VTEP IP

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

• Configure BGP

router bgp 200
router-id 40.1.1.1
  neighbor 10.1.1.1 remote-as 100
    update-source loopback0
    ebgp-multihop 3
    allowas-in
    send-community extended
    address-family l2vpn
      allowas-in
      send-community extended
  neighbor 20.1.1.1 remote-as 100
    update-source loopback0
    ebgp-multihop 3
    allowas-in
    send-community extended
    address-family l2vpn
      allowas-in
      send-community extended
  vrf vxlan-900001
    advertise l2vpn evpn
Example of VXLAN BGP EVPN (IBGP)

An example of a VXLAN BGP EVPN (IBGP):

*Figure 15: VXLAN BGP EVPN Topology (IBGP)*

**IBGP between Spine and Leaf**

- Spine (9504-A)
  - Enable the EVPN control plane
    
    ```
    nv overlay evpn
    ```

  - Enable the relevant protocols

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

  - Configure Loopback for local VTEP IP, and BGP

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

- Configure Loopback for Anycast RP
**Example of VXLAN BGP EVPN (IBGP)**

```
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 ssm range 232.0.0.0/8
  ip pim anycast-rp 100.1.1.1 10.1.1.1
  ip pim anycast-rp 100.1.1.1 20.1.1.1

• Enable OSPF for underlay routing

  router ospf 1

• Configure interfaces for Spine-leaf interconnect

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

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

• Configure BGP

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

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

    nv overlay evpn
    feature ospf
    feature bgp
    feature pim

• Configure Anycast RP

  ip pim rp-address 100.1.1.1 group-list 224.0.0.0/4
```
ip pim ssm range 232.0.0.0/8
ip pim anycast-rp 100.1.1.1 10.1.1.1
ip pim anycast-rp 100.1.1.1 20.1.1.1

• Configure interfaces for Spine-leaf interconnect

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

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

• Configure Loopback for local VTEP IP, and BGP

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

• Configure Loopback for Anycast RP

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

• Enable OSPF for underlay routing

router ospf 1

• Configure BGP

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

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

  nv overlay evpn
  • Enable the relevant protocols
feature ospf
feature bgp
feature pim
feature interface-vlan

• Enable VXLAN with distributed anycast-gateway using BGP EVPN

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

• Enabling OSPF for underlay routing

  router ospf 1

• Configure Loopback for local VTEP IP, and BGP

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

• Configure interfaces for Spine-leaf interconnect

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

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

• Configure PIM RP

  ip pim rp-address 100.1.1.1 group-list 224.0.0.0/4
  ip pim ssm range 232.0.0.0/8

• Create VLANs

  vlan 1-1002

• Create overlay VRF VLAN and configure vn-segment

  vlan 101
    vn-segment 900001

• Configure VRF overlay VLAN/SVI for the VRF

  interface Vlan101
    no shutdown
vrf member vxlan-900001
ip forward

• Create VLAN and provide mapping to VXLAN

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

• Create VRF and configure VNI

vrf context vxlan-900001
  vni 900001
  rd auto
    address-family ipv4 unicast
    route-target both auto
    route-target both auto evpn
    address-family ipv6 unicast
    route-target both auto
    route-target both auto evpn

• Create server facing SVI and enable distributed anycast-gateway

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

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

• Configure ACL TCAM region for ARP suppression

Note: The `hardware access-list tcam region arp-ether 256 double-wide` command is not needed for Cisco Nexus 9300-EX, 9300-FX, and 9300-FX2.

• Create the network virtualization endpoint (NVE) interface

interface nve1
  no shutdown
  source-interface loopback0
  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

• 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  
    advertise l2vpn evpn
evpn  
  vni 2001001 12  
  vni 2001002 12
rd auto  
  route-target import auto  
  route-target export auto

• Leaf (9396-B)

  • Enable the EVPN control plane functionality and the relevant protocols

  nv overlay evpn
  feature ospf
  feature bgp
  feature pim
  feature interface-vlan
  feature vn-segment-vlan-based
  feature nv overlay

  • Enable VXLAN with distributed anycast-gateway using BGP EVPN

  fabric forwarding anycast-gateway-mac 0000.2222.3333

  • Configure PIM RP

  ip pim rp-address 100.1.1.1 group-list 224.0.0.0/4  
  ip pim ssm range 232.0.0.0/8
• Create VLANs
  vlan 1-1002

• Create overlay VRF VLAN and configure vn-segment
  vlan 101
  vn-segment 900001

• Create VLAN and provide mapping to VXLAN
  vlan 1001
  vn-segment 2001001
  vlan 1002
  vn-segment 2001002

• Create VRF and configure VNI
  vrf context vxlan-900001
  vni 900001
  rd auto
  address-family ipv4 unicast
  route-target both auto
  route-target both auto evpn
  address-family ipv6 unicast
  route-target both auto
  route-target both auto evpn

• Configure ACL TCAM region for ARP suppression

  Note  The `hardware access-list tcam region arp-ether 256 double-wide` command
  is not needed for Cisco Nexus 9300-EX, 9300-FX, and 9300-FX2 switches.

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

• Configure internal control VLAN/SVI for the VRF
  interface Vlan101
  no shutdown
  vrf member vxlan-900001
  ip forward

• 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 411:0:1::1/64
  fabric forwarding mode anycast-gateway
interface Vlan1002
  no shutdown
  vrf member vxlan-900001
  ip address 4.2.2.1/24
  ipv6 address 4:2:0:1::1/64
  fabric forwarding mode anycast-gateway

• Create the network virtualization endpoint (NVE) interface

interface nve1
  no shutdown
  source-interface loopback0
  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

• Configure interfaces for hosts/servers

interface Ethernet1/47
  switchport
  switchport access vlan 1002

interface Ethernet1/48
  switchport
  switchport access vlan 1001

• Configure interfaces for Spine-leaf interconnect

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

interface Ethernet2/3
  no switchport
  ip address 192.168.2.23/24
  ip router ospf 1 area 0.0.0.0
  ip pim sparse-mode
  no shutdown

• Configure Loopback for local VTEP IP, and BGP

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

• Enabling OSPF for underlay routing

    router ospf 1

• Configure BGP
router bgp 65535
router-id 40.1.1.1
neighbor 10.1.1.1 remote-as 65535
    update-source loopback0
    address-family l2vpn evpn
    send-community both
neighbor 20.1.1.1 remote-as 65535
    update-source loopback0
    address-family l2vpn evpn
    send-community both
vrf vxlan-900001
    address-family ipv4 unicast
    advertise l2vpn evpn
evpn
    vni 2001001 12
        rd auto
        route-target import auto
        route-target export auto
    vni 2001002 12
        rd auto
        route-target import auto
        route-target export auto

evpn
    vni 2001001 12
        rd auto
        route-target import auto
        route-target export auto
    vni 2001002 12
        rd auto
        route-target import auto
        route-target export auto

Example Show Commands

• show nve peers

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

• show nve vni

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

<table>
<thead>
<tr>
<th>Interface</th>
<th>VNI</th>
<th>Multicast-group</th>
<th>State</th>
<th>Mode</th>
<th>Type [BD/VRF]</th>
<th>Flags</th>
</tr>
</thead>
<tbody>
<tr>
<td>nve1</td>
<td>900001</td>
<td>n/a</td>
<td>Up</td>
<td>CP</td>
<td>L3 [vxlan-900001]</td>
<td></td>
</tr>
<tr>
<td>nve1</td>
<td>2001001</td>
<td>225.4.0.1</td>
<td>Up</td>
<td>CP</td>
<td>L2 [1001]</td>
<td>SA</td>
</tr>
<tr>
<td>nve1</td>
<td>2001002</td>
<td>225.4.0.1</td>
<td>Up</td>
<td>CP</td>
<td>L2 [1002]</td>
<td>SA</td>
</tr>
</tbody>
</table>
**show ip arp suppression-cache detail**

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

<table>
<thead>
<tr>
<th>Ip Address</th>
<th>Age</th>
<th>Mac Address</th>
<th>Vlan</th>
<th>Physical-ifindex</th>
<th>Flags</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1.1.54</td>
<td>00:06:41</td>
<td>00:54.0000.0000</td>
<td>1001</td>
<td>Ethernet1/48</td>
<td>L</td>
</tr>
<tr>
<td>4.1.1.51</td>
<td>00:20:33</td>
<td>00:51.0000.0000</td>
<td>1001</td>
<td>(null)</td>
<td>R</td>
</tr>
<tr>
<td>4.2.2.53</td>
<td>00:06:41</td>
<td>00:53.0000.0000</td>
<td>1002</td>
<td>Ethernet1/47</td>
<td>L</td>
</tr>
<tr>
<td>4.2.2.52</td>
<td>00:20:33</td>
<td>00:52.0000.0000</td>
<td>1002</td>
<td>(null)</td>
<td>R</td>
</tr>
</tbody>
</table>

**show vxlan interface**

```
Interface Vlan VPL I findex LTL HW VP
--------- ---- ----------- ---- ----
Eth1/47   1002 0x4c07d22e 0x10000 5697
Eth1/48   1001 0x4c07d02f 0x10001 5698
```

**show bgp l2vpn evpn summary**

```
BGP summary information for VRF default, address family L2VPN EVPN
BGP router identifier 40.1.1.1, local AS number 65535
BGP table version is 27, L2VPN EVPN config peers 2, capable peers 2
14 network entries and 18 paths using 2984 bytes of memory
BGP attribute entries [14/2240], BGP AS path entries [0/0]
BGP community entries [0/0], BGP clusterlist entries [2/8]
```

<table>
<thead>
<tr>
<th>Neighbor</th>
<th>V</th>
<th>AS</th>
<th>MsgRcvd</th>
<th>MsgSent</th>
<th>TblVer</th>
<th>InQ</th>
<th>OutQ</th>
<th>Up/Down</th>
<th>State/PfxRcd</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.1.1.1</td>
<td>4</td>
<td>65535</td>
<td>30199</td>
<td>30194</td>
<td>27</td>
<td>0</td>
<td>0</td>
<td>2w6d</td>
<td>4</td>
</tr>
<tr>
<td>20.1.1.1</td>
<td>4</td>
<td>65535</td>
<td>30199</td>
<td>30194</td>
<td>27</td>
<td>0</td>
<td>0</td>
<td>2w6d</td>
<td>4</td>
</tr>
</tbody>
</table>

**show bgp l2vpn evpn**

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

<table>
<thead>
<tr>
<th>Network</th>
<th>Route Distinguisher: 30.1.1.1:33768</th>
<th>Metric</th>
<th>LocPrf</th>
<th>Weight</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>*i&gt;1[2]:[0]:[0]:[48]:[d8b1.9071.e903]:[0]:[0.0.0.0]216</td>
<td>100</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>30.1.1.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>30.1.1.1</td>
<td>100</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10.1.1.1</td>
<td>100</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30.1.1.1</td>
<td>100</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30.1.1.1</td>
<td>100</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30.1.1.1</td>
<td>100</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Route Distinguisher: 30.1.1.1:33769
```

```
Configuring VXLAN BGP EVPN

Example Show Commands

• show l2route evpn mac all

9396-B# show l2route evpn mac all

<table>
<thead>
<tr>
<th>Topology</th>
<th>Mac Address</th>
<th>Prod</th>
<th>Flags</th>
<th>Seq No</th>
<th>Next-Hops</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>6412.2574.9f27 VXLAN</td>
<td>Rmac</td>
<td>0</td>
<td>30.1.1.1</td>
<td></td>
</tr>
<tr>
<td>1001</td>
<td>d8b1.9071.e903 BGP</td>
<td>SplRcv</td>
<td>0</td>
<td>30.1.1.1</td>
<td></td>
</tr>
<tr>
<td>1001</td>
<td>f8c2.8890.2a45 Local</td>
<td>L,</td>
<td>0</td>
<td>Eth1/48</td>
<td></td>
</tr>
<tr>
<td>1002</td>
<td>d8b1.9071.e903 BGP</td>
<td>SplRcv</td>
<td>0</td>
<td>30.1.1.1</td>
<td></td>
</tr>
<tr>
<td>1002</td>
<td>f8c2.8890.2a45 Local</td>
<td>L,</td>
<td>0</td>
<td>Eth1/47</td>
<td></td>
</tr>
</tbody>
</table>

• show l2route evpn mac-ip all

9396-B# show l2route evpn mac-ip all

<table>
<thead>
<tr>
<th>Topology</th>
<th>Mac Address</th>
<th>Prod</th>
<th>Flags</th>
<th>Seq No</th>
<th>Host IP</th>
<th>Next-Hops</th>
</tr>
</thead>
<tbody>
<tr>
<td>1001</td>
<td>d8b1.9071.e903 BGP</td>
<td>--</td>
<td>0</td>
<td>4.1.1.12</td>
<td>30.1.1.1</td>
<td></td>
</tr>
<tr>
<td>1001</td>
<td>f8c2.8890.2a45 BGP</td>
<td>--</td>
<td>0</td>
<td>4.1.1.122</td>
<td>Local</td>
<td></td>
</tr>
<tr>
<td>1002</td>
<td>d8b1.9071.e903 BGP</td>
<td>--</td>
<td>0</td>
<td>4.2.2.11</td>
<td>30.1.1.1</td>
<td></td>
</tr>
<tr>
<td>1002</td>
<td>f8c2.8890.2a45 BGP</td>
<td>--</td>
<td>0</td>
<td>4.2.2.111</td>
<td>Local</td>
<td></td>
</tr>
</tbody>
</table>
Example Show Commands
CHAPTER 5

Configuring VXLAN OAM

This chapter contains the following sections:

- VXLAN OAM Overview, on page 115
- Loopback (Ping) Message, on page 116
- Traceroute or Pathtrace Message, on page 117
- Configuring VXLAN OAM, on page 119
- Configuring NGOAM Profile, on page 122
- NGOAM Authentication, on page 123

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 7: VXLAN OAM Tools

<table>
<thead>
<tr>
<th>Category</th>
<th>Tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fault Verification</td>
<td>Loopback Message</td>
</tr>
</tbody>
</table>
### 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)

---

<table>
<thead>
<tr>
<th>Category</th>
<th>Tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fault Isolation</td>
<td>Path Trace Message</td>
</tr>
<tr>
<td>Performance</td>
<td>Delay Measurement, Loss Measurement</td>
</tr>
<tr>
<td>Auxiliary</td>
<td>Address Binding Verification, IP End Station Locator, Error Notification, OAM Command Messages, and Diagnostic Payload Discovery for ECMP Coverage</td>
</tr>
</tbody>
</table>
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.
intermediate device does not support the NVO3 draft Tissa channel, the pathtrace behaves as a simple traceroute and it provides only the hop information.

**Traceroute**

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

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

**Pathtrace**

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

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

**Figure 17: Traceroute Message**
## Configuring VXLAN OAM

### Before you begin
As a prerequisite, ensure that the VXLAN configuration is complete.

### Procedure

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

### Example
See the following examples of the configuration topology.
VXLAN OAM provides the visibility of the host at the switch level, that allows a leaf to ping the host using the \texttt{ping} \texttt{nve} command.

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

\begin{verbatim}
switch# ping nve ip 209.165.201.5 vrf vni-31000 source 1.1.1.1 verbose
\end{verbatim}

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

\begin{quote}
\textbf{Note}
\end{quote}
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 VM2 via Spine 1.

\begin{verbatim}
switch# traceroute nve ip 209.165.201.5 vrf vni-31000 source 1.1.1.1 verbose
\end{verbatim}
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
================================================================================
Configuring NGOAM Profile

Complete the following steps to configure NGOAM profile.

