



Cisco Nexus 9000 Series NX-OS VXLAN Configuration Guide, Release 7.x

First Published: 2015-01-27

Last Modified: 2022-04-06

Americas Headquarters

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

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS REFERENCED IN THIS DOCUMENTATION ARE SUBJECT TO CHANGE WITHOUT NOTICE. EXCEPT AS MAY OTHERWISE BE AGREED BY CISCO IN WRITING, ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS DOCUMENTATION ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.

The Cisco End User License Agreement and any supplemental license terms govern your use of any Cisco software, including this product documentation, and are located at: <http://www.cisco.com/go/softwareterms>. Cisco product warranty information is available at <http://www.cisco.com/go/warranty>. US Federal Communications Commission Notices are found here <http://www.cisco.com/c/en/us/products/us-fcc-notice.html>.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any products and features described herein as in development or available at a future date remain in varying stages of development and will be offered on a when-and if-available basis. Any such product or feature roadmaps are subject to change at the sole discretion of Cisco and Cisco will have no liability for delay in the delivery or failure to deliver any products or feature roadmap items that may be set forth in this document.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

The documentation set for this product strives to use bias-free language. For the purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on RFP documentation, or language that is used by a referenced third-party product.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: [www.cisco.com go trademarks](http://www.cisco.com/go/trademarks). Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2016–2022 Cisco Systems, Inc. All rights reserved.



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
Licensing Requirements	7
Supported Platforms	7
VXLAN Overview	7
VXLAN Encapsulation and Packet Format	8
VXLAN Tunnel Endpoint	8
VXLAN Packet Forwarding Flow	9
Cisco Nexus 9000 as Hardware-Based VXLAN Gateway	9
vPC Consistency Check for vPC VTEPs	9
Static Ingress Replication	11
Bud Node Topology	11
VXLAN BGP EVPN Control Plane	12

CHAPTER 3

Configuring VXLAN	15
Information About VXLAN	15
Guidelines and Limitations for VXLAN	15

Considerations for VXLAN Deployment	22
vPC Considerations for VXLAN Deployment	24
Network Considerations for VXLAN Deployments	27
Considerations for the Transport Network	27
Considerations for Tunneling VXLAN	29
Configuring VXLAN	30
Enabling VXLANs	30
Mapping VLAN to VXLAN VNI	30
Guidelines and Limitations for Port VLAN Mapping	30
Configuring Port VLAN Mapping on a Trunk Port	31
Configuring Inner VLAN and Outer VLAN Mapping on a Trunk Port	35
Creating and Configuring an NVE Interface and Associate VNIs	36
Configuring Static MAC for VXLAN VTEP	37
Disabling VXLANs	38
Configuring BGP EVPN Ingress Replication	38
Configuring Static Ingress Replication	39
Guidelines and Limitations for Q-in-VNI	39
Configuring Q-in-VNI	40
Configuring Selective Q-in-VNI	43
Configuring Q-in-VNI with LACP Tunneling	45
Configuring QinQ-QinVNI	47
Overview for QinQ-QinVNI	47
Guidelines and Limitations for QinQ-QinVNI	48
Configuring QinQ-QinVNI	48
Removing a VNI	49
Configuring FHRP Over VXLAN	50
Overview for FHRP Over VXLAN	50
Guidelines and Limitations for FHRP Over VXLAN	50
Only Supported Deployments for FHRP Over VXLAN	51
New Supported Topology for Configuring FHRP Over VXLAN	52
Configuring IGMP Snooping Over VXLAN	54
Overview of IGMP Snooping Over VXLAN	54
Guidelines and Limitations for IGMP Snooping Over VXLAN	54
Configuring IGMP Snooping Over VXLAN	54

Configuring Line Cards for VXLAN	55
Centralized VRF Route Leaking using Default-Routes and Aggregates	56
Overview	56
Deploying EVPN	56
Reachability between Leaves	58
VPN to Default-VRF Reachability	58
Guidelines and Limitations	58
Configuration Examples for Centralized VRF Route Leak	59
VXLAN Tunnel Egress QoS Policy	61
About VXLAN Tunnel Egress QoS Policy	61
Guidelines and Limitations for VXLAN Tunnel Egress QoS Policy	61
Configuring VXLAN Tunnel Egress QoS Policy	62
Verifying the VXLAN Configuration	63
Example of VXLAN Bridging Configuration	65