**Procedure**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><code>switch(config)#[no] feature ngoam</code></td>
<td>Enables or disables NGOAM feature</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><code>switch(config)#[no] ngoam profile &lt;profile-id&gt;</code></td>
<td>Configures OAM profile. The range for the profile-id is &lt;1 – 1023&gt;. This command does not have a default value. Enters the config-ngoam-profile submode to configure NGOAM specific commands.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><code>switch(config-ng-oam-profile)# ?</code></td>
<td>Displays the options for configuring NGOAM profile.</td>
</tr>
</tbody>
</table>

**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
```

**Example**

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

```
switch(config)#
```
NGOAM Authentication

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

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

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

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

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

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

```
switch# pathtrace nve ip 12.0.22.1 profile 1 vni 31000 req-stats ver
Path trace Request to peer ip 12.0.22.1 source ip 11.0.22.1
Hop Code ReplyIP IngressI/f EgressI/f State
====================================================
1 !Reply from 55.55.55.2, Eth5/7/1 Eth5/7/2 UP / UP
  Input Stats: PktRate:0 ByteRate:0 Load:0 Bytes:339573434 unicast:14657 mcast:307581
  bcast:67 discards:0 errors:3 unknown:0 bandwidth:42949672970000000
  Output Stats: PktRate:0 ByteRate:0 load:0 bytes:237399176 unicast:2929 mcast:535710
  bcast:10408 discards:0 errors:0 bandwidth:42949672970000000
  2 !Reply from 12.0.22.1, Eth1/7 Unknown UP / DOWN
```
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:4294967297000000
   Output Stats: PktRate:0 ByteRate:0 load:0 bytes:237405790 unicast:2929 mcast:535716
   bcast:10408 discards:0 errors:0 bandwidth:4294967297000000
2 !Reply from 12.0.22.1, Eth1/17 Unknown UP / DOWN
```
CHAPTER 6

Configuring VXLAN EVPN Multihoming

This chapter contains the following sections:

• VXLAN EVPN Multihoming Overview, on page 125
• Configuring VXLAN EVPN Multihoming, on page 129
• Configuring Layer 2 Gateway STP, on page 131
• Configuring VXLAN EVPN Multihoming Traffic Flows, on page 135
• Configuring VLAN Consistency Checking, on page 147
• Configuring ESI ARP Suppression, on page 150

VXLAN EVPN Multihoming Overview

Introduction to Multihoming

Cisco Nexus platforms support vPC-based multihoming, where a pair of switches act as a single device for redundancy and both switches function in an active mode. With Cisco Nexus 9000 Series switches in VXLAN BGP EVPN environment, there are two solutions to support Layer 2 multihoming; the solutions are based on the Traditional vPC (emulated or virtual IP address) and the BGP EVPN techniques.

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

BGP EVPN Multihoming

When using BGP EVPN control plane, each switch can use its own local IP address as the VTEP IP address and it still provides an active/active redundancy. BGP EVPN based multihoming further provides fast convergence during certain failure scenarios, that otherwise cannot be achieved without a control protocol (data plane flood and learn).

BGP EVPN Multihoming Terminology

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

### EVPN Multihoming Implementation

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

There are 5 different route types currently defined:

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

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

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

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

To summarize the new implementation concepts for the EVPN multihoming:

• EAD/ES: Ethernet Auto Discovery Route per ES that is also referred to as type-1 route. This route is used to converge the traffic faster during access failure scenarios. This route has Ethernet Tag of 0xFFFFFFFF.
• EAD/EVI: Ethernet Auto Discovery Route per EVI that is also referred to as type-1 route. This route is used for aliasing and load balancing when the traffic only hashes to one of the switches. This route cannot have Ethernet Tag value of 0xffffff to differentiate it from the EAD/ES route.

• ES: Ethernet Segment route that is also referred to as type-4 route. This route is used for DF election for BUM traffic.

• Aliasing: It is used for load balancing the traffic to all the connected switches for a given Ethernet Segment using the type-1 EAD/EVI route. This is done irrespective of the switch where the hosts are actually learned.

• Mass Withdrawal: It is used for fast convergence during the access failure scenarios using the type-1 EAD/ES route.

• DF Election: It is used to prevent forwarding of the loops and the duplicates as only a single switch is allowed to decap and forward the traffic for a given Ethernet Segment.

• Split Horizon: It is used to prevent forwarding of the loops and the duplicates for the BUM traffic. Only the BUM traffic that originates from a remote site is allowed to be forwarded to a local site.

**EVPN Multihoming Redundancy Group**

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

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

Note that there is no Peer Link between L1 and L2 switches and each switch can have similar multiple bundle links that are shared with the same set of neighbors.

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

**Ethernet Segment Identifier**

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

**LACP Bundling**

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

The access switch receives the same configured MAC value from both switches (L1 and L2). Therefore, it puts the bundled link in the UP state. Since the ES MAC can be shared across all the Ethernet-segments on the switch, LACP PDUs use ES MAC as system MAC address and the admin_key carries the ES ID.
Cisco recommends running LACP between the switches and the access devices since LACP PDUs have a mechanism to detect and act on the misconfigured ES IDs. In case there is mismatch on the configured ES ID under the same PO, LACP brings down one of the links (first link that comes online stays up). By default, on most Cisco Nexus platforms, LACP sets a port to the suspended state if it does not receive an LACP PDU from the peer. This is based on the `lacp suspend-individual` command that is enabled by default. This command helps in preventing loops that are created due to the ESI configuration mismatch. Therefore, it is recommended to enable this command on the port-channels on the access switches and the servers.

In some scenarios (for example, POAP or NetBoot), it can cause the servers to fail to boot up because they require LACP to logically bring up the port. In case you are using static port channel and you have mismatched ES IDs, the MAC address gets learned from both L1 and L2 switches. Therefore, both the switches advertise the same MAC address belonging to different ES IDs that triggers the MAC address move scenario. Eventually, no traffic is forwarded to that node for the MAC addresses that are learned on both L1 and L2 switches.

### Guidelines and Limitations for VXLAN EVPN Multihoming

See the following limitations for configuring VXLAN EVPN Multihoming:

- VXLAN EVPN Multihoming works with the iBGP or eBGP control plane. iBGP is preferred.

- If iBGP is used with VXLAN EVPN Multihoming, the administrative distance for local learned endpoints value must be lower than the value of iBGP.

  | Note | The default value for local learned endpoints is 190, the default value for eBGP is 20, and the default value for iBGP is 200. |

- If eBGP is used with VXLAN EVPN Multihoming, the administrative distance for local learned endpoints must be lower than the value of eBGP. The administrative distance can be changed by entering the `fabric forwarding admin-distance distance` command.

  | Note | The default value for local learned endpoints is 190, the default value for eBGP is 20, and the default value for iBGP is 200. |

- VXLAN Multihoming is supported on the Cisco Nexus 9300 platform switches only and it is not supported on the Cisco Nexus 9200, 9300-EX/-FX/-FXP/-FX2 and 9500 platform switches. The Cisco Nexus 9500 platform switches can be used as Spine switches, but they cannot be used as VTEPs.

- VXLAN Multihoming requires that all switches in a given network must be VXLAN Multihoming capable. Mixing platforms with and without VXLAN Multihoming is not supported.

- VXLAN multihoming is not supported on FEX.

- Beginning with Cisco NX-OS Release 7.0(3)I5(2), ARP suppression is supported with VXLAN multihoming.

- VXLAN Multihoming is supported with multihoming to two switches only.

- To enable VXLAN Multihoming, the spine switches must be running the minimum software version as Cisco NX-OS Release 7.0(3)I5(2) or later.

- Switchport trunk native VLAN is not supported on the trunk interfaces.
• Cisco recommends enabling LACP on ES PO.
• IPv6 is not currently supported.
• ISSU is not supported if ESI is configured on the Cisco Nexus 9300 Series switches.

Configuring VXLAN EVPN Multihoming

Enabling EVPN Multihoming

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

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

Complete the following steps to configure EVPN multihoming:

**Before you begin**

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

**Procedure**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>evpn esi multihoming</code></td>
<td>Enables EVPN multihoming globally.</td>
</tr>
</tbody>
</table>
| Step 2 | `address-family l2vpn evpn maximum-paths <> maximum-paths ibgp <>` | Enables BGP maximum-path to enable ECMP for the MAC routes. Otherwise, the MAC routes have only 1 VTEP as the next-hop. This configuration is needed under BGP in Global level. **Example:**
  
  ```
  address-family l2vpn evpn
  maximum-paths 64
  maximum-paths ibgp 64
  ```
| Step 3 | `evpn multihoming core-tracking` | Enables EVPN multihoming core-links. It tracks the uplink interfaces towards the core. If all uplinks are down, the local ES based the POs is shut down/suspended. This is mainly used to avoid black-holing South-to-North traffic when no uplinks are available. |
| Step 4 | `interface port-channel Ethernet-segment <> System-mac <>` | Configures the local Ethernet Segment ID. The ES ID has to match on VTEPs where the PO is multihomed. The Ethernet Segment ID should be unique per PO. **Example:**

---

Cisco Nexus 9000 Series NX-OS VXLAN Configuration Guide, Release 7.x
### VXLAN EVPN Multihoming Configuration Examples

See the sample VXLAN EVPN multihoming configuration on the switches:

#### Switch 1 (L1)
```plaintext
evpn esi multihoming

router bgp 1001
    address-family l2vpn evpn

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

interface Ethernet2/2
    no switchport
    evpn multihoming core-tracking
    mtu 9216
    ip address 10.1.1.5/30
    ip pim sparse-mode
    no shutdown

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

#### Switch 2 (L2)
```plaintext
evpn esi multihoming

router bgp 1001
    address-family l2vpn evpn

```

---

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>ethernet-segment 11 system-mac 0000.0000.0011</td>
<td>Configures the local system-mac ID that has to match on the VTEPs where the PO is multihomed. The system-mac address can be shared across multiple POs.</td>
</tr>
<tr>
<td>Step 5 hardware access-list tcam region vpc-convergence 256</td>
<td>Configures the TCAM. This command is used to configure the split horizon ACLs in the hardware. This command avoids BUM traffic duplication on the shared ES POs.</td>
</tr>
</tbody>
</table>
Configuring Layer 2 Gateway STP

Layer 2 Gateway STP Overview

Beginning with Cisco NX-OS Release 7.0(3)I5(2), EVVPN multihoming is supported with the Layer 2 Gateway Spanning Tree Protocol (L2G-STP). The Layer 2 Gateway Spanning Tree Protocol (L2G-STP) builds a loop-free tree topology. However, the Spanning Tree Protocol root must always be in the VXLAN fabric. A bridge ID for the Spanning Tree Protocol consists of a MAC address and the bridge priority. When the system is running in the VXLAN fabric, the system automatically assigns the VTEPs with the MAC address c84c.75fa.6000 from a pool of reserved MAC addresses. As a result, each switch uses the same MAC address for the bridge ID emulating a single logical pseudo root.

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

All the access ports from VTEPs connecting to the customer access switches are in a `desg` forwarding state by default. All ports on the customer access switches connecting to VTEPs are either in root-port forwarding or alt-port blocking state. The root-guard kicks in if better or superior STP information is received from the customer access switches and it puts the ports in the `blk l2g_inc` state to secure the root on the overlay-fabric and to prevent a loop.
Guidelines for Moving to Layer 2 Gateway STP

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

- With Layer 2 Gateway STP, root guard is enabled by default on all the access ports.

- With Layer 2 Gateway STP enabled, the VXLAN fabric (all VTEPs) emulates a single pseudo-root switch for the customer access switches.

- All access ports from VTEPs connecting to the customer access switches are in the Desg FWD state by default.

- All ports on customer access switches connecting to VTEPs are either in the root-port FWD or Altn BLK state.

- Root guard is activated if superior spanning-tree information is received from the customer access switches. This process puts the ports in BLK L2GW_Inc state to secure the root on the VXLAN fabric and prevent a loop.

- Explicit domain ID configuration is needed to enable spanning-tree BPDU tunneling across the fabric.

- As a best practice, you should configure all VTEPs with the lowest spanning-tree priority of all switches in the spanning-tree domain to which they are attached. By setting all the VTEPs as the root bridge, the entire VXLAN fabric appears to be one virtual bridge.

- ESI interfaces should not be enabled in spanning-tree edge mode to allow Layer 2 Gateway STP to run across the VTEP and access layer.

- You can continue to use ESIs or orphans (single-homed hosts) in spanning-tree edge mode if they directly connect to hosts or servers that do not run Spanning Tree Protocol and are end hosts.

- Configure all VTEPs that are connected by a common customer access layer in the same Layer 2 Gateway STP domain. Ideally, all VTEPs on the fabric on which the hosts reside and to which the hosts can move.

- The Layer 2 Gateway STP domain scope is global, and all ESIs on a given VTEP can participate in only one domain.

- Mappings between Multiple Spanning Tree (MST) instances and VLANs must be consistent across the VTEPs in a given Layer 2 Gateway STP domain.

- Non-Layer 2 Gateway STP enabled VTEPs cannot be directly connected to Layer 2 Gateway STP-enabled VTEPs. Performing this action results in conflicts and disputes because the non-Layer 2 Gateway STP VTEP keeps sending BPDUs and it can steer the root outside.

- Keep the current edge and the BPDU filter configurations on both the Cisco Nexus switches and the access switches after upgrading to the latest build.

- Enable Layer 2 Gateway STP on all the switches with a recommended priority and the mst instance mapping as needed. Use the commands spanning-tree domain enable and spanning-tree mst <instance-id's> priority 8192.

- Remove the BPDU filter configurations on the switch side first.

- Remove the BPDU filter configurations and the edge on the customer access switch.

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

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

Procedure

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

Example

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

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

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

switch# show spanning-tree vlan 1001
MST0000
Spanning tree enabled protocol mstp

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

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

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

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

switch# show spanning-tree

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

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

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

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

switch(config)# spanning-tree domain disable

switch# show spanning-tree summary
Switch is in mst mode (IEEE Standard)
With Layer 2 Gateway STP, the access ports on VTEPs cannot be in an edge port, because they behave like normal spanning-tree ports, receiving BPDU’s from the access switches. In that case, the access ports on VTEPs lose the advantage of rapid transmission, instead forwarding on Ethernet segment link flap. (They have to go through a proposal and agreement handshake before assuming the FWD-Def role).

Configuring VXLAN EVPN Multihoming Traffic Flows

EVPN Multihoming Local Traffic Flows

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

Locally Bridged Traffic

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

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

Note

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

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

Locally Routed Traffic

Consider H1, H2, and H3 being in different subnets and L1/L2 being distributed anycast gateways.
Any packet that is routed from H1 to H2 is directly sent from L1 via native routing. However, host H3 is not a locally attached adjacency, unlike in vPC case where the ARP entry syncs to L1 as a locally attached adjacency. Instead, H3 shows up as a remote host in the IP table at L1, installed in the context of L3 VNI. This packet must be encapsulated in the router-MAC of L2 and routed to L2 via VXLAN overlay.

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

Figure 22: L1 is Distributed Anycast Gateway. H1, H2, and H3 are in different VLANs. H1->H3 routing happens via VXLAN tunnel encapsulation. In vPC, H3 ARP would have been synced via Peer Link and direct routing.

Access Failure for Locally Routed Traffic

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

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

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

Figure 24: H1, H2, and H3 are in different VLANs. Core fails on L1. Access is brought down. H1 loses all connectivity.
EVPN Multihoming Remote Traffic Flows

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

![Layer 2 VXLAN Gateway. L3 performs MAC ECMP to L1/L2.](image)

Remote Bridged Traffic

Consider a remote host H5 that wants to bridge traffic to host H2 that is positioned behind the EVPN MH Complex (L1, L2). Host H2 builds an ECMP list in accordance to the rules defined in RFC 7432. The MAC table at switch L3 displays that the MAC entry for H2 points to an ECMP PathList comprising of IP-L1 and IP-L2. Any bridged traffic going from H5 to H2 is VXLAN encapsulated and load balanced to switches L1 and L2. When making the ECMP list, the following constructs need to be kept in mind:

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

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

**Ethernet Auto Discovery Route (Type 1) per ES**

EVPN defines a mechanism to efficiently and quickly signal the need to update their forwarding tables upon the occurrence of a failure in connectivity to an Ethernet Segment. Having each PE advertise a set of one or more Ethernet A-D per ES route for each locally attached Ethernet Segment does this.
**Ethernet Auto Discovery Route (Route Type 1) per ES**

<table>
<thead>
<tr>
<th>NLRI</th>
<th>Route Type</th>
<th>Ethernet Segment (Type 1)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Route Distinguisher</td>
<td>Router-ID: Segment-ID (VNID &lt;&lt; 8)</td>
</tr>
<tr>
<td>ESI</td>
<td>&lt;Type: 1B&gt;&lt;MAC: 6B&gt;&lt;LD: 3B&gt;</td>
<td></td>
</tr>
<tr>
<td>Ethernet Tag</td>
<td>MAX-ET</td>
<td></td>
</tr>
<tr>
<td>MPLS Label</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

**ATTRS**

<table>
<thead>
<tr>
<th>ESI Label Extended Community</th>
<th>Single Active = False</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESI Label = 0</td>
<td></td>
</tr>
<tr>
<td>Next-Hop</td>
<td>NVE Loopback IP</td>
</tr>
<tr>
<td>Route Target</td>
<td>Subset of List of RTs of MAC-VRFs associated to all the EVIs active on the ES</td>
</tr>
</tbody>
</table>

**MAC-IP Route (Type 2)**

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

**MAC IP Route (Route Type 2)**

<table>
<thead>
<tr>
<th>NLRI</th>
<th>Route Type</th>
<th>MAC IP Route (Type 2)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Route Distinguisher</td>
<td>RD of MAC-VRF associated to the Host</td>
</tr>
<tr>
<td>ESI</td>
<td>&lt;Type : 1B&gt;&lt;MAC : 6B&gt;&lt;LD : 3B&gt;</td>
<td></td>
</tr>
<tr>
<td>Ethernet Tag</td>
<td>MAX-ET</td>
<td></td>
</tr>
<tr>
<td>MAC Addr</td>
<td>MAC Address of the Host</td>
<td></td>
</tr>
<tr>
<td>IP Addr</td>
<td>IP Address of the Host</td>
<td></td>
</tr>
<tr>
<td>Labels</td>
<td>L2VNI associated to the MAC-VRF</td>
<td></td>
</tr>
<tr>
<td></td>
<td>L3VNI associated to the L3-VRF</td>
<td></td>
</tr>
</tbody>
</table>

**ATTRS**

<table>
<thead>
<tr>
<th>Next-Hop</th>
<th>Loopback of NVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>RT Export</td>
<td>RT configured under MAC-VRF (AND/OR) L3-VRF associated to the host</td>
</tr>
</tbody>
</table>
Access Failure for Remote Bridged Traffic

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

Figure 26: Layer 2 VXLAN Gateway. ESI failure on L1. L3 withdraws L1 from MAC ECMP list. This will happen due to EAD/ES mass withdrawal from L1.

Core Failure for Remote Bridged Traffic

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

Figure 27: Layer 2 VXLAN Gateway. Core failure at L1. L3 withdraws L1 from MAC ECMP list. This will happen due to route reachability to L1 going away at L3.

Remote Routed Traffic

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

**Figure 28: Layer 3 VXLAN Gateway. L3 does IP ECMP to L1/L2 for inter subnet traffic.**

**Access Failure for Remote Routed Traffic**

If the access link pointing to ES1 goes down on L1, the mass withdrawal route is sent in the form of EAD/ES and that causes L3 to remove L1 from the MAC ECMP PathList, leading the intra-subnet (L2) traffic to converge quickly. L1 now treats H2 as a remote route reachable via VxLAN Overlay as it is no longer directly connected through the ESI link. This causes the traffic destined to H2 to take the suboptimal path L3->L1->L2. Inter-Subnet traffic H5->H2 will follow the following path:

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

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

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

Figure 30: Layer 3 VXLAN Gateway. Core failure. All L3 ECMP paths to L1 are withdrawn at L3 due to route reachability going away.

EVPN Multihoming BUM Flows

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

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

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

Split Horizon and Local Bias

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

Remote switches decap and forward it to their ESI and orphan links based on the DF state. However, this packet is also received at L2 that belongs to the same redundancy group as the originating switch L1. L2 must decap the packet to send it to orphan ports. However, even through L2 is the DF for ES1, L2 must not forward this packet to ES1 link. This packet was received from a peer that shares ES1 with L1 as L1 would have done local-bias and duplicate copies should not be received on ES2. Therefore L2 (DF) applies a split-horizon filter for L1-IP on ES1 and ES2 that it shares with L1. This filter is applied in the context of a VLAN.
Figure 32: BUM traffic originating at L1. L2 is the DF for ES1 and ES2. However, L2 must perform split horizon check here as it shares ES1 and ES2 with L1. L2 however

Ethernet Segment Route (Type 4)

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

<table>
<thead>
<tr>
<th>Ethernet Segment Route (Route Type 4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NLRI</td>
</tr>
<tr>
<td>RD</td>
</tr>
<tr>
<td>ESI</td>
</tr>
<tr>
<td>Originator IP</td>
</tr>
<tr>
<td>ATTRS</td>
</tr>
</tbody>
</table>

DF Election and VLAN Carving

Upon configuration of the ESI, both L1 and L2 advertises the ES route. The ESI MAC is common between L1 and L2 and unique in the network. Therefore, only L1 and L2 import each other’s ES routes.
IF VLAN % COUNT EQUALS TO ORDINAL, TAKE UP DF ROLE.

Core and Site Failures for BUM Traffic

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

BGP EVPN multihoming on Cisco Nexus 9000 Series switches provides minimum operational and cabling expenditure, provisioning simplicity, flow based load balancing, multi pathing, and fail-safe redundancy.

Configuring VLAN Consistency Checking

Overview of VLAN Consistency Checking

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

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

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

When a trigger (for example, device booting up, VLAN configuration change, VLANs administrative state change on the ethernet-segment port-channel) is issued on one of the multihoming peers, a broadcast request...
with a snapshot of configured and administratively up VLANs for the ethernet-segment (ES) is sent to all the CFS peers.

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

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

VLAN consistency checking runs for the following events:

- Global VLAN configuration: Add, delete, shut, or no shut events.
- Port channel VLAN configuration: Trunk allowed VLANs added or removed or access VLAN changed.
- CFS events: CFS peer added or deleted or CFSoIP configuration is removed.
- ES Peer Events: ES peer added or deleted.

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

**VLAN Consistency Checking Guidelines and Limitations**

See the following guidelines and limitations for VLAN consistency checking:

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

**Configuring VLAN Consistency Checking**

Use the `cfs ipv4 mcast-address <mcast address>` CLI command to override the default CFS multicast address. Use the `cfs ipv4 distribute` CLI command to enable CFSoIP.

To enable or disable the VLAN consistency checking, use the new `vlan-consistency-check` CLI command that has been added under the `evpn esi multihoming` mode.