CHAPTER 4

Configuring VXLAN BGP EVPN	73
Information About VXLAN BGP EVPN	73
Guidelines and Limitations for VXLAN BGP EVPN	73
Considerations for VXLAN BGP EVPN Deployment	76
vPC Considerations for VXLAN BGP EVPN Deployment	77
Network Considerations for VXLAN Deployments	80
Considerations for the Transport Network	81
Considerations for Tunneling VXLAN	81
BGP EVPN Considerations for VXLAN Deployment	82
Configuring VXLAN BGP EVPN	85
Enabling VXLAN	85
Configuring VLAN and VXLAN VNI	86
Configuring VRF for VXLAN Routing	86
About RD Auto	87
About Route-Target Auto	87
Configuring SVI for Hosts for VXLAN Routing	88
Configuring VRF Overlay VLAN for VXLAN Routing	88
Configuring VNI Under VRF for VXLAN Routing	88
Configuring Anycast Gateway for VXLAN Routing	89

Configuring the NVE Interface and VNIs	89
Configuring BGP on the VTEP	89
Configuring RD and Route Targets for VXLAN Bridging	90
About RD Auto	91
About Route-Target Auto	91
Configuring VXLAN EVPN Ingress Replication	92
Configuring BGP for EVPN on the Spine	93
Suppressing ARP	94
Disabling VXLANs	95
Duplicate Detection for IP and MAC Addresses	95
Enabling Nuage Controller Interoperability	97
Verifying the VXLAN BGP EVPN Configuration	98
Example of VXLAN BGP EVPN (EBGP)	99
Example of VXLAN BGP EVPN (IBGP)	108
Example Show Commands	117

CHAPTER 5
Configuring VXLAN OAM 121

VXLAN OAM Overview	121
Loopback (Ping) Message	122
Traceroute or Pathtrace Message	123
Configuring VXLAN OAM	125
Configuring NGOAM Profile	128
NGOAM Authentication	129

CHAPTER 6
Configuring VXLAN EVPN Multihoming 131

VXLAN EVPN Multihoming Overview	131
Introduction to Multihoming	131
BGP EVPN Multihoming	131
BGP EVPN Multihoming Terminology	131
EVPN Multihoming Implementation	132
EVPN Multihoming Redundancy Group	133
Ethernet Segment Identifier	133
LACP Bundling	133
Guidelines and Limitations for VXLAN EVPN Multihoming	134

Configuring VXLAN EVPN Multihoming	135
Enabling EVPN Multihoming	135
VXLAN EVPN Multihoming Configuration Examples	136
Configuring Layer 2 Gateway STP	137
Layer 2 Gateway STP Overview	137
Guidelines for Moving to Layer 2 Gateway STP	138
Enabling Layer 2 Gateway STP on a Switch	139
Configuring VXLAN EVPN Multihoming Traffic Flows	141
EVPN Multihoming Local Traffic Flows	141
EVPN Multihoming Remote Traffic Flows	146
EVPN Multihoming BUM Flows	150
Configuring VLAN Consistency Checking	153
Overview of VLAN Consistency Checking	153
VLAN Consistency Checking Guidelines and Limitations	154
Configuring VLAN Consistency Checking	154
Displaying Show command Output for VLAN Consistency Checking	155
Configuring ESI ARP Suppression	156
Overview of ESI ARP Suppression	156
Limitations for ESI ARP Suppression	156
Configuring ESI ARP Suppression	156
Displaying Show Commands for ESI ARP Suppression	157

CHAPTER 7

Configuring VIP/PIP	159
Advertising Primary IP Address	159
BorderPE Switches in a vPC Setup	160
DHCP Configuration in a vPC Setup	160
IP Prefix Advertisement in vPC Setup	160

CHAPTER 8

Configuring VXLAN EVPN Multi-Site	163
About VXLAN EVPN Multi-Site	163
Guidelines and Limitations for VXLAN EVPN Multi-Site	164
Enabling VXLAN EVPN Multi-Site	165
Configuring VNI Dual Mode	166
Configuring Fabric/DCI Link Tracking	167

Configuring Fabric External Neighbors 168

CHAPTER 9

Configuring Tenant Routed Multicast 171

About Tenant Routed Multicast 171

About Tenant Routed Multicast Mixed Mode 173

Guidelines and Limitations for Tenant Routed Multicast 173

Guidelines and Limitations for Layer 3 Tenant Routed Multicast 174

Guidelines and Limitations for Layer 2/Layer 3 Tenant Routed Multicast (Mixed Mode) 174