```
switch (config)# sh running-config | in cfs
 cfs ipv4 mcast-address 239.255.200.200
 cfs ipv4 distribute
```
Displaying Show command Output for VLAN Consistency Checking

See the following show commands output for VLAN consistency checking.

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

```
switch# sh cfs peers name nve
Scope : Physical-ip
---
Switch WWN IP Address
---
20:00:f8:c2:88:90:c6:21 172.31.201.172 [Not Merged]
20:00:f8:c2:88:23:22:8f 172.31.203.38 [Not Merged]
20:00:f8:c2:88:23:1d:e1 172.31.150.132 [Not Merged]
20:00:f8:c2:88:23:05:1d 172.31.150.134 [Not Merged]
```

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

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

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

See the following Syslog output:
Configuring ESI ARP Suppression

Overview of ESI ARP Suppression

ESI ARP suppression is an extension of already available ARP suppression solution in VXLAN-EVPN. This feature is supported on top of ESI multihoming solution, that is on top of VXLAN-EVPN solution. ARP suppression is an optimization on top of BGP-EVPN multihoming solution. ARP broadcast is one of the most significant part of broadcast traffic in data centers. ARP suppression significantly cuts down on ARP broadcast in the data center.

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

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

ESI ARP suppression is a per-VNI (L2-VNI) feature. ESI ARP suppression is supported in both L2 (no SVI) and L3 modes. Beginning with Cisco NX-OS Release 7.0(3)I5(2), only L3 mode is supported.

The ESI ARP suppression cache is built by:

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

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

Limitations for ESI ARP Suppression

See the following limitations for ESI ARP suppression:

- ESI multihoming solution is supported only on Cisco Nexus 9300 Series switches at the leafs.
- ESI ARP suppression is only supported in L3 [SVI] mode.
- ESI ARP suppression cache limit is 64K that includes both local and remote entries.
Configuring ESI ARP Suppression

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

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

Displaying Show Commands for ESI ARP Suppression

See the following Show commands output for ESI ARP suppression:

```
switch# show ip arp suppression-cache ?
detail   Show details
local    Show local entries
remote   Show remote entries
statistics Show statistics
summary  Show summary
vlan     L2vlan

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

Ip Address Age  Mac Address  Vlan  Physical-ifindex  Flags Remote Vtep Addrs
61.1.1.20     00:07:54 0000.0610.0020 610   port-channel20 L
61.1.1.30     00:07:54 0000.0610.0030 610   port-channel2 L[PS RO]
61.1.1.10     00:07:54 0000.0610.0010 610   Ethernet1/96 L

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

Ip Address Age  Mac Address  Vlan  Physical-ifindex  Flags Remote Vtep Addrs
61.1.1.40     00:48:37 0000.0610.0040 610 (null)   R VTEP1, VTEP2,.. VTEPn

switch# show ip arp suppression-cache detail
Flags: + - Adjacencies synced via CFSoE
L - Local Adjacency
R - Remote Adjacency
L2 - Learnt over L2 interface
```
### Displaying Show Commands for ESI ARP Suppression

**Configuring VXLAN EVPN Multihoming**

**Displaying Show Commands for ESI ARP Suppression**

- **PS** - Added via L2RIB, Peer Sync
- **RO** - Derived from L2RIB Peer Sync Entry

<table>
<thead>
<tr>
<th>Ip Address</th>
<th>Age</th>
<th>Mac Address</th>
<th>Vlan</th>
<th>Physical-ifindex</th>
<th>Flags</th>
</tr>
</thead>
<tbody>
<tr>
<td>61.1.1.20</td>
<td>00:00:07</td>
<td>0000.0610.0020</td>
<td>610</td>
<td>port-channel20</td>
<td>L</td>
</tr>
<tr>
<td>61.1.1.30</td>
<td>00:00:07</td>
<td>0000.0610.0030</td>
<td>610</td>
<td>port-channel2</td>
<td>L[PS RO]</td>
</tr>
<tr>
<td>61.1.1.10</td>
<td>00:00:07</td>
<td>0000.0610.0010</td>
<td>610</td>
<td>Ethernet1/96</td>
<td>L</td>
</tr>
<tr>
<td>61.1.1.40</td>
<td>00:00:07</td>
<td>0000.0610.0040</td>
<td>610</td>
<td>(null)</td>
<td>R</td>
</tr>
</tbody>
</table>

**VTEP1, VTEP2.. VTEPn**

```
switch# show ip arp suppression-cache summary
IP ARP suppression-cache Summary
Remote :1
Local :3
Total :4
switch# show ip arp suppression-cache statistics
ARP packet statistics for suppression-cache
Suppressed:
Total 0, Requests 0, Requests on L2 0, Gratuitous 0, Gratuitous on L2 0
Forwarded :
Total: 364
L3 mode : Requests 364, Replies 0
Request on core port 364, Reply on core port 0
Dropped 0
L2 mode : Requests 0, Replies 0
Request on core port 0, Reply on core port 0
Dropped 0

Received:
Total: 3016
L3 mode: Requests 376, Replies 2640
Local Request 12, Local Responses 2640
Gratuitous 0, Dropped 0
L2 mode : Requests 0, Replies 0
Gratuitous 0, Dropped 0

switch# sh ip arp multihoming-statistics vrf all
ARP Multihoming statistics for all contexts
Route Stats

```

----------
Received ADD from L2RIB :1756 | 1756:Processed ADD from L2RIB Received DEL from L2RIB :88 | 87:Processed DEL from L2RIB Received PC shut from L2RIB :0 | 1755:Processed PC shut from L2RIB Received remote UPD from L2RIB :5004 | 0:Processed remote UPD from L2RIB

----------
Multihoming ADD error invalid flag :0
Multihoming DEL error invalid flag :0
Multihoming ADD error invalid current state:0
Multihoming DEL error invalid current state:0
Peer sync DEL error MAC mismatch :0
Peer sync DEL error second delete :0
Peer sync DEL error deleting TL route :0
True local DEL error deleting PS RO route :0

switch#
Configuring VIP/PIP

This chapter contains the following sections:

- Advertising Primary IP Address, on page 153
- BorderPE Switches in a vPC Setup, on page 154
- DHCP Configuration in a vPC Setup, on page 154
- IP Prefix Advertisement in vPC Setup, on page 154

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`.

Note

On the Cisco Nexus 9300-FX2 switch, the `advertise-pip` command was not supported prior to Cisco NX-OS Release 7.0(3)I7(4). For more information, see CSCvi42831.

The following is a sample configuration:

```bash
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 prefix routes or leaf-generated 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.

For Cisco Nexus 9504 and 9508 switches with -R line cards, always configure `advertise virtual-rmac` without `advertise-pip`.