Rendezvous Point for Tenant Routed Multicast 175

Configuring a Rendezvous Point for Tenant Routed Multicast 175

Configuring a Rendezvous Point Inside the VXLAN Fabric 176

Configuring an External Rendezvous Point 177

Configuring RP Everywhere with PIM Anycast 179

Configuring a TRM Leaf Node for RP Everywhere with PIM Anycast 180

Configuring a TRM Border Leaf Node for RP Everywhere with PIM Anycast 180

Configuring an External Router for RP Everywhere with PIM Anycast 182

Configuring RP Everywhere with MSDP Peering 184

Configuring a TRM Leaf Node for RP Everywhere with MSDP Peering 185

Configuring a TRM Border Leaf Node for RP Everywhere with MSDP Peering 186

Configuring an External Router for RP Everywhere with MSDP Peering 188

Configuring Layer 3 Tenant Routed Multicast 190

Configuring TRM on the VXLAN EVPN Spine 194

Configuring Tenant Routed Multicast in Layer 2/Layer 3 Mixed Mode 196

Configuring Layer 2 Tenant Routed Multicast 201

CHAPTER 10

Configuring VXLAN QoS 203

Information About VXLAN QoS 203

VXLAN QoS Terminology 203

VXLAN QoS Features 205

Trust Boundaries 205

Classification 205

Marking 205

Policing 205

Queuing and Scheduling 206

Traffic Shaping	206
Network QoS	206
VXLAN Priority Tunneling	206
MQC CLI	207
VXLAN QoS Topology and Roles	207
Ingress VTEP and Encapsulation in the VXLAN Tunnel	207
Transport Through the VXLAN Tunnel	208
Egress VTEP and Decapsulation of the VXLAN Tunnel	208
Classification at the Ingress VTEP, Spine, and Egress VTEP	208
IP to VXLAN	208
Inside the VXLAN Tunnel	209
VXLAN to IP	210
Decapsulated Packet Priority Selection	210
Guidelines and Limitations for VXLAN QoS	211
Default Settings for VXLAN QoS	212
Configuring VXLAN QoS	213
Configuring Type QoS on the Egress VTEP	213
Verifying the VXLAN QoS Configuration	215
VXLAN QoS Configuration Examples	215

APPENDIX A

VXLAN Bud Node Over VPC	217
VXLAN Bud Node Over VPC Overview	217
VXLAN Bud Node Over vPC Topology Example	218

APPENDIX B

DHCP Relay in VXLAN BGP EVPN	223
DHCP Relay in VXLAN BGP EVPN Overview	223
DHCP Relay in VXLAN BGP EVPN Example	224
Basic VXLAN BGP EVPN Configuration	225
DHCP Relay on VTEPs	228
Client on Tenant VRF and Server on Layer 3 Default VRF	228
Client on Tenant VRF (SVI X) and Server on the Same Tenant VRF (SVI Y)	232
Client on Tenant VRF (VRF X) and Server on Different Tenant VRF (VRF Y)	235
Client on Tenant VRF and Server on Non-Default Non-VXLAN VRF	238
Configuring VPC Peers Example	240

vPC VTEP DHCP Relay Configuration Example 242

APPENDIX C**EVPN with Transparent Firewall Insertion 245**

Overview of EVPN with Transparent Firewall Insertion 245

EVPN with Transparent Firewall Insertion Example 247

Show Command Examples 250

APPENDIX D**IPv6 Across a VXLAN EVPN Fabric 253**

Overview of IPv6 Across a VXLAN EVPN Fabric 253

Configuring IPv6 Across a VXLAN EVPN Fabric Example 253

Show Command Examples 256



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:

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

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

Examples use the following conventions:

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

Related Documentation for Cisco Nexus 9000 Series Switches

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

http://www.cisco.com/en/US/products/ps13386/tsd_products_support_series_home.html

Documentation Feedback

To provide technical feedback on this document, or to report an error or omission, please send your comments to nexus9k-docfeedback@cisco.com. We appreciate your feedback.