---

**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 12vpn 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.
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 is always 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:

```bash
switch(config)# router bgp 100
  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 prefix routes or leaf generated routes if vPC is enabled.
CHAPTER 8

Configuring VXLAN EVPN Multi-Site

This chapter contains the following sections:

- About VXLAN EVPN Multi-Site, on page 157
- Licensing Requirements for VXLAN EVPN Multi-Site, on page 158
- Guidelines and Limitations for VXLAN EVPN Multi-Site, on page 158
- Enabling VXLAN EVPN Multi-Site, on page 159
- Configuring VNI Dual Mode, on page 160
- Configuring Fabric/DCI Link Tracking, on page 162
- Configuring Fabric External Neighbors, on page 162

About VXLAN EVPN Multi-Site

The VXLAN EVPN Multi-Site solution uses border gateways is either anycast or virtual port channel configuration in the data plane to terminate and interconnect overly domains.

The border gateways provide the network control boundary that is necessary for traffic enforcement and failure containment functionality.

In the control plane, BGP sessions between the border gateways rewrite the next hop information of EVPN routes and re-originates them. VXLAN Tunnel Endpoints (VTEPs) are only aware of their overlay domain internal neighbors including the border gateways. All routes external to the fabric have a next hop on the border gateways for Layer 2 and Layer 3 traffic.

The VXLAN EVPN Multi-Site feature is a solution to interconnect two or more BGP-based Ethernet VPN (EVPN) site's fabrics in a scalable fashion over an IP-only network.

The Border Gateway (BG) 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. Border Gateways on one hand are also part of site-specific EVPN domain and on the other hand a part of a common EVPN domain to interconnect with Border Gateways from other sites. For a given site, these Border Gateways facilitate site-specific nodes to visualize all other sites to be reachable only via them. This would mean:
• Site-local bridging domains are interconnected only via Border Gateways with bridging domains from other sites.
• Site-local routing domains are interconnected only via Border Gateways with routing domains from other sites.
• Site-local flood domains are interconnected only via Border Gateways with flood domains from other sites.

Selective Advertisement is defined as the configuration of the per-tenant information on the border gateway. Specifically, this means IP-VRF or MAC-VRF (EVPN Instance). In cases where External Connectivity (VRF-lite) and EVPN Multi-Site co-exist on the same border gateway, the advertisements are always enabled.

**Licensing Requirements for VXLAN EVPN Multi-Site**

The following table shows the licensing requirements for VXLAN EVPN Multi-Site:

<table>
<thead>
<tr>
<th>Product</th>
<th>License Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco NX-OS</td>
<td>• For all border gateways, the N93-FAB1K9 or N95-FAB1K9 license is required.</td>
</tr>
<tr>
<td></td>
<td>• If the border gateways are N9300-EX or N9300-FX leafs, the N93-FAB1K9 license is required.</td>
</tr>
<tr>
<td></td>
<td>• If the border gateways are N9500 modular switches with -EX or -FX line cards, the N95-FAB1K9 license is required.</td>
</tr>
</tbody>
</table>

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

**Guidelines and Limitations for VXLAN EVPN Multi-Site**

VXLAN EVPN Multi-Site has the following configuration guidelines and limitations:

• Beginning with Cisco NX-OS Release 7.0(3)I7(3), support for VXLAN EVPN Multi-Site functionality on the Cisco Nexus N9K-C9336C-FX and N9K-C93240YC-FX2 is added. N9K-C9348GC-FXP does not support VXLAN EVPN Multi-Site functionality.

• Beginning with Cisco NX-OS Release 7.0(3)I7(2), VXLAN EVPN Multi-Site and Tenant Routed Multicast (TRM) is supported between source and receivers deployed in the same site.

• Beginning with Cisco NX-OS Release 7.0(3)I7(2), the Multi-Site border gateway allows the co-existence of Multi-Site extensions (Layer 2 unicast/multicast and Layer 3 unicast) as well as Layer 3 unicast and multicast external connectivity.

• The following switches support VXLAN EVPN Multi-Site:
  • Cisco Nexus 9300-EX, 9300-FX, and 9500 platform switches with X9700-EX line cards, beginning with Cisco NX-OS Release 7.0(3)I7(1)
The Cisco Nexus 9348GC-FXP switch does not support VXLAN EVPN Multi-Site functionality.

- Cisco Nexus 9396C switch and Cisco Nexus 9500 platform switches with X9700-FX line cards, beginning with Cisco Nexus NX-OS Release 7.0(3)I7(2)
- Cisco Nexus 9336C-FX2 switch, beginning with Cisco Nexus NX-OS Release 7.0(3)I7(3)

- The number of border gateways per site is limited to four.
- Border Gateways (BGWs) in a vPC topology are not supported.
- Support for Multicast Flood Domain between inter-site/fabric border gateways is not supported.
- Multicast Underlay between sites is not supported.
- iBGP EVPN Peering between border gateways of different fabrics/sites is not supported.
- Configure the peer-type fabric-external command only in Multi-Site deployments. We do not recommend configuring this command on a pseudo border gateway.

Enabling VXLAN EVPN Multi-Site

This procedure enables the VXLAN EVPN Multi-Site feature. Multi-Site is enabled on the border gateways only. The site-id must be the same on all border gateways in the fabric/site.

Procedure

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>switch# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td>evpn multisite border-gateway ms-id</td>
<td>Configure 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 border gateways within the same fabric/site.</td>
</tr>
<tr>
<td>Example:</td>
<td>switch(config)# evpn multisite border-gateway 100</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>interface nve 1</td>
<td>Creates a VXLAN overlay interface that terminates VXLAN tunnels.</td>
</tr>
<tr>
<td>Example:</td>
<td>switch(config-evpn-msite-bgw)# interface nve 1</td>
<td>Only 1 NVE interface is allowed on the switch</td>
</tr>
<tr>
<td>Step 4</td>
<td>source-interface loopback src-if</td>
<td>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</td>
</tr>
</tbody>
</table>
### Configuring VNI Dual Mode

This procedure describes the configuration of 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.
In cases where only a Layer 3 extension is configured on the BGW, an additional loopback interface is required. The loopback interface must be present in the same VRF instance on all BGWs and with an individual IP address per BGW. Ensure the loopback interfaces IP address is redistributed into BGP EVPN, specially towards Site-External. For more information about configuring the mcast-group (or ingress-replication protocol bgp) for a large number of VNIs, see Example of VXLAN BGP EVPN (EBGP), on page 94.

**Procedure**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: switch# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> interface nve 1</td>
<td>Creates a VXLAN overlay interface that terminates VXLAN tunnels.</td>
</tr>
<tr>
<td>Example: switch(config)# interface nve 1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> member vni vni-range</td>
<td>Configure 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.</td>
</tr>
<tr>
<td>Example: switch(config-if-nve)# member vni 200</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> mcast-group ip-addr</td>
<td>Configure the NVE Multicast group IP prefix within the fabric.</td>
</tr>
<tr>
<td>Example: switch(config-if-nve-vni)# mcast-group 255.0.4.1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> ingress-replication protocol bgp</td>
<td>Enables BGP EVPN with ingress replication for the VNI within the fabric.</td>
</tr>
<tr>
<td>Example: switch(config-if-nve-vni)# ingress-replication protocol bgp</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> multisite ingress-replication</td>
<td>Defines the Multi-Site BUM replication method. Per-VNI knob for extending Layer 2 VNI.</td>
</tr>
<tr>
<td>Example: switch(config-if-nve-vni)# multisite ingress-replication</td>
<td></td>
</tr>
</tbody>
</table>
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 re-origination of EVPN routes either from or to a site if all the DCI/fabric links go down.

**Procedure**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><code>configure terminal</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>switch# configure terminal</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Enters interface configuration mode for DCI interface.</td>
</tr>
<tr>
<td><code>interface ethernet port</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>switch(config)# interface ethernet1/1</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>Enter one of the following commands in Step 3 or Step 4.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Configure DCI interface tracking.</td>
</tr>
<tr>
<td><code>evpn multisite dci-tracking</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>switch(config-if)# evpn multisite dci-tracking</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Enters interface configuration mode for fabric interface.</td>
</tr>
<tr>
<td><code>interface ethernet port</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>switch(config)# interface ethernet1/2</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>Enters interface configuration mode for fabric interface.</td>
</tr>
<tr>
<td><code>evpn multisite fabric-tracking</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>switch(config-if)# evpn multisite fabric-tracking</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>Configure IP features.</td>
</tr>
<tr>
<td><code>ip address ip-addr</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>switch(config-if)# ip address 192.1.1.1</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>Negate shutdown command.</td>
</tr>
<tr>
<td><code>no shutdown</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>switch(config-if)# no shutdown</td>
</tr>
</tbody>
</table>

Configuring Fabric External Neighbors

This procedure describes the configuration of Fabric External/DCI Neighbors for communication to other site/fabric border gateways.
### Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1**  
configure terminal  
Example:  
`switch# configure terminal` | Enters global configuration mode. |
| **Step 2**  
router bgp as-num  
Example:  
`switch(config)# router bgp 100` | Configure 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` | Configure a BGP neighbor. |
| **Step 4**  
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 border gateways. It is not required for pseudo border gateways. |
| **Step 5**  
address-family l2vpn evpn  
Example:  
`switch(config-router-neighbor)# address-family l2vpn evpn` | |
| **Step 6**  
rewrite-evpn-rt-asn  
Example:  
`switch(config-router-neighbor)# rewrite-evpn-rt-asn` | Rewrites the route target information to simplify MAC-VRF and IP-VRF configuration. Normalizes the outgoing route target's AS number to match the remote AS number. Uses the BGP configured neighbors remote AS. |
Configuring Fabric External Neighbors
CHAPTER 9

Configuring Tenant Routed Multicast

This chapter contains the following sections:

- About Tenant Routed Multicast, on page 165
- About Tenant Routed Multicast Mixed Mode, on page 167
- Guidelines and Limitations for Tenant Routed Multicast, on page 167
- Guidelines and Limitations for Layer 3 Tenant Routed Multicast, on page 168
- Guidelines and Limitations for Layer 2/Layer 3 Tenant Routed Multicast (Mixed Mode), on page 168
- Rendezvous Point for Tenant Routed Multicast, on page 169
- Configuring a Rendezvous Point for Tenant Routed Multicast, on page 169
- Configuring RP Everywhere with PIM Anycast, on page 173
- Configuring RP Everywhere with MSDP Peering, on page 178
- Configuring Layer 3 Tenant Routed Multicast, on page 184
- Configuring TRM on the VXLAN EVPN Spine, on page 188
- Configuring Tenant Routed Multicast in Layer 2/Layer 3 Mixed Mode, on page 190
- Configuring Layer 2 Tenant Routed Multicast, on page 195

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.
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

Guidelines and Limitations for Tenant Routed Multicast

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

- With TRM enabled, `advertise-pip` and `advertise virtual-rmac` configurations are not supported.
- The Guidelines and Limitations for VXLAN, on page 13 also apply to TRM.
- With TRM enabled, FEX is not supported.
- Beginning with Cisco NX-OS Release 7.0(3)I7(2), the VXLAN EVPN Multi-site Border Gateway (BGW) and TRM Border Leaf can co-exist on the same physical switch.
  - Within EVPN Multi-Site, TRM enabled East-West multicast traffic is not supported. In case the same external RP is used for multiple sites, overlapping multicast groups between sites must be avoided.
- 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.
• Coexistence of vPC and external PIM peering is not supported. A border leaf that is also a part of a vPC domain cannot hold TRM external connectivity for multicast.

• Internal RP: The internal RP must be configured on all TRM-enabled VTEPs including the TRM border.

• External RP: The external RP must be external to the TRM border.
  • 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.

• RP Everywhere: When configuring RP-Everywhere, the fabric TRM border and external RPs must be in a PIM Anycast-RP Set.

• TRM supports multiple border nodes. Beginning with Cisco NX-OS Release 7.0(3)I7(6), reachability to an external RP via multiple border leaf switches is supported (ECMP). In prior releases, the external RP could only be reachable via a single border leaf (non ECMP).

Guidelines and Limitations for Layer 3 Tenant Routed Multicast

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

• When configuring TRM VXLAN BGP EVPN, the following platforms are supported:
  • Cisco Nexus 9200, 9300-EX, 9300-FX, and 9300-FX2
  • Cisco Nexus 9500 platform switches with 9700-EX line cards, 9700-FX line cards, or a combination of both line cards.

• Layer 3 mode is supported only for Cloud Scale Nexus 9000 Series switches.

• Well known local scope multicast (224.0.0.0/24) is excluded from TRM and is bridged.

• Whenever the NVE interface is brought down on the TRM border, the internal overlay RP (per VRF) must also 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.

• All anchor DR must perform the overlay RP.

• Cisco Nexus 9000 Series switches that operate in non-TRM or Layer 2 TRM mode are using Control-Plane based signalization (BGP IMET/SMET).

• 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.

• Whenever the NVE interface is brought down on the TRM border, the internal overlay RP (per VRF) must also be brought down.

• Anchor DR is supported only on the following hardware platforms:
  • Cisco Nexus 9200, 9300-EX, 9300-FX, and 9300-FX2
  • Cisco Nexus 9500 platform switches with 9700-EX line cards, 9700-FX line cards, or a combination of both line cards.

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.

<table>
<thead>
<tr>
<th></th>
<th>RP Internal</th>
<th>RP External</th>
<th>PIM-Based RP Everywhere</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRM L2 Mode</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>TRM L3 Mode</td>
<td>7.0(3)I7(1), 9.2(x)</td>
<td>7.0(3)I7(4), 9.2(3)</td>
<td>Supported in 7.0(3)I7(x) releases starting from 7.0(3)I7(5) Not supported in 9.2(x)</td>
</tr>
<tr>
<td>TRM L2L3 Mode</td>
<td>7.0(3)I7(1), 9.2(x)</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Configuring a Rendezvous Point for Tenant Routed Multicast

For Tenant Routed Multicast, there are four rendezvous point options:

• Configuring a Rendezvous Point Inside the VXLAN Fabric, on page 170
• Configuring an External Rendezvous Point, on page 171
• Configuring RP Everywhere with PIM Anycast, on page 173
  • Configuring a TRM Leaf Node for RP Everywhere with PIM Anycast, on page 173
  • Configuring a TRM Border Leaf Node for RP Everywhere with PIM Anycast, on page 174
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).

Figure 36:
Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enter configuration mode.</td>
</tr>
<tr>
<td>Example: switch# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> interface loopback loopback_number</td>
<td>Configure the loopback interface on all distributed TRM enabled nodes.</td>
</tr>
<tr>
<td>Example: switch(config)# interface loopback 11</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> vrf member vxlan-number</td>
<td>Configure VRF name.</td>
</tr>
<tr>
<td>Example: switch(config-if)# vrf member vrf100</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> ip address ip-address</td>
<td>Specify IP address.</td>
</tr>
<tr>
<td>Example: switch(config-if)# ip address 209.165.200.1/32</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> ip pim sparse-mode</td>
<td>Configure sparse-mode PIM on an interface.</td>
</tr>
<tr>
<td>Example: switch(config-if)# ip pim sparse-mode</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> vrf context vrf-name</td>
<td>Create a VXLAN tenant VRF.</td>
</tr>
<tr>
<td>Example: switch(config-if)# vrf context vrf100</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> ip pim rp-address ip-address-of-router group-list group-range-prefix</td>
<td>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.</td>
</tr>
<tr>
<td>Example: switch(config-vrf)# ip pim rp-address 209.165.200.1 group-list 224.0.0.0/4</td>
<td></td>
</tr>
</tbody>
</table>

**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. With TRM enabled and an external RP in use, ensure that only one routing path is active. Routing between the TRM fabric and the external RP must be via a single border leaf (non ECMP).
### Procedure

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>configure terminal</code></td>
<td>Enter configuration mode.</td>
</tr>
<tr>
<td>Example</td>
<td>switch# configure terminal</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td><code>vrf context vrf100</code></td>
<td>Enter configuration mode.</td>
</tr>
<tr>
<td>Example</td>
<td>switch(config)# vrf context vrf100</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td><code>ip pim rp-address ip-address-of-router</code></td>
<td>The value of the <code>ip-address-of-router</code> parameter is that of the RP. The same IP address must be on all of the edge devices (VTEPs) for a fully distributed RP.</td>
</tr>
<tr>
<td>Example</td>
<td>switch(config-vrf)# ip pim rp-address 209.165.200.1 group-list 224.0.0.0/4</td>
<td></td>
</tr>
</tbody>
</table>
Configuring RP Everywhere with PIM Anycast

RP Everywhere configuration with PIM Anycast solution.

Configuring a TRM Leaf Node for RP Everywhere with PIM Anycast

Configuration of Tenant Routed Multicast (TRM) leaf node for RP Everywhere.
Configuring a TRM Border Leaf Node for RP Everywhere with PIM Anycast

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

Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enter configuration mode.</td>
</tr>
<tr>
<td>Example: switch# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> interface loopback  loopback_number</td>
<td>Configure the loopback interface on all VXLAN VTEP devices.</td>
</tr>
<tr>
<td>Example: switch(config)# interface loopback 11</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> vrf member  vrf-name</td>
<td>Configure VRF name.</td>
</tr>
<tr>
<td>Example: switch(config-if)# vrf member vrf100</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> ip address  ip-address</td>
<td>Specify IP address.</td>
</tr>
<tr>
<td>Example: switch(config-if)# ip address 209.165.200.1/32</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> ip pim sparse-mode</td>
<td>Configure sparse-mode PIM on an interface.</td>
</tr>
<tr>
<td>Example: switch(config-if)# ip pim sparse-mode</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> vrf context  vxlan</td>
<td>Create a VXLAN tenant VRF.</td>
</tr>
<tr>
<td>Example: switch(config-if)# vrf context vrf100</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> ip pim rp-address  ip-address-of-router  group-list  group-range-prefix</td>
<td>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.</td>
</tr>
<tr>
<td>Example: switch(config-vrf)# ip pim rp-address 209.165.200.1 group-list 224.0.0.0/4</td>
<td></td>
</tr>
<tr>
<td>Step</td>
<td>Command or Action</td>
</tr>
<tr>
<td>------</td>
<td>------------------</td>
</tr>
</tbody>
</table>
| 2    | ip pim evpn-border-leaf  
Example:  
switch(config)# ip pim evpn-border-leaf | Configure VXLAN VTEP as TRM border leaf node. |
| 3    | interface loopback  
Example:  
switch(config)# interface loopback 11 | Configure the loopback interface on all VXLAN VTEP devices. |
| 4    | vrf member  
Example:  
switch(config-if)# vrf member vrf100 | Configure VRF name. |
| 5    | ip address  
Example:  
switch(config-if)# ip address 209.165.200.1/32 | Specify IP address. |
| 6    | ip pim sparse-mode  
Example:  
switch(config-if)# ip pim sparse-mode | Configure sparse-mode PIM on an interface. |
| 7    | interface loopback  
Example:  
switch(config)# interface loopback 12 | Configure the PIM Anycast set RP loopback interface. |
| 8    | vrf member  
Example:  
switch(config-if)# vrf member vxlan-number | Configure VRF name. |
| 9    | ip address  
Example:  
switch(config-if)# ip address 209.165.200.11/32 | Specify IP address. |
| 10   | ip pim sparse-mode  
Example:  
switch(config-if)# ip pim sparse-mode | Configure sparse-mode PIM on an interface. |
| 11   | vrf context  
Example:  
switch(config-if)# vrf context vrf100 | Create a VXLAN tenant VRF. |
| 12   | ip pim rp-address  
Example:  
switch(config-if)# ip pim rp-address ip-address-of-router  
group-list group-range-prefix | The value of the ip-address-of-router parameters is that of the RP. The same IP |
## Configuring an External Router for RP Everywhere with PIM Anycast

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

### Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>switch(config-vrf)# ip pim anycast-rp address-of-rp</code></td>
<td>address must be on all the edge devices (VTEPs) for a fully distributed RP.</td>
</tr>
<tr>
<td><strong>Step 13</strong></td>
<td></td>
</tr>
<tr>
<td><code>ip pim anycast-rp anycast-rp-address address-of-rp</code></td>
<td>Configure PIM Anycast RP set.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Step 14</strong></td>
<td></td>
</tr>
<tr>
<td><code>ip pim anycast-rp anycast-rp-address address-of-rp</code></td>
<td>Configure PIM Anycast RP set.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Step 15</strong></td>
<td></td>
</tr>
<tr>
<td><code>ip pim anycast-rp anycast-rp-address address-of-rp</code></td>
<td>Configure PIM Anycast RP set.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Step 16</strong></td>
<td></td>
</tr>
<tr>
<td><code>ip pim anycast-rp anycast-rp-address address-of-rp</code></td>
<td>Configure PIM Anycast RP set.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-----------------------------------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>switch(config-if)# vrf member vfr100</td>
<td></td>
</tr>
</tbody>
</table>

**Step 4**

<table>
<thead>
<tr>
<th>ip address ip-address</th>
<th>Specify IP address.</th>
</tr>
</thead>
</table>
| Example:
  switch(config-if)# ip address
  209.165.200.1/32                                           |                     |

**Step 5**

<table>
<thead>
<tr>
<th>ip pim sparse-mode</th>
<th>Configure sparse-mode PIM on an interface.</th>
</tr>
</thead>
</table>
| Example:
  switch(config-if)# ip pim sparse-mode                   |                                           |

**Step 6**

<table>
<thead>
<tr>
<th>interface loopback loopback_number</th>
<th>Configure the PIM Anycast set RP loopback interface.</th>
</tr>
</thead>
</table>
| Example:
  switch(config)# interface loopback 12                 |                                                    |

**Step 7**

<table>
<thead>
<tr>
<th>vrf member vxlan-number</th>
<th>Configure VRF name.</th>
</tr>
</thead>
</table>
| Example:
  switch(config-if)# vrf member vrf100                   |                                                    |

**Step 8**

<table>
<thead>
<tr>
<th>ip address ip-address</th>
<th>Specify IP address.</th>
</tr>
</thead>
</table>
| Example:
  switch(config-if)# ip address
  209.165.200.13/32                                         |                                                    |

**Step 9**

<table>
<thead>
<tr>
<th>ip pim sparse-mode</th>
<th>Configure sparse-mode PIM on an interface.</th>
</tr>
</thead>
</table>
| Example:
  switch(config-if)# ip pim sparse-mode                   |                                                    |

**Step 10**

<table>
<thead>
<tr>
<th>vrf context vxlan</th>
<th>Create a VXLAN tenant VRF.</th>
</tr>
</thead>
</table>
| Example:
  switch(config-if)# vrf context vrf100                   |                                                    |

**Step 11**

<table>
<thead>
<tr>
<th>ip pim rp-address ip-address-of-router group-list group-range-prefix</th>
<th>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.</th>
</tr>
</thead>
</table>
| Example:
  switch(config-vrf)# ip pim rp-address
  209.165.200.1 group-list 224.0.0.0/4                                |                                                                                                 |

**Step 12**

<table>
<thead>
<tr>
<th>ip pim anycast-rp anycast-rp-address address-of-rp</th>
<th>Configure PIM Anycast RP set.</th>
</tr>
</thead>
</table>
| Example:
  switch(config-vrf)# ip pim anycast-rp
  209.165.200.1 209.165.200.11                                     |                                                    |

**Step 13**

<table>
<thead>
<tr>
<th>ip pim anycast-rp anycast-rp-address address-of-rp</th>
<th>Configure PIM Anycast RP set.</th>
</tr>
</thead>
</table>
| Example:
  switch(config-vrf)# ip pim anycast-rp
  209.165.200.1 209.165.200.11                                     |                                                    |
### Command or Action

<table>
<thead>
<tr>
<th>Example:</th>
</tr>
</thead>
<tbody>
<tr>
<td>switch(config-vrf)# <code>ip pim anycast-rp 209.165.200.1 209.165.200.12</code></td>
</tr>
</tbody>
</table>

#### Purpose
- **Step 14**  `ip pim anycast-rp anycast-rp-address address-of-rp`
  - Configure PIM Anycast RP set.

Example:
- `switch(config-vrf)# ip pim anycast-rp 209.165.200.1 209.165.200.13`

#### Purpose
- **Step 15**  `ip pim anycast-rp anycast-rp-address address-of-rp`
  - Configure PIM Anycast RP set.

Example:
- `switch(config-vrf)# ip pim anycast-rp 209.165.200.1 209.165.200.14`

---

### Configuring RP Everywhere with MSDP Peering

RP Everywhere configuration with MSDP RP solution.
### Configuring a TRM Leaf Node for RP Everywhere with MSDP Peering

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

#### Procedure

<table>
<thead>
<tr>
<th>Step 1</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>configure terminal</td>
<td>Enter configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>switch# configure terminal</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td>interface loopback loopback_number</td>
<td>Configure the loopback interface on all VXLAN VTEP devices.</td>
</tr>
<tr>
<td>Example:</td>
<td>switch(config)# interface loopback 11</td>
<td></td>
</tr>
</tbody>
</table>
### Configuration of VRF Context

**Command or Action:** vrf member vrf-name

**Example:**
```
switch(config-if)# vrf member vrf100
```

**Purpose:** Configure VRF name.

**Step 4**

**Command or Action:** ip address ip-address

**Example:**
```
switch(config-if)# ip address 209.165.200.1/32
```

**Purpose:** Specify IP address.

**Step 5**

**Command or Action:** ip pim sparse-mode

**Example:**
```
switch(config-if)# ip pim sparse-mode
```

**Purpose:** Configure sparse-mode PIM on an interface.

**Step 6**

**Command or Action:** vrf context vrf-name

**Example:**
```
switch(config-if)# vrf context vrf100
```

**Purpose:** Create a VXLAN tenant VRF.

**Step 7**

**Command or Action:** 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
```

**Purpose:** The value of the ip-address-of-router parameters is that of the RP. The same IP address must be on all the edge devices (VTEPs) for a fully distributed RP.

---

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

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

**Procedure**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure terminal</td>
<td>Enter configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>switch# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>feature msdp</td>
<td>Enable feature MSDP.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>switch(config)# feature msdp</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>ip pim evpn-border-leaf</td>
<td>Configure VXLAN VTEP as TRM border leaf node,</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>switch(config)# ip pim evpn-border-leaf</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>interface loopback loopback_number</td>
<td>Configure the loopback interface on all VXLAN VTEP devices.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>switch(config)# interface loopback 2</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td><code>switch(config)# interface loopback 11</code></td>
<td>Configure VRF name.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td><strong>vrf member vrf-name</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;<code>switch(config-if)# vrf member vrf100</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td><strong>ip address ip-address</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;<code>switch(config-if)# ip address 209.165.200.1/32</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td><strong>ip pim sparse-mode</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;<code>switch(config-if)# ip pim sparse-mode</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td><strong>interface loopback loopback_number</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;<code>switch(config)# interface loopback 12</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td><strong>vrf member vrf-name</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;<code>switch(config-if)# vrf member vrf100</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 10</strong></td>
<td><strong>ip address ip-address</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;<code>switch(config-if)# ip address 209.165.200.11/32</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 11</strong></td>
<td><strong>ip pim sparse-mode</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;<code>switch(config-if)# ip pim sparse-mode</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 12</strong></td>
<td><strong>vrf context vrf-name</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;<code>switch(config-if)# vrf context vrf100</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 13</strong></td>
<td><strong>ip pim rp-address ip-address-of-router group-list group-range-prefix</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;<code>switch(config-vrf)# ip pim rp-address 209.165.200.1 group-list 224.0.0.0/4</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 14</strong></td>
<td><strong>ip pim anycast-rp anycast-rp-address address-of-rp</strong>&lt;br&gt;<strong>Example:</strong></td>
<td>Configure PIM Anycast RP set.</td>
</tr>
</tbody>
</table>
Configuring an External Router for RP Everywhere with MSDP Peering

Procedure

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>configure terminal</td>
<td>Enter configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch# configure terminal</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>feature msdp</td>
<td>Enable feature MSDP.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config)# feature msdp</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>interface loopback loopback_number</td>
<td>Configure the loopback interface on all VXLAN VTEP devices.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config)# interface loopback 1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>vrf member vrf-name</td>
<td>Configure VRF name.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config-if)# vrf member vrf100</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>ip address ip-address</td>
<td>Specify IP address.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config-if)# ip address 209.165.201.1/32</td>
<td></td>
</tr>
<tr>
<td>Step</td>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------</td>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td>6</td>
<td><code>ip pim sparse-mode</code></td>
<td>Configure sparse-mode PIM on an interface.</td>
</tr>
</tbody>
</table>
|      | **Example:**  
|      | `switch(config-if)# ip pim sparse-mode` | |
| 7    | `interface loopback` *loopback_number* | Configure the PIM Anycast set RP loopback interface. |
|      | **Example:**  
|      | `switch(config)# interface loopback 12` | |
| 8    | `vrf member` *vrf-name* | Configure VRF name. |
|      | **Example:**  
|      | `switch(config-if)# vrf member vrf100` | |
| 9    | `ip address` *ip-address* | Specify IP address. |
|      | **Example:**  
|      | `switch(config-if)# ip address 209.165.201.11/32` | |
| 10   | `ip pim sparse-mode` | Configure sparse-mode PIM on an interface. |
|      | **Example:**  
|      | `switch(config-if)# ip pim sparse-mode` | |
| 11   | `vrf context` *vrf-name* | Create a VXLAN tenant VRF. |
|      | **Example:**  
|      | `switch(config-if)# vrf context vrf100` | |
| 12   | `ip pim rp-address` *ip-address-of-router*  
|      | `group-list` *group-range-prefix* | 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. |
|      | **Example:**  
|      | `switch(config-vrf)# ip pim rp-address 209.165.201.1 group-list 224.0.0.0/4` | |
| 13   | `ip msdp originator-id loopback12` | Configure MSDP originator ID. |
|      | **Example:**  
|      | `switch(config-vrf)# ip msdp originator-id loopback12` | |
| 14   | `ip msdp peer` *ip-address*  
|      | `connect-source` *loopback12* | Configure MSDP peering between external RP router and all TRM border nodes. |
|      | **Example:**  
|      | `switch(config-vrf)# ip msdp peer 209.165.200.11 connect-source loopback12` | |
Configuring Layer 3 Tenant Routed Multicast

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

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

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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong>configure terminal</td>
<td>Enter configuration mode.</td>
</tr>
<tr>
<td>**Example:**switch# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong>feature ngmvpn</td>
<td>Enables the Next-Generation Multicast VPN (ngMVPN) control plane. New address family commands become available in BGP.</td>
</tr>
<tr>
<td>**Example:**switch(config)# feature ngmvpn</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong>ip igmp snooping vxlan</td>
<td>Configure IGMP snooping for VXLANs.</td>
</tr>
<tr>
<td>**Example:**switch(config)# ip igmp snooping vxlan</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong>interface nve1</td>
<td>Configure the NVE interface.</td>
</tr>
<tr>
<td>**Example:**switch(config)# interface nve 1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong>member vni vni-range associate-vrf</td>
<td>Configure the virtual network identifier. The range of vni-range is from 1 to 16,777,214.</td>
</tr>
<tr>
<td>**Example:**switch(config-if-nve)# member vni 200100 associate-vrf</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong>mcast-group ip-prefix</td>
<td>Builds the default multicast distribution tree for the VRF VNI (Layer 3 VNI).</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td>switch(config-if-nve-vni)# mcast-group 225.3.3.3</td>
<td>The multicast group is used in the underlay (core) for all multicast routing within the associated Layer 3 VNI (VRF).</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>We recommend that underlay multicast groups for Layer 2 VNI, default MDT, and data MDT not be shared. Use separate, non-overlapping groups.</td>
</tr>
</tbody>
</table>

| Step 7 | **exit** | Exits command mode. |
| Example: | | |
| switch(config-if-nve-vni)# exit | | |

| Step 8 | **exit** | Exits command mode. |
| Example: | | |
| switch(config-if)# exit | | |

| Step 9 | **router bgp 100** | Set autonomous system number. |
| Example: | | |
| switch(config)# router bgp 100 | | |

| Step 10 | **exit** | Exits command mode. |
| Example: | | |
| switch(config-router)# exit | | |

| Step 11 | **neighbor ip-addr** | Configure IP address of the neighbor. |
| Example: | | |
| switch(config-router)# neighbor 1.1.1.1 | | |

| Step 12 | **address-family ipv4 mvpn** | Configure multicast VPN. |
| Example: | | |
| switch(config-router-neighbor)# address-family ipv4 mvpn | | |

| Step 13 | **send-community extended** | Enables ngMVPN for address family signalization. The send community both command ensures both standard and extended communities are exchanged for this address family. |
| Example: | | |
| switch(config-router-neighbor-af)# send-community extended | | |

<p>| Step 14 | <strong>exit</strong> | Exits command mode. |
| Example: | | |
| switch(config-router-neighbor-af)# exit | | |</p>
<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td><code>exit</code></td>
<td>Exits command mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>switch(config-router)# exit</code></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td><code>vrf context vrf_name</code></td>
<td>Configure VRF name.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>switch(config-router)#vrf context vrf100</code></td>
<td></td>
</tr>
</tbody>
</table>
| 17    | `ip pim rp-address ip-address-of-router
group-list group-range-prefix` | 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 169 section. |
<p>|       | Example:         |         |
|       | <code>switch(config-vrf)# ip pim rp-address 99.99.99.1 group-list 226.0.0.0/8</code> |         |
| 18    | <code>address-family ipv4 unicast</code> | Configure unicast address family. |
|       | Example:         |         |
|       | <code>switch(config-vrf)# address-family ipv4 unicast</code> |         |
| 19    | <code>route-target both auto mvpn</code> | 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. |
|       | Example:         |         |
|       | <code>switch(config-vrf-af-ipv4)# route-target both auto mvpn</code> |         |
| 20    | <code>ip multicast overlay-spt-only</code> | Add the <code>ip multicast overlay-spt-only</code> command on all MVPN-enabled Cisco Nexus 9000 Series switches (typically leaf node). |
|       | Example:         |         |
|       | <code>switch(config)# ip multicast overlay-spt-only</code> |         |
| 21    | <code>interface vlan_id</code> | Configures the first-hop gateway (distributed anycast gateway for the Layer 2 VNI. No router PIM peering must ever happen with this interface. |
|       | Example:         |         |
|       | <code>switch(config)# interface vlan11</code> |         |
| 22    | <code>no shutdown</code> | Disables an interface. |
|       | Example:         |         |
|       | <code>switch(config-if)# no shutdown</code> |         |
| 23    | <code>vrf member vrf-num</code> | Configure VRF name. |
|       | Example:         |         |
|       | <code>switch(config-if)# vrf member vrf100</code> |         |</p>
<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>ip address ip_address</td>
<td>Configure IP address.</td>
</tr>
<tr>
<td></td>
<td>Example: switch(config-if)# ip address 11.1.1.1/24</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>ip pim sparse-mode</td>
<td>Enables IGMP and PIM on the SVI. This is required if multicast sources and/or receivers exist in this VLAN.</td>
</tr>
<tr>
<td></td>
<td>Example: switch(config-if)# ip pim sparse-mode</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>fabric forwarding mode anycast-gateway</td>
<td>Configure Anycast Gateway Forwarding Mode.</td>
</tr>
<tr>
<td></td>
<td>Example: switch(config-if)# fabric forwarding mode anycast-gateway</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>ip pim neighbor-policy NONE*</td>
<td>Creates an IP PIM neighbor policy to avoid PIM neighborship with PIM routers within the VLAN. The none keyword is a configured route map to deny any ipv4 addresses to avoid establishing PIM neighborship policy using anycast IP.</td>
</tr>
<tr>
<td></td>
<td>Example: switch(config-if)# ip pim neighbor-policy NONE*</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Note: Do not use Distributed Anycast Gateway for PIM Peerings.</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>exit</td>
<td>Exits command mode.</td>
</tr>
<tr>
<td></td>
<td>Example: switch(config-if)# exit</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>interface vlan_id</td>
<td>Configure Layer 3 VNI.</td>
</tr>
<tr>
<td></td>
<td>Example: switch(config)# interface vlan100</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>no shutdown</td>
<td>Disable an interface.</td>
</tr>
<tr>
<td></td>
<td>Example: switch(config-if)# no shutdown</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>vrf member vrf100</td>
<td>Configure VRF name.</td>
</tr>
<tr>
<td></td>
<td>Example: switch(config-if)# vrf member vrf100</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>ip forward</td>
<td>Enable IP forwarding on interface.</td>
</tr>
<tr>
<td></td>
<td>Example: switch(config-if)# ip forward</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>ip pim sparse-mode</td>
<td>Configure sparse-mode PIM on interface. There is no PIM peering happening in the</td>
</tr>
</tbody>
</table>
### 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 BGP for EVPN on the Spine, on page 88.