Communications, Services, and Additional Information

- To receive timely, relevant information from Cisco, sign up at [Cisco Profile Manager](#).
- To get the business impact you're looking for with the technologies that matter, visit [Cisco Services](#).
- To submit a service request, visit [Cisco Support](#).
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit [Cisco 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

Feature	Description	Changed in Release	Where Documented
Port VLAN Routing	Added support for Cisco Nexus 9300-EX, 9300-FX, and 9300-FX2.	7.0(3)I7(5)	Configuring Port VLAN Mapping on a Trunk Port, on page 31
RP Everywhere		7.0(3)I7(5)	Rendezvous Point for Tenant Routed Multicast, on page 175
VXLAN Tunnel Egress QoS Policy	Added support for VXLAN Tunnel Egress QoS Policy	7.0(3)I7(5)	About VXLAN Tunnel Egress QoS Policy, on page 61
Switching and routing on overlapped VLAN interfaces	Added support for the Nexus 9300, and 9500 switches.	7.0(3)I7(4)	Configuring VXLAN, on page 15
NGOAM Support	Added support for the Cisco Nexus 9336C-FX, 93300YC-FX, and 93240YC-FX2Z switches.	7.0(3)I7(4)	Configuring VXLAN OAM, on page 121
Configuring VXLAN IGMP Snooping	Added support for configuring VXLAN IGMP snooping on Cisco Nexus 9508 switches with 9636-RX line cards.	7.0(3)F3(4)	Guidelines and Limitations for VXLAN, on page 15

Feature	Description	Changed in Release	Where Documented
FC/FCoE NPV coexistence with VXLAN	FCoE N-port Virtualization (NPV) can co-exist with VXLAN.	7.0(3)I7(3)	Guidelines and Limitations for VXLAN, on page 15
VXLAN on Cisco Nexus 9348GC-FXP switch.	VXLAN supported added for the Cisco Nexus 9348GC-FXP switch.	7.0(3)I7(3)	Guidelines and Limitations for VXLAN, on page 15
QinQ-QinVNI	Tunneling feature that allows configuring a trunk port as a multi-tag port.	7.0(3)I7(3)	Overview for QinQ-QinVNI, on page 47
IP Unnumbered	IP Unnumbered support has been added for VXLAN.	7.0(3)I7(2)	IP Unnumbered
Tenant Routed Multicast + VXLAN EVPN Multi-Site	Both features can co-exist on the same physical switch.	7.0(3)I7(2)	Configuring Tenant Routed Multicast, on page 171
TRM VXLAN BGP EVPN	Added support for Cisco Nexus 9364C platform switches.	7.0(3)I7(2)	Guidelines and Limitations for Layer 3 Tenant Routed Multicast, on page 174
Tenant Routed Multicast + VXLAN EVPN Multi-Site	Both features can co-exist on the same physical switch.	7.0(3)F3(3)	Configuring Tenant Routed Multicast, on page 171 Configuring VXLAN EVPN Multi-Site, on page 163
VXLAN L2/L3 GW	Support added for: <ul style="list-style-type: none"> • PIM/ASM • ARP Suppression • IPv6 in the Overlay 	7.0(3)F3(3)	Configuring VXLAN, on page 15
PBR over VXLAN	Policy Based Routing over Virtual Extensible LAN	7.0(3)I7(1)	Configuring Policy-Based Routing
Centralized VRF Route Leaks	Centralized VRF Route Leaking using Default-Routes and Aggregates	7.0(3)I7(1)	Centralized VRF Route Leaking using Default-Routes and Aggregates, on page 56

Feature	Description	Changed in Release	Where Documented
Tenant Routed Multicast	Tenant Routing Multicast (TRM) enables multicast forwarding on the VXLAN fabric using the BGP-based EVPN control plane. TRM supports Layer 2 and Layer 3 multicast for sender and receivers on the same or different VTEPs in a tenant VRF.	7.0(3)I7(1)	Configuring Tenant Routed Multicast, on page 171
VIP/PIP	Advertises type-5 routes using the primary IP address of the VTEP interfaces as the next hop address in the VXLAN EVPN fabric.	7.0(3)I7(1)	Configuring VIP/PIP, on page 159
VXLAN EVPN Multi-Site	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.	7.0(3)I7(1)	Configuring VXLAN EVPN Multi-Site, on page 163
Configuring new CLI command lACP vpc-convergence	Added support for configuring CLI command lACP vpc-convergence for better convergence of Layer 2 EVPN VXLAN.	7.0(3)I6(1)	Guidelines and Limitations for VXLAN, on page 15
Configuring port-VLAN with VXLAN	Added support for configuring port-VLAN with VXLAN on Cisco Nexus 9300-EX and 9500 Series switches with 9700-EX line cards.	7.0(3)I6(1)	Guidelines and Limitations for VXLAN, on page 15
Configuring VXLAN on Cisco Nexus 3232C and 3264Q switches	Added support for configuring VXLAN on Cisco Nexus 3232C and 3264Q switches.	7.0(3)I6(1)	Guidelines and Limitations for VXLAN, on page 15
Configuring NGOAM Authentication	Added support for configuring NGOAM authentication.	7.0(3)I6(1)	NGOAM Authentication, on page 129

Feature	Description	Changed in Release	Where Documented
Configuring VXLAN EVPN Multihoming, VLAN Consistency Checking, and ESI ARP Suppression	Added support for VXLAN EVPN Multihoming, VLAN Consistency Checking, and ESI ARP suppression.	7.0(3)I5(2)	Introduction to Multihoming, on page 131 , Overview of VLAN Consistency Checking, on page 153 , and Overview of ESI ARP Suppression, on page 156
Configuring Selective Q-in-VNI	Added support for configuring selective Q-in-VNI.	7.0(3)I5(2)	Configuring Selective Q-in-VNI, on page 43
Configuring FHRP over VXLAN	Added support for configuring FHRP over VXLAN on the Cisco Nexus 9200, 9300, and 9300-EX Series switches.	7.0(3)I5(2)	Guidelines and Limitations for FHRP Over VXLAN, on page 50
Configuring VXLAN IGMP Snooping	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.	7.0(3)I5(2)	Guidelines and Limitations for IGMP Snooping Over VXLAN, on page 54
Configuring VRRP (FHRP) over VXLAN	Added support for configuring VRRP (FHRP) over VXLAN.	7.0(3)I5(1)	Configuring FHRP Over VXLAN, on page 50
Configuring VXLAN OAM	Added support for configuring VXLAN OAM.	7.0(3)I5(1)	VXLAN OAM Overview, on page 121
Configuring IGMP Snooping over VXLAN	Added support for configuring IGMP Snooping over VXLAN.	7.0(3)I5(1)	Configuring IGMP Snooping Over VXLAN, on page 54
VXLAN	Added support for the Cisco Nexus 93108TC-EX and 93180YC-EX switches and the X9732C-EX line card.	7.0(3)I4(2)	Guidelines and Limitations for VXLAN, on page 15
Support for suppress mac-route command	Added support for suppress mac-route command.	7.0(3)I4(1)	Commands for BGP EVPN
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 Guidelines and Limitations for VXLAN BGP EVPN
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

Feature	Description	Changed in Release	Where Documented
VXLAN FEX HIF support	Added VXLAN support for FEX host interface port.	7.0(3)I2(1)	Guidelines and Limitations for VXLAN Guidelines and Limitations for VXLAN BGP EVPN
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
BGP EVPN DHCP Relay support	Enables DHCP relay support in BGP EVPN environment.	7.0(3)I2(1)	DHCP Relay in VXLAN BGP EVPN
Q-in-VNI support	Added support for Q-in-VNI.	7.0(3)I2(1)	Configuring Q-in-VNI
Inner VLAN and Outer VLAN Mapping on a Trunk Port support	Added support for inner VLAN and outer VLAN mapping on a trunk port.	7.0(3)I2(1)	Configuring Inner VLAN and Outer VLAN Mapping on a Trunk Port
VXLAN EVPN ingress replication	Replicates BUM traffic to remote VTEP peers that are learned through the BGP EVPN control plane on Cisco Nexus 9300 Series switches.	7.0(3)I1(2)	Configuring VXLAN EVPN Ingress Replication
Port VLAN mapping on a trunk port	Enables VLAN translation between the ingress VLAN and a local VLAN on a port on Cisco Nexus 9300 Series switches.	7.0(3)I1(2)	Configuring Port VLAN Mapping on a Trunk Port
vPC Consistency Check for vPC VTEPs	Enables two switches configured as a vPC pair to exchange and verify their configuration compatibility.	7.0(3)I1(2)	vPC Consistency Check for vPC VTEPs
Static MAC for VXLAN VTEP support	Enables the configuration of static MAC addresses behind a peer VTEP on Cisco Nexus 9300 Series switches.	7.0(3)I1(2)	Configuring Static MAC for VXLAN VTEP
VXLAN BGP EVPN support	Added support for Cisco Nexus 9500 Series switches.	7.0(3)I1(2)	Configuring VXLAN BGP EVPN