**Procedure**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>configure terminal</td>
<td>Enter configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example: switch# configure terminal</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>route-map permitall permit 10</td>
<td>Configure the route-map. Note The route-map keeps the next-hop unchanged for EVPN routes • Required for eBGP • Options for iBGP</td>
</tr>
<tr>
<td></td>
<td>Example: switch(config)# route-map permitall permit 10</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>set ip next-hop unchanged</td>
<td>Set next hop address. Note The route-map keeps the next-hop unchanged for EVPN routes • Required for eBGP • Options for iBGP</td>
</tr>
<tr>
<td></td>
<td>Example: switch(config-route-map)# set ip next-hop unchanged</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>exit</td>
<td>Return to exec mode.</td>
</tr>
<tr>
<td></td>
<td>Example: switch(config-route-map)# exit</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>router bgp [autonomous system] number</td>
<td>Specify BGP.</td>
</tr>
<tr>
<td></td>
<td>Example: switch(config)# router bgp 65002</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>address-family ipv4 mvpn</td>
<td>Configure the address family IPv4 MVPN under the BGP.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td><strong>switch(config-router)# address-family ipv4 mvpn</strong></td>
<td>Configure retain route-target all under address-family IPv4 MVPN [global].</td>
<td></td>
</tr>
<tr>
<td><strong>Switch 7 retain route-target all</strong></td>
<td>Configure retain route-target all under address-family IPv4 MVPN [global]. <strong>Note</strong> 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.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8 neighbor ip-address [remote-as number]</strong></td>
<td>Define neighbor.</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>switch(config-router-af)# neighbor 100.100.100.1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 9 address-family ipv4 mvpn</strong></td>
<td>Configure address family IPv4 MVPN under the BGP neighbor.</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>switch(config-router-neighbor)# address-family ipv4 mvpn</td>
<td></td>
</tr>
<tr>
<td><strong>Step 10 disable-peer-as-check</strong></td>
<td>Disables checking the peer AS number during route advertisement. Configure this parameter on the spine for eBGP when all leaves are using the same AS but the spines have a different AS than leaves. <strong>Note</strong> Required for eBGP.</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>switch(config-router-neighbor-af)# disable-peer-as-check</td>
<td></td>
</tr>
<tr>
<td><strong>Step 11 rewrite-rt-asn</strong></td>
<td>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.</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>switch(config-router-neighbor-af)# rewrite-rt-asn</td>
<td></td>
</tr>
<tr>
<td><strong>Step 12 send-community extended</strong></td>
<td>Configures community for BGP neighbors.</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>switch(config-router-neighbor-af)# send-community extended</td>
<td></td>
</tr>
<tr>
<td><strong>Step 13 route-reflector-client</strong></td>
<td>Configure route reflector. <strong>Note</strong> Required for iBGP with route-reflector.</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>switch(config-router-neighbor-af)# route-reflector-client</td>
<td></td>
</tr>
<tr>
<td><strong>Step 14 route-map permit all out</strong></td>
<td>Applies route-map to keep the next-hop unchanged.</td>
<td></td>
</tr>
</tbody>
</table>
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 same such as Cisco Nexus 9000 Series switches (1st generation) or Cisco Nexus 7000 Series switches. 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

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>configure terminal</td>
<td>Enter configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch# configure terminal</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>feature ngmvpn</td>
<td>Enables the Next-Generation Multicast VPN (ngMVPN) control plane. New address family commands become available in BGP.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config)# feature ngmvpn</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>advertise evpn multicast</td>
<td>Advertises IMET and SMEt routes into BGP EVPN towards non-TRM capable switches.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config)# advertise evpn multicast</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>ip igmp snooping vxlan</td>
<td>Configure IGMP snooping for VXLANs.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config)# ip igmp snooping vxlan</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>ip multicast overlay-spt-only</td>
<td>Gratuitously originate (S,A) route when source is locally connected.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config)# ip multicast overlay-spt-only</td>
<td></td>
</tr>
<tr>
<td>Step</td>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------</td>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td>6</td>
<td>ip multicast overlay-distributed-dr</td>
<td>Enables distributed anchor DR function on this VTEP.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config)# ip multicast overlay-distributed-dr</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>interface nve1</td>
<td>Configure the NVE interface.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config)# interface nve 1</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>member vni vni-range associate-vrf</td>
<td>Configure the virtual network identifier. The range of vni-range is from 1 to 16,777,214.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config-if-nve)# member vni 200100 associate-vrf</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>mcast-group ip-prefix</td>
<td>Configures the multicast group on distributed anchor DR.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config-if-nve-vni)# mcast-group 225.3.3.3</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>exit</td>
<td>Exits command mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config-if-nve-vni)# exit</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>interface loopback loopback_number</td>
<td>Configure the loopback interface on all distributed anchor DR devices.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config-if-nve)# interface loopback 10</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>ip address ip_address</td>
<td>Configure IP address. This IP address is the same on all distributed anchor DR.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config-if)# ip address 100.100.1.1/32</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>ip router ospf process-tag area ospf-id</td>
<td>OSPF area ID in IP address format.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config-if)# ip router ospf 100 area 0.0.0.0</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>ip pim sparse-mode</td>
<td>Configure sparse-mode PIM on interface.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config-if)# ip pim sparse-mode</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>interface nve1</td>
<td>Configure NVE interface.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>switch(config-if)# interface nve1</td>
<td></td>
</tr>
<tr>
<td>Step</td>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------</td>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td>16</td>
<td>shutdown</td>
<td>Disable the interface.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>switch(config-if-nve)# shutdown</td>
</tr>
<tr>
<td>17</td>
<td>mcast-routing override source-interface loopback int-num</td>
<td>Enables that TRM is using a different source-interface interface than the VTEPs default source-interface. The loopback variable must be configured on every TRM-enabled VTEP (Anchor DR) in the underlay. This loopback and the respective override command is needed to serve TRM VTEPs in co-existence with non-TRM VTEPs. The loopback has to be configured with the same IP address.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>switch(config-if-nve)# mcast-routing override source-interface loopback 10</td>
</tr>
<tr>
<td>18</td>
<td>exit</td>
<td>Exits command mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>switch(config-if-nve)# exit</td>
</tr>
<tr>
<td>19</td>
<td>router bgp 100</td>
<td>Set autonomous system number.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>switch(config)# router bgp 100</td>
</tr>
<tr>
<td>20</td>
<td>neighbor ip-addr</td>
<td>Configure IP address of the neighbor.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>switch(config-router)# neighbor 1.1.1.1</td>
</tr>
<tr>
<td>21</td>
<td>address-family ipv4 mvpn</td>
<td>Configure multicast VPN.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>switch(config-router-neighbor)# address-family ipv4 mvpn</td>
</tr>
<tr>
<td>22</td>
<td>send-community extended</td>
<td>Send community attribute.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>switch(config-router-neighbor-af)# send-community extended</td>
</tr>
<tr>
<td>23</td>
<td>exit</td>
<td>Exits command mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>switch(config-router-neighbor-af)# exit</td>
</tr>
<tr>
<td>24</td>
<td>exit</td>
<td>Exits command mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>switch(config-router)# exit</td>
</tr>
<tr>
<td>Step</td>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------</td>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td>25</td>
<td><code>vrf vrf_name vrf100</code></td>
<td>Configure VRF name.</td>
</tr>
</tbody>
</table>
|      | **Example:**
|      | `switch(config)# vrf context vrf100` | |
| 26   | `ip pim rp-address ip-address-of-router`<br>`group-list group-range-prefix` | 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 169 - Internal RP section. |
|      | **Example:**
|      | `switch(config-vrf)# ip pim rp-address 99.99.99.1 group-list 226.0.0.0/8` | |
| 27   | `address-family ipv4 unicast` | Configure unicast address family. |
|      | **Example:**
|      | `switch(config-vrf)# address-family ipv4 unicast` | |
| 28   | `route-target both auto mvnp` | Specify target for mvpn routes. |
|      | **Example:**
|      | `switch(config-vrf-af-ipv4)# route-target both auto mvnp` | |
| 29   | `exit` | Exits command mode. |
|      | **Example:**
|      | `switch(config-vrf-af-ipv4)# exit` | |
| 30   | `exit` | Exits command mode. |
|      | **Example:**
|      | `switch(config-vrf)# exit` | |
| 31   | `interface vlan_id` | Configure Layer 2 VNI. |
|      | **Example:**
|      | `switch(config)# interface vlan11` | |
| 32   | `no shutdown` | Disable an interface. |
|      | **Example:**
|      | `switch(config-if)# no shutdown` | |
| 33   | `vrf member vrf100` | Configure VRF name. |
|      | **Example:**
<p>|      | <code>switch(config-if)# vrf member vrf100</code> | |
| 34   | <code>ip address ip_address</code> | Configure IP address. |
|      | <strong>Example:</strong> | |</p>
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>switch(config-if)# ip address 11.1.1.1/24</td>
<td></td>
</tr>
<tr>
<td><strong>Step 35</strong></td>
<td>ip pim sparse-mode</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>configure sparse-mode PIM on interface.</td>
</tr>
<tr>
<td>e</td>
<td>switch(config-if)# ip pim sparse-mode</td>
</tr>
<tr>
<td><strong>Step 36</strong></td>
<td>fabric forwarding mode anycast-gateway</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>configure Anycast Gateway Forwarding Mode.</td>
</tr>
<tr>
<td>switch(config-if)# fabric forwarding mode anycast-gateway</td>
<td></td>
</tr>
<tr>
<td><strong>Step 37</strong></td>
<td>ip pim neighbor-policy NONE*</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>The <code>none</code> keyword is a configured route map to deny any IPv4 addresses to avoid establishing a PIM neighborship policy using anycase IP.</td>
</tr>
<tr>
<td>switch(config-if)# ip pim neighbor-policy NONE*</td>
<td></td>
</tr>
<tr>
<td><strong>Step 38</strong></td>
<td>exit</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>exits command mode.</td>
</tr>
<tr>
<td>switch(config-if)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 39</strong></td>
<td>interface vlan_id</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>configure Layer 3 VNI.</td>
</tr>
<tr>
<td>switch(config)# interface vlan100</td>
<td></td>
</tr>
<tr>
<td><strong>Step 40</strong></td>
<td>no shutdown</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>disable an interface.</td>
</tr>
<tr>
<td>switch(config-if)# no shutdown</td>
<td></td>
</tr>
<tr>
<td><strong>Step 41</strong></td>
<td>vrf member vrf100</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>configure VRF name.</td>
</tr>
<tr>
<td>switch(config-if)# vrf member vrf100</td>
<td></td>
</tr>
<tr>
<td><strong>Step 42</strong></td>
<td>ip forward</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>enable IP forwarding on interface.</td>
</tr>
<tr>
<td>switch(config-if)# ip forward</td>
<td></td>
</tr>
<tr>
<td><strong>Step 43</strong></td>
<td>ip pim sparse-mode</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>configure sparse-mode PIM on interface.</td>
</tr>
<tr>
<td>switch(config-if)# ip pim sparse-mode</td>
<td></td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>configure terminal</td>
<td>Enter configuration mode.</td>
</tr>
<tr>
<td></td>
<td>switch# configure terminal</td>
<td>Examples:</td>
</tr>
<tr>
<td>2</td>
<td>feature ngmvpn</td>
<td>Enables EVPN/MVPN feature.</td>
</tr>
<tr>
<td></td>
<td>switch(config)# feature ngmvpn</td>
<td>Examples:</td>
</tr>
<tr>
<td>3</td>
<td>advertise evpn multicast</td>
<td>Advertise L2 multicast capability.</td>
</tr>
<tr>
<td></td>
<td>switch(config)# advertise evpn multicast</td>
<td>Examples:</td>
</tr>
<tr>
<td>4</td>
<td>ip igmp snooping vxlan</td>
<td>Configure IGMP snooping for VXLANs.</td>
</tr>
<tr>
<td></td>
<td>switch(config)# ip igmp snooping vxlan</td>
<td>Examples:</td>
</tr>
</tbody>
</table>
CHAPTER 10

Configuring VXLAN QoS

This chapter contains the following sections:

• Information About VXLAN QoS, on page 197
• Licensing Requirements for VXLAN QoS, on page 205
• Guidelines and Limitations for VXLAN QoS, on page 205
• Default Settings for VXLAN QoS, on page 206
• Configuring VXLAN QoS, on page 207
• Verifying the VXLAN QoS Configuration, on page 208
• VXLAN QoS Configuration Examples, on page 209

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

VXLAN QoS Terminology

This section defines VXLAN QoS terminology.
### VXLAN QoS Terminology

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frames</td>
<td>Carries traffic at Layer 2. Layer 2 frames carry Layer 3 packets.</td>
</tr>
<tr>
<td>Packets</td>
<td>Carries traffic at Layer 3.</td>
</tr>
<tr>
<td>VXLAN packet</td>
<td>Carries original frame, encapsulated in VXLAN IP/UDP header.</td>
</tr>
<tr>
<td>Original frame</td>
<td>A Layer 2 or Layer 2 frame that carries the Layer 3 packet before encapsulation in a VXLAN header.</td>
</tr>
<tr>
<td>Decapsulated frame</td>
<td>A Layer 2 or a Layer 2 frame that carries a Layer 3 packet after the VXLAN header is decapsulated.</td>
</tr>
<tr>
<td>Ingress VTEP</td>
<td>The point where traffic is encapsulated in the VXLAN header and enters the VXLAN tunnel.</td>
</tr>
<tr>
<td>Egress VTEP</td>
<td>The point where traffic is decapsulated from the VXLAN header and exits the VXLAN tunnel.</td>
</tr>
<tr>
<td>Class of Service (CoS)</td>
<td>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.</td>
</tr>
<tr>
<td>IP precedence</td>
<td>The 3 most significant bits of the ToS byte in the IP header.</td>
</tr>
<tr>
<td>Differentiated Services Code Point (DSCP)</td>
<td>The first six bits of the ToS byte in the IP header. DSCP is only present in an IP packet.</td>
</tr>
<tr>
<td>Explicit Congestion Notification (ECN)</td>
<td>The last two bits of the ToS byte in the IP header. ECN is only present in an IP packet.</td>
</tr>
<tr>
<td>QoS tags</td>
<td>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.</td>
</tr>
<tr>
<td>Term</td>
<td>Definition</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Classification</td>
<td>The process used for selecting traffic for QoS</td>
</tr>
<tr>
<td>Marking</td>
<td>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.</td>
</tr>
<tr>
<td>Policing</td>
<td>Limiting bandwidth used by a flow of traffic. Policing can mark or drop traffic.</td>
</tr>
<tr>
<td>MQC</td>
<td>The Cisco Modular QoS command line interface (MQC) framework, which is a modular and highly extensible framework for deploying QoS.</td>
</tr>
</tbody>
</table>

**VXLAN QoS Features**

**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 polices monitor the specified committed information rate (CIR) of traffic. Dual-rate polices 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 policy dictates that the rate of a given interface should 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 packet or 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.

**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 then perform action 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 37: VXLAN Network](image)

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.

**Ingress and egress VTEPs are the boundary between the VXLAN tunnel and the IP network.**

**Ingress VTEP and Encapsulation in the VXLAN Tunnel**

At the ingress VTEP, the VTEP processes packets as follows:

**Procedure**

1. **Step 1**
   - Layer 2 or Layer 3 traffic enters the edge of the VXLAN network.
2. **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:

Procedure

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 process packets as follows:

Procedure

Step 1  Packets encapsulated in VXLAN packets 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 a packet, and does a lookup that is based on the decapsulated packet 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, for Layer-3 packet's DSCP value is copied from original header to the outer header of the VXLAN encapsulated packet. This behavior is illustrated in the following figure:

*Figure 38: 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 206. In this way, the original QoS attributes are preserved in the VXLAN tunnel. This behavior is illustrated in the following figure:

*Figure 39: 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.
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.

- 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.
Licensing Requirements for VXLAN QoS

The following table shows the licensing requirements for VXLAN QoS:

<table>
<thead>
<tr>
<th>Product</th>
<th>License Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco NX-OS</td>
<td>The Essentials license is required.</td>
</tr>
<tr>
<td></td>
<td>For a complete explanation of the Cisco NX-OS licensing scheme and how to obtain</td>
</tr>
<tr>
<td></td>
<td>and apply licenses, see the Cisco NX-OS Licensing Guide.</td>
</tr>
</tbody>
</table>

Guidelines and Limitations for VXLAN QoS

Note

QoS policy must be configured end-to-end for this feature to work as designed.
VXLAN QoS has the following configuration guidelines and limitations:

- Beginning with Cisco NX-OS Release 7.0(3)I7(5), support is added for VXLAN QoS.
- VXLAN QoS is not supported on Cisco Nexus 9200 platform switches, Cisco Nexus 9300 platform switches with 9400, 9500, or 9600 line cards.
- This feature 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.
- Entering the `policy-map type qos` command in the output direction for egress policing is not supported in the ingress VTEP.
- If 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 9300-EX platform switches and the Cisco Nexus 9364C switch.
- 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.

### Default Settings for VXLAN QoS

The following table lists the default CoS to DSCP mapping in the ingress VTEP for Layer 2 frames:

<table>
<thead>
<tr>
<th>CoS of Original Layer 2 Frame</th>
<th>DSCP of Outer VXLAN Header</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>3</td>
<td>26</td>
</tr>
<tr>
<td>4</td>
<td>32</td>
</tr>
<tr>
<td>5</td>
<td>46</td>
</tr>
<tr>
<td>6</td>
<td>48</td>
</tr>
<tr>
<td>7</td>
<td>56</td>
</tr>
</tbody>
</table>
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.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 VLXAN traffic, the service policy needs to 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.x.

VLXAN 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 VLXAN 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.

Before you begin

Procedure

<table>
<thead>
<tr>
<th>Step 1</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>switch# configure terminal</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 2</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>[no] class-map [type qos] [match-all]</td>
<td>Creates or accesses the class map class-map-name and enters class-map mode. The class-map-name can contain alphanumeric, 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.)</td>
</tr>
<tr>
<td>Example:</td>
<td>switch(config)# class-map type qos class1</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 3</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>[no] match [access-group</td>
<td>cos</td>
</tr>
<tr>
<td>Example:</td>
<td>switch(config-cmap-qos)# match dscp 26</td>
<td></td>
</tr>
</tbody>
</table>
Verifying the VXLAN QoS Configuration

### Command or Action

| Step 4 | [no] policy-map type qos policy-map-name  
Example: switch(config-cmap-qos)# policy-map type qos policy | Purpose |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Creates or accesses the policy map 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.</td>
<td></td>
</tr>
</tbody>
</table>
| Step 5 | [no] class class-name  
Example: switch(config-pmap-qos)# class class1 | Purpose |
|        | 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 | Purpose |
|        | 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. |         |
| Step 7 | exit  
Example: switch(config-pmap-c-qos)# exit | Purpose |
|        | Exits class-map mode. |         |
| Step 8 | [no] interface nve nve-interface-number  
Example: switch(config)# interface nve 1 | Purpose |
|        | 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 policy1 | Purpose |
|        | 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 | Purpose |
|        | 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

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>show class map</td>
<td>Displays information about all configured class maps.</td>
</tr>
<tr>
<td>Command</td>
<td>Purpose</td>
</tr>
<tr>
<td>--------------------------</td>
<td>--------------------------------------------------------------</td>
</tr>
<tr>
<td>show policy-map</td>
<td>Displays information about all configured policy maps.</td>
</tr>
<tr>
<td>show running ipqos</td>
<td>Displays configured QoS configuration on the switch.</td>
</tr>
</tbody>
</table>

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.

```plaintext
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.

```plaintext
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 queueing 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
VXLAN Bud Node Over VPC Overview

Figure 43: Underlay Network Based on PIM-SM and OSPF

- VXLAN Bud Node Over VPC Overview, on page 211
- VXLAN Bud Node Over vPC Topology Example, on page 212
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:

```plaintext
Interface Vlan2
ip proxy-arp
```

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 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:

  ```plaintext
  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.

  ```plaintext
  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.

  ```plaintext
  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 redirect
  ip proxy-arp
  ip address 10.200.1.252/24
  no ipv6 redirect
  ip route ospf 1 area 0.0.0.0
  ip pim sparse-mode
  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 Peer Link.

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

• Backup SVI over Peer Link 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
    ip igmp static-oif route-map match-mcast-groups

  • 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
    ip igmp static-oif 225.1.1.1

• vPC interface configuration that carries the infra VLAN
interface port-channel1
  switchport mode trunk
  switchport trunk allowed vlan 2
  vpc 1

• Peer Link configuration

interface port-channel100
  switchport mode trunk
  spanning-tree port type network
  vpc peer-link

• NVE configuration

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

• 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

<table>
<thead>
<tr>
<th>Interface</th>
<th>VNI</th>
<th>Multicast-group</th>
<th>State</th>
<th>Mode</th>
<th>Type</th>
<th>[BD/VRF]</th>
<th>Flags</th>
</tr>
</thead>
<tbody>
<tr>
<td>nve1</td>
<td>10001</td>
<td>225.1.1.1</td>
<td>Up</td>
<td>DP</td>
<td>L2</td>
<td>[101]</td>
<td></td>
</tr>
<tr>
<td>nve1</td>
<td>10002</td>
<td>225.1.1.1</td>
<td>Up</td>
<td>DP</td>
<td>L2</td>
<td>[102]</td>
<td></td>
</tr>
<tr>
<td>nve1</td>
<td>10003</td>
<td>225.1.1.1</td>
<td>Up</td>
<td>DP</td>
<td>L2</td>
<td>[103]</td>
<td></td>
</tr>
</tbody>
</table>

tor1# sh nve peers

<table>
<thead>
<tr>
<th>Interface</th>
<th>Peer-IP</th>
<th>State</th>
<th>LearnType</th>
<th>Uptime</th>
<th>Router-Mac</th>
</tr>
</thead>
<tbody>
<tr>
<td>nve1</td>
<td>10.200.1.1</td>
<td>Up</td>
<td></td>
<td>00:07:23 n/a</td>
<td></td>
</tr>
<tr>
<td>nve1</td>
<td>10.200.1.2</td>
<td>Up</td>
<td></td>
<td>00:07:18 n/a</td>
<td></td>
</tr>
<tr>
<td>nve1</td>
<td>102.102.102.102</td>
<td>Up</td>
<td></td>
<td>00:07:23 n/a</td>
<td></td>
</tr>
</tbody>
</table>

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: 00:07:40, nve
(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

<table>
<thead>
<tr>
<th>vPC domain id</th>
<th>: 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peer status</td>
<td>: peer adjacency formed ok</td>
</tr>
<tr>
<td>vPC keep-alive status</td>
<td>: peer is alive</td>
</tr>
<tr>
<td>Configuration consistency status</td>
<td>: success</td>
</tr>
<tr>
<td>Per-vlan consistency status</td>
<td>: success</td>
</tr>
<tr>
<td>Type-2 consistency status</td>
<td>: success</td>
</tr>
<tr>
<td>vPC role</td>
<td>: secondary, operational primary</td>
</tr>
<tr>
<td>Number of vPCs configured</td>
<td>: 4</td>
</tr>
<tr>
<td>Peer Gateway</td>
<td>: Enabled</td>
</tr>
<tr>
<td>Dual-active excluded VLANS</td>
<td>: -</td>
</tr>
<tr>
<td>Graceful Consistency Check</td>
<td>: Enabled</td>
</tr>
<tr>
<td>Auto-recovery status</td>
<td>: Disabled</td>
</tr>
<tr>
<td>Delay-restore status</td>
<td>: Timer is off.(timeout = 180s)</td>
</tr>
<tr>
<td>Delay-restore SVI status</td>
<td>: Timer is off.(timeout = 10s)</td>
</tr>
</tbody>
</table>

vPC Peer-link status

<table>
<thead>
<tr>
<th>id</th>
<th>Port</th>
<th>Status</th>
<th>Active vlans</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Po100</td>
<td>up</td>
<td>1-4,101-103,2000</td>
</tr>
</tbody>
</table>

vPC status

<table>
<thead>
<tr>
<th>id</th>
<th>Port</th>
<th>Status</th>
<th>Consistency</th>
<th>Reason</th>
<th>Active vlans</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Po1</td>
<td>up</td>
<td>success</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>Po2</td>
<td>up</td>
<td>success</td>
<td></td>
<td>2</td>
</tr>
</tbody>
</table>
tor1# sh vpc consistency-parameters global

Legend:
Type 1: vPC will be suspended in case of mismatch

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Local Value</th>
<th>Peer Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vlan to Vn-segment Map</td>
<td>1</td>
<td>3 Relevant Map(s)</td>
<td>3 Relevant Map(s)</td>
</tr>
<tr>
<td>STP Mode</td>
<td>1</td>
<td>Rapid-PVST</td>
<td>Rapid-PVST</td>
</tr>
<tr>
<td>STP Disabled</td>
<td>1</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>STP MST Region Name</td>
<td>1</td>
<td>&quot;&quot;</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>STP MST Region Revision</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>STP MST Region Instance to VLAN Mapping</td>
<td>1</td>
<td>Disabled</td>
<td>Disabled</td>
</tr>
<tr>
<td>STP Loopguard</td>
<td>1</td>
<td>Disabled</td>
<td>Disabled</td>
</tr>
<tr>
<td>STP Bridge Assurance</td>
<td>1</td>
<td>Enabled</td>
<td>Enabled</td>
</tr>
<tr>
<td>STP Port Type, Edge</td>
<td>1</td>
<td>Normal, Disabled,</td>
<td>Normal, Disabled,</td>
</tr>
<tr>
<td>BPDUFilter, Edge BPDUGuard</td>
<td>1</td>
<td>Disabled</td>
<td>Disabled</td>
</tr>
<tr>
<td>STP MST Simulate PVST</td>
<td>1</td>
<td>Enabled</td>
<td>Enabled</td>
</tr>
<tr>
<td>IP, Host Reach Mode</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nve Vni Configuration</td>
<td>1</td>
<td>10001-10003</td>
<td>10001-10003</td>
</tr>
<tr>
<td>Interface-vlan admin up</td>
<td>2</td>
<td>2,2000</td>
<td>2,2000</td>
</tr>
<tr>
<td>Interface-vlan routing capability</td>
<td>2</td>
<td>1-4,2000</td>
<td>1-4,2000</td>
</tr>
<tr>
<td>Allowed VLANs</td>
<td>-</td>
<td>1-4,101-103,2000</td>
<td>1-4,101-103,2000</td>
</tr>
<tr>
<td>Local suspended VLANs</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>
DHCP Relay in VXLAN BGP EVPN

DHCP relay is supported by VXLAN BGP EVPN and is useful in a multi-tenant VXLAN EVPN deployment to provision DHCP service to EVPN tenant clients.

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 44: Example Topology**

- **ASR1K-DHCP**
  - 192.1.42.3
- **VLAN10**
- **9372-1**
  - E1/50
  - E1/49
- **VXLAN EVPN**
- **9372-2**
  - E1/50
- **PC-1**
  - DHCP client-1:
  - VLAN1001
  - MAC:0000.1001.5151
  - Interface Vlan1001
  - no shutdown
  - vrf member vxlan-900001
  - ip address 172.16.16.120
  - fabric forwarding mode anycast-gateway
- **PC-2**
  - DHCP client-2:
  - VLAN1001
  - MAC:0000.1001.7575
  - Interface Vlan1001
  - no shutdown
  - vrf member vxlan-900001
  - ip address 172.16.16.130
  - fabric forwarding mode anycast-gateway

**Topology characteristics:**

- Switches 9372-1 and 9372-2 are VTEPs connected to 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

```bash
ip vrf vxlan900001
ip dhcp excluded-address vrf vxlan900001 172.16.16.1 172.16.16.9
ip dhcp pool one
  vrf vxlan900001
  network 172.16.16.0 255.255.240.0
default-router 172.16.16.1
```
Basic VXLAN BGP EVPN Configuration

- 9372-1

```plaintext
version 7.0(3)I1(3)
hostname 9372-1

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

fabric forwarding anycast-gateway-mac 0000.1111.2222

vlan 101
   vn-segment 900001
vlan 1001
   vn-segment 2001001

vrf context vxlan-900001
   vni 900001
   rd auto
   address-family ipv4 unicast
       route-target both auto
       route-target both auto evpn

interface Vlan101
   no shutdown
   vrf member vxlan-900001
   ip forward

interface Vlan1001
   no shutdown
   vrf member vxlan-900001
   ip address 172.16.16.1/20
   fabric forwarding mode anycast-gateway

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

interface Ethernet1/49
   switchport mode trunk
   switchport trunk allowed vlan 10,1001
   spanning-tree port type edge trunk

interface Ethernet1/50
```
no switchport
ip address 192.1.33.2/24
ip router ospf 1 area 0.0.0.0
ip pim sparse-mode
no shutdown

interface loopback0
ip address 1.1.1.1/32
ip router ospf 1 area 0.0.0.0
ip pim sparse-mode

interface loopback1
vrf member vxlan-900001
ip address 11.11.11.11/32

router bgp 65535
router-id 1.1.1.1
log-neighbor-changes
neighbor 2.2.2.2 remote-as 65535
update-source loopback0
address-family l2vpn evpn
send-community both
vrf vxlan-900001
address-family ipv4 unicast
network 11.11.11.11/32
network 192.1.42.0/24
advertise l2vpn evpn
evpn
vni 2001001 12

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

• 9372-2

version 7.0(3)I1(3)
hostname 9372-1

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

fabric forwarding anycast-gateway-mac 0000.1111.2222

vlan 101
vn-segment 900001
vlan 1001
vn-segment 2001001

vrf context vxlan-900001
vni 900001
rd auto
address-family ipv4 unicast
route-target both auto
route-target both auto evpn

interface Vian101
no shutdown
vrf member vxlan-900001
ip forward
interface Vlan1001
  no shutdown
  vrf member vxlan-900001
  ip address 172.16.16.1/20
  fabric forwarding mode anycast-gateway
  rd auto
    address-family ipv4 unicast
      route-target both auto
      route-target both auto evpn

interface Vlan1001
  no shutdown
  vrf member vxlan-900001
  ip forward

interface Vlan1001
  no shutdown
  vrf member vxlan-900001
  ip address 172.16.16.1/20
  fabric forwarding mode anycast-gateway

interface nve1
  no shutdown
  source-interface loopback1
  host-reachability protocol bgp
  member vni 10000 associate-vrf
  mcast-group 224.1.1.1
  member vni 10001 associate-vrf
  mcast-group 224.1.1.1
  member vni120000
  suppress-arp
  mcast-group 225.1.1.1
  member vni 20001
  suppress-arp
  mcast-group 225.1.1.1

interface Ethernet1/49
  switchport mode trunk
  switchport trunk allowed vlan 10,1001
  spanning-tree port type edge trunk

interface Ethernet1/50
  no switchport
  ip address 192.1.34.2/24
  ip router ospf 1 area 0.0.0.0
  ip pim sparse-mode
  no shutdown

interface loopback0
  ip address 2.2.2.2/32
  ip router ospf 1 area 0.0.0.0
  ip pim sparse-mode

interface loopback1
  vrf member vxlan-900001
  ip address 22.22.22.22/32

router bgp 65535
  router-id 2.2.2.2
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 (192.1.42.3) 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
version 7.0(3)I1(3)
interface Vlan10
  no shutdown
  ip address 192.1.42.1/24
  ip router ospf 1 area 0.0.0.0
9372-1# ping 192.1.42.3 cou 1
PING 192.1.42.3 (192.1.42.3): 56 data bytes
64 bytes from 192.1.42.3: icmp_seq=0 ttl=254 time=0.593 ms
- 192.1.42.3 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 192.1.42.3 cou 1
```
PING 192.1.42.3 (192.1.42.3): 56 data bytes
64 bytes from 192.1.42.3: icmp_seq=0 ttl=252 time=0.609 ms
- 192.1.42.3 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 2015

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 Vlan1001
  ip dhcp relay address 192.1.42.3 use-vrf default

- 9372-2

9372-2# sh run dhcp

!Command: show running-config dhcp
!Time: Mon Aug 24 08:26:16 2015

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 Vlan1001
  ip dhcp relay address 192.1.42.3 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
20150824 08:35:25.066530 0.0.0.0 -> 255.255.255.255 DHCP DHCP Discover - Transaction
ID 0x636a38fd
20150824 08:35:25.068141 192.1.42.1 -> 192.1.42.3 DHCP DHCP Discover - Transaction ID 0x636a38fd
20150824 08:35:27.069494 192.1.42.3 -> 192.1.42.1 DHCP DHCP Offer Transaction - ID 0x636a38fd
20150824 08:35:27.071029 172.16.16.1 -> 172.16.16.11 DHCP DHCP Offer Transaction - ID 0x636a38fd
20150824 08:35:27.071488 0.0.0.0 -> 255.255.255.255 DHCP DHCP Request Transaction - ID 0x636a38fd
20150824 08:35:27.072447 192.1.42.1 -> 192.1.42.3 DHCP DHCP Request Transaction - ID 0x636a38fd
20150824 08:35:27.073008 192.1.42.3 -> 192.1.42.1 DHCP DHCP ACK Transaction - ID 0x636a38fd
20150824 08:35:27.073692 172.16.16.1 -> 172.16.16.11 DHCP DHCP ACK Transaction - ID 0x636a38fd

Ethalyzer 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 192.1.42.1 (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: 192.1.42.1 (192.1.42.1)
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
ASR1K-DHCP# sh ip dhcp bin
Bindings from all pools not associated with VRF:
<table>
<thead>
<tr>
<th>IP address</th>
<th>ClientID/ Lease expiration</th>
<th>Type</th>
<th>State</th>
<th>Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>172.16.16.10</td>
<td>0100.0010.0175.75Aug 25 2015 09:21 AM</td>
<td>Automatic</td>
<td>Active</td>
<td>GigabitEthernet2/1/0</td>
</tr>
<tr>
<td>172.16.16.11</td>
<td>0100.0010.0151.51Aug 25 2015 08:54 AM</td>
<td>Automatic</td>
<td>Active</td>
<td>GigabitEthernet2/1/0</td>
</tr>
</tbody>
</table>

ASR1K-DHCP# 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>

11.11.11.11/32, ubest/mbest: 2/0, attached
  *via 11.11.11.11, Lo1, [0/0], 18:31:57, local
  *via 11.11.11.11, Lo1, [0/0], 18:31:57, direct
22.22.22/32, ubest/mbest: 1/0
  *via 2.2.2.2%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 2.2.2.2%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
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

Length: 4
Link selection: 172.16.16.0 (172.16.16.0)
Client on Tenant VRF (SVI X) and Server on the Same Tenant VRF (SVI Y)

Put DHCP server (192.1.42.3) 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 2015
version 7.0(3)I1(3)
interface Vlan10
  no shutdown
  vrf member vxlan-900001
  ip address 192.1.42.1/24

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 loopback 1 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
version 7.0(3)I1(3)
interface loopback1
  vrf member vxlan-900001
  ip address 11.11.11.11/32

9372-1# ping 192.1.42.3 vrf vxlan900001 source 11.11.11.11 cou 1
PING 192.1.42.3 (192.1.42.3) from 11.11.11.11: 56 data bytes
64 bytes from 192.1.42.3: icmp_seq=0 ttl=254 time=0.575 ms
- 192.1.42.3 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 2015
version 7.0(3)I1(3)
interface loopback1
  vrf member vxlan900001
  ip address 22.22.22.22/32

9372-2# ping 192.1.42.3 vrf vxlan-900001 source 22.22.22.22 cou 1
PING 192.1.42.3 (192.1.42.3) from 22.22.22.22: 56 data bytes
64 bytes from 192.1.42.3: icmp_seq=0 ttl=253 time=0.662 ms
- 192.1.42.3 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 2015

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 Vlan1001
  ip dhcp relay address 192.1.42.3
  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 2015

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 Vlan1001
  ip dhcp relay address 192.1.42.3
  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
20150824 09:31:38.129393 0.0.0.0 -> 255.255.255.255 DHCP DHCP Discover - Transaction ID 0x860cd13
20150824 09:31:40.130134 192.1.42.3 -> 11.11.11.11 DHCP DHCP Offer - Transaction ID 0x860cd13
20150824 09:31:40.130552 172.16.16.1 -> 172.16.16.11 DHCP DHCP Offer - Transaction ID 0x860cd13
```

Cisco Nexus 9000 Series NX-OS VXLAN Configuration Guide, Release 7.x
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 11.11.11.11 (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: 0
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: 11.11.11.11 (11.11.11.11)
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) Agent 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)
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.
DHCP Relay in VXLAN BGP EVPN

version 7.0(3)I1(3)
interface Vlan10
   no shutdown
   vrf member vxlan900002
   ip address 192.1.42.1/24

9372-1# sh run int lo2
!Command: show running-config interface loopback2
version 7.0(3)I1(3)
interface loopback2
   vrf member vxlan900002
   ip address 33.33.33.33/32

9372-2# sh run int lo2
!Command: show running-config interface loopback2
version 7.0(3)I1(3)
interface loopback2
   vrf member vxlan900002
   ip address 44.44.44.44/32

9372-1# ping 192.1.42.3 vrf vxlan-900002 source 33.33.33.33 count 1
PING 192.1.42.3 (192.1.42.3) from 33.33.33.33: 56 data bytes
64 bytes from 192.1.42.3: icmp_seq=0 ttl=254 time=0.544 ms
- 192.1.42.3 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 192.1.42.3 vrf vxlan-900002 source 44.44.44.44 count 1
PING 192.1.42.3 (192.1.42.3) from 44.44.44.44: 56 data bytes
64 bytes from 192.1.42.3: icmp_seq=0 ttl=253 time=0.678 ms
- 192.1.42.3 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 2015

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 Vlan1001
   ip dhcp relay address 192.1.42.3 use=vrf vxlan-900002
   ip dhcp relay source-interface loopback2

- 9372-2
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 192.1.42.3 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
20150825 08:59:35.758314 0.0.0.0 -> 255.255.255.255 DHCP DHCP Discover - Transaction ID 0x3eebccae
20150825 08:59:35.758878 33.33.33.33 -> 192.1.42.3 DHCP DHCP Discover - Transaction ID 0x3eebccae
20150825 08:59:37.759560 192.1.42.3 -> 33.33.33.33 DHCP DHCP Offer - Transaction ID 0x3eebccae
20150825 08:59:37.759905 172.16.16.1 -> 172.16.16.11 DHCP DHCP Offer - Transaction ID 0x3eebccae
20150825 08:59:37.760313 0.0.0.0 -> 255.255.255.255 DHCP DHCP Request - Transaction ID 0x3eebccae
20150825 08:59:37.760733 33.33.33.33 -> 192.1.42.3 DHCP DHCP Request - Transaction ID 0x3eebccae
20150825 08:59:37.761297 192.1.42.3 -> 33.33.33.33 DHCP DHCP ACK - Transaction ID 0x3eebccae
20150825 08:59:37.761554 172.16.16.11 -> 172.16.16.1 DHCP DHCP ACK - Transaction ID 0x3eebccae

• DHCP Discover packet 9372-1 sent to DHCP server.
  giaddr is set to 33.33.33.33 (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: 0x3eebccae
  Seconds elapsed: 0
  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: 33.33.33.33 (33.33.33.33)
  Client MAC address: i-lughes_01:51:51 (00:00:10:01:51:51)
Client on Tenant VRF and Server on Non-Default Non-VXLAN VRF

The DHCP server is placed into the management VRF and is reachable 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 2015
version 7.0(3)I1(3)
interface mgmt0
  vrf member management
  ip address 10.122.165.134/25
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
64 bytes from 10.122.164.147: icmp_seq=0 ttl=251 time=1.024 ms
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 2015
version 7.0(3)I1(3)
interface mgmt0
  vrf member management
  ip address 10.122.165.148/25
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
64 bytes from 10.122.164.147: icmp_seq=0 ttl=251 time=1.03 ms
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
Cisco Nexus 9000 Series NX-OS VXLAN Configuration Guide, Release 7.x

DHCP Relay in VXLAN BGP EVPN

9372-1# sh run dhcp 9372-2# sh run dhcp

!Command: show running-config dhcp
!Time: Mon Aug 24 08:26:00 2015

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 Vlan1001
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 2015

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 Vlan1001
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
20150825 09:30:54.214998 0.0.0.0 -> 255.255.255.255 DHCP DHCP Discover - Transaction ID 0x28a8606d
20150825 09:30:56.216491 172.16.16.1 -> 172.16.16.11 DHCP DHCP Offer - Transaction ID 0x28a8606d
20150825 09:30:56.216931 0.0.0.0 -> 255.255.255.255 DHCP DHCP Request - Transaction ID 0x28a8606d
20150825 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
20150825 09:30:54.215499 10.122.165.134 -> 10.122.164.147 DHCP DHCP Discover - Transaction ID 0x28a8606d
20150825 09:30:56.215499 10.122.165.134 -> 10.122.164.147 DHCP DHCP Discover - Transaction ID 0x28a8606d
20150825 09:30:56.215499 10.122.165.134 -> 10.122.164.147 DHCP DHCP Discover - 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: 0
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 160.1.0.43/32
   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 10.1.1.42/32
```
• Configure DHCP relay on the SVI under the VRF.

```plaintext
interface Vlan1601
  vrf member evpn-tenant-kl1
  ip address 10.160.1.254/24
  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`.

```plaintext
interface Vlan1600
  vrf member evpn-tenant-kl1
  ip forward
```

• Create the routing VLAN/SVI for the VPC VRF.

```plaintext
Vlan 1605
interface Vlan1605
  vrf member evpn-tenant-kl1
  ip address 10.160.5.43/24
  ip router ospf 1 area 0.0.0.41
```

• Create the VRF routing.

```plaintext
router ospf 1
  vrf evpn-tenant-kl1
  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 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 Peer Link. 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 192.168.1.1/30
   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 192.168.1.2/30
   ip router ospf UNDERLAY area 0.0.0.0

• Configuration of SVI using static routes between vPC peers across the Peer Link:

/* vPC Peer-1 */

interface Vlan2000
   no shutdown
   mtu 9216
   vrf member tenant-vrf
   ip address 192.168.1.1/30

vrf context tenant-vrf
ip route 192.168.1.2/30 192.168.1.1

/* vPC Peer-2 */

interface Vlan2000
   no shutdown
   mtu 9216
   vrf member tenant-vrf
   ip address 192.168.1.2/30

vrf context tenant-vrf
ip route 192.168.1.1/30 192.168.1.2
EVPN with Transparent Firewall Insertion

In certain scenarios there is a requirement to send all routing traffic through a Layer 2 transparent firewall. However, by default, VXLAN EVPN requires a distributed anycast gateway on all LEAFs.

To address the Layer 2 transparent firewall requirement with VXLAN EVPN, a special topology can be used.
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 a HSRP gateway.
VXLAN flood and learn only supports a centralized gateway. This means that only one VPC pair VTEP can have an SVI per VXLAN. No other VTEP can have an SVI on a VXLAN VLAN. VXLAN only supports an anycast gateway, not a centralized gateway.

EVPN with Transparent Firewall Insertion Example

- Host in VLAN X: 10.0.94.101
- 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 12
    rd auto
    route-target import auto
    route-target export auto

• Service leaf 1 configuration

vlan 94
  description untrusted_vlan
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
  router-id 1.1.2.1
  neighbor 1.1.1.1 remote-as 64500
  address-family 12vpn evpn
    send-community extended
  neighbor 1.1.1.2 remote-as 64500
  address-family 12vpn 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 12vpn evpn
evpn
  vni 100094 12
  rd auto
  route-target import auto
  route-target export auto

• Service leaf 2 configuration

  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 12
rd auto
route-target import auto
route-target export auto

Show Command Examples

• Display information about ingress LEAF learned local MAC from host:

N93721# 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 service leaf found MAC of host:

Note
In VLAN 94, the service leaf learned the host MAC from the remote peer by BGP.

N93321# 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)

N93322# 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)

N93321# 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

N93322# 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:

N93322# 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.
N93322# sh ip route vrf ten-1 9.9.9.9

IP Route Table for VRF "Ten-1"
'*' denotes best unicast nexthop
'**' denotes best mcast nexthop
'[/x/y]' denotes [preference/metric]
'%' 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: VXLA

* Display information about border leaf learned host routes by BGP:

N93965# sh ip route 10.0.94.101

IP Route Table for VRF "default"
'*' denotes best unicast nexthop
'**' denotes best mcast nexthop
'[/x/y]' denotes [preference/metric]
'%' 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
Overview of IPv6 Across a VXLAN EVPN Fabric

This section provides an example configuration that enables IPv6 in the overlay of a VXLAN EVPN fabric. The VXLAN encapsulation mechanism encapsulates the IPv6 packets in the overlay as IPv4 UDP packets and uses IPv4 routing to transport the VXLAN encapsulated traffic.

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

Note

These routes are advertised as EVPN routes on the SPINE.

Configuring IPv6 Across a VXLAN EVPN Fabric Example

Topology for the example:
In the example:

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

- Configure the Layer 2 VLAN.

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

- Configure the VLAN for L3 VNI.

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

- Define the anycast gateway MAC.

```
fabric forwarding anycast-gateway-mac 0000.2222.3333
```
• Define the NVE interface.

interface nve1
   no shutdown
   source-interface loopback1
   host-reachability protocol bgp
   member vni 10000 associate-vrf
   mcast-group 224.1.1.1
   member vni 10001 associate-vrf
   mcast-group 224.1.1.1
   member vni 20000
   mcast-group 225.1.1.1
   member vni 20001
   mcast-group 225.1.1.1

   evpn
   vni 10010 12

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

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

interface Vlan10
   description RED
   no shutdown
   vrf member RED
   no ip redirects
   ip address 10.1.1.1/24
   ipv6 address 2001::1/64
   fabric forwarding mode anycast-gateway

   Note
   IPv6 ND suppression is not supported on Cisco Nexus 9000 Series switches.

• Configure SVI definition for VLAN 100.

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

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

• Add configuration to the VRF definition.
vrf context RED
  vni 20010

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

evpn
  vni 10010 12

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

- Add configuration to the VRF definition under BGP.

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

---

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

  vpc domain 1
    ipv6 nd synchronize

---

Show Command Examples

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

- Display ND information for the connected server.

  9396-B_VTEP# show ipv6 neighbor vrf RED

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

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

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

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

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

Verify the route is present locally in the BGP table.

```
9396-B_VTEP# show bgp l2vpn evpn 2001::64
BGP routing table information for VRF default, address family L2VPN EVPN
Route Distinguisher: 198.19.0.15:34180 (L2VNI 10010)
BGP routing table entry for [2]:[0]:[0]:[48]:[7c69.f614.2bc1]:[128]:[2001::64]/368, version 678
Paths: (1 available, best #1)
Flags: (0x00010a) on xmit-list, is not in l2rib/evpn
  Advertised path-id 1
  Path type: local, path is valid, is best path, no labeled nexthop
  AS-Path: NONE, path locally originated
    198.19.0.15 (metric 0) from 0.0.0.0 (198.19.0.15)
    Origin IGP, MED not set, localpref 100, weight 32768
    Received label 10010 20010
  Path-id 1 advertised to peers:
    198.19.0.3
    198.19.0.4
```

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

```
9396-A-VTEP# show bgp l2vpn evpn 2001::64
BGP routing table information for VRF default, address family L2VPN EVPN
Route Distinguisher: 198.19.0.14:34180 (L2VNI 10010)
BGP routing table entry for [2]:[0]:[0]:[48]:[7c69.f614.2bc1]:[128]:[2001::64]/368, version 305
Paths: (1 available, best #1)
Flags: (0x00021a) on xmit-list, is in l2rib/evpn, is not in HW,
  Advertised path-id 1
  Path type: internal, path is valid, is best path, no labeled nexthop
  AS-Path: NONE, path sourced internal to AS
    198.19.0.15:34180:[2]:[0]:[0]:[48]:[7c69.f614.2bc1]:[128]:[2001::64]/240
    Origin IGP, MED not set, localpref 100, weight 0
    Received label 10010 20010
    Extcommunity: RT:64567:10010 RT:64567:20010 ENCAP:8 Router MAC:5087.89a1.a52f
    Originator: 198.19.0.15 Cluster list: 198.19.0.3
```

Check the L2ROUTE and ensure that the MAC-IP was learned on the remote VTEP - 9396-A-VTEP.
rswV1leaf14# show l2route evpn mac-ip evi 1413 host-ip 2001::64

<table>
<thead>
<tr>
<th>Mac Address</th>
<th>Prod Host IP</th>
<th>Next Hop (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>7c69.f614.2bc1</td>
<td>BGP 2001::64</td>
<td>198.19.0.15</td>
</tr>
</tbody>
</table>
# INDEX

## A
- Address-family ipv4 unicast 81, 85
- Address-family ipv6 unicast 81, 85
- Address-family l2vpn evpn 85, 88, 89
- Advertise 85
- Associate-vrf 78

## C
- Clear nve peer-ip 59, 60
- Clear nve peers 59, 60
- Clear nve vni 60

## E
- Evpn 85

## F
- Fabric forwarding 78
- Fabric forwarding anycast-gateway-mac 84
- Fabric forwarding mode anycast-gateway 84
- Feature nv overlay 34, 80
- Feature vn-segment 80
- Feature vn-segment-vlan-based 34
- Force 29

## H
- Hardware access-list tcam region arp-ether double-wide 20, 73, 89
- Host-reachability protocol bgp 78, 84, 87
- How interface 30

## I
- Ingress-replication protocol bgp 35, 87
- Ingress-replication protocol static 10, 35
- Interface 84
- Interface nve 33, 35
- Interface nve 1 89
- Ip address 83

## M
- Mac address-table static 33
- Mcast-group 33, 84
- Member vni 33, 35, 78, 84, 87, 89

## N
- Neighbor 84, 88
- No feature nv overlay 90
- No feature vn-segment-vlan-based 90
- No nv overlay evpn 90
- Nv overlay evpn 78, 80

## O
- Overlay-encapsulation vxlan-with-tag 38

## P
- Peer-ip 35

## R
- Rd auto 81, 85
- Retain route-target all 88
- Route-map permitall out 89
- Route-map permitall permit 10 88
- Route-target both auto 81
- Route-target both auto evpn 81, 82
- Route-target export auto 85
- Route-target import auto 85
- Router bgp 78, 84, 88
- Router-id 84

## S
- Send-community extended 85, 89
- Set ip next-hop unchanged 88
- Show bgp l2vpn evpn 78, 93, 112
- Show bgp l2vpn evpn summary 78, 112
- Show interface 32, 60
- Show ip arp suppression-cache 93
- Show ip arp suppression-cache detail 112
INDEX

show l2route evpn fl all  93
show l2route evpn imet all  93
show l2route evpn mac  93
show l2route evpn mac all  113
show l2route evpn mac-ip all  93, 113
show l2route evpn mac-ip all detail  93
show l2route topology  93
show logging level nve  59, 60
show mac address-table static interface nve  34
show mac address-table static interface nve 1  60
show nve interface  59, 60
show nve peers  59, 60, 111
show nve vni  59, 60, 78, 111
show nve vni ingress-replication  60
show nve vni summary  78
show nve vrf  93
show nve vxlan-params  60
show run interface nve  59, 60
show run track  61
show tech-support nve  59, 60
show tech-support vxlan  59, 60
show track  61
show vxlan interface  61, 93, 112
show vxlan interface | count  61, 93
source-interface  33, 35
source-interface config  19, 72
source-interface hold-down-time  20, 72
spanning-tree bpdufilter enable  38
suppress-arp  78, 90
suppress-mac-route  80
switchport access vlan  38
switchport mode dot1q-tunnel  37
switchport mode trunk  32
switchport vlan mapping  29, 32
switchport vlan mapping all  30
switchport vlan mapping enable  29, 32
system vlan nve-overlay id  79

V

vlan  26, 81, 83
vn-segment  26, 81, 83
vni  81, 83, 85
vrf  85
vrf context  78, 81, 83
vrf member  83