Feature	Description	Changed in Release	Where Documented
Ingress replication support	Enables the replication of BUM traffic to remote peers.	7.0(3)I1(1)	Configuring Static Ingress Replication
Bud node topology support	Enables a VXLAN VTEP device to also be an IP transit device.	7.0(3)I1(1)	Bud Node Topology
VXLAN BGP EVPN support	Enables the learning of remote VTEPs, overlay MACs, and routes through the BGP EVPN control plane protocol on Cisco Nexus 9300 Series switches.	7.0(3)I1(1)	Configuring VXLAN BGP EVPN



CHAPTER 2

Overview

This chapter contains the following sections:

- [Licensing Requirements, on page 7](#)
- [Supported Platforms, on page 7](#)
- [VXLAN Overview, on page 7](#)
- [VXLAN Encapsulation and Packet Format, on page 8](#)
- [VXLAN Tunnel Endpoint, on page 8](#)
- [VXLAN Packet Forwarding Flow, on page 9](#)
- [Cisco Nexus 9000 as Hardware-Based VXLAN Gateway, on page 9](#)
- [vPC Consistency Check for vPC VTEPs, on page 9](#)
- [Static Ingress Replication, on page 11](#)
- [Bud Node Topology, on page 11](#)
- [VXLAN BGP EVPN Control Plane , on page 12](#)

Licensing Requirements

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

Supported Platforms

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

VXLAN Overview

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.

Parameter	vPC Check Type	Description
VLAN-VNI mapping	Type-1-nongraceful	Brings down the affected VLANs on vPC ports on both sides.
VTEP-Member-VNI	Type-1-nongraceful	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.)

Parameter	vPC Check Type	Description
VTEP-emulated IP	Type-1-nongraceful	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.
NVE Oper State	Type-1-nongraceful	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.
NVE Host-Reachability Protocol	Type-1-nongraceful	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.

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.



Note 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

Name	Type	Local Value	Peer Value
Vlan to Vn-segment Map	1	1024 Relevant Map(s)	1024 Relevant Map(s)
STP Mode	1	MST	MST
STP Disabled	1	None	None
STP MST Region Name	1	""	""
STP MST Region Revision	1	0	0
STP MST Region Instance to	1		
VLAN Mapping			
STP Loopguard	1	Disabled	Disabled
STP Bridge Assurance	1	Enabled	Enabled

STP Port Type, Edge	1	Normal, Disabled,	Normal, Disabled,
BPDUFILTER, Edge BPDUGuard		Disabled	Disabled
STP MST Simulate PVST	1	Enabled	Enabled
Nve Oper State, Secondary	1	Up, 4.4.4.4	Up, 4.4.4.4
IP			
Nve Vni Configuration	1	10002-11025	10002-11025
Allowed VLANs	-	1-1025	1-1025
Local suspended VLANs	-	-	-

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.



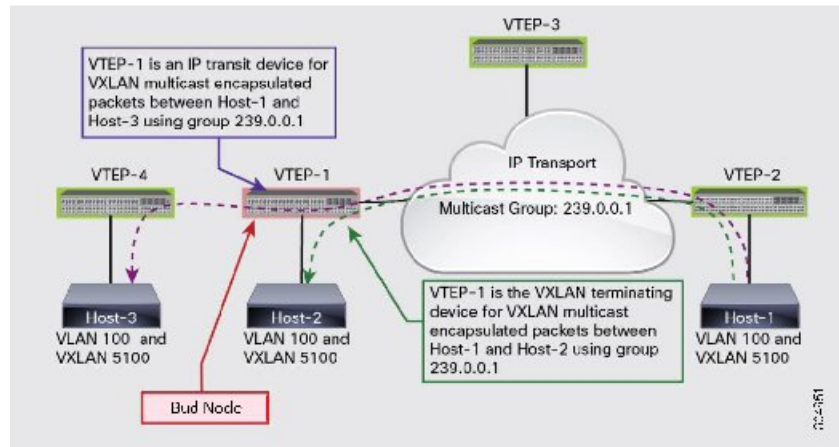
Note Cisco NX-OS supports multiple remote peers in one segment and also allows the same remote peer in multiple segments.

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 15](#)
- [Configuring VXLAN, on page 30](#)
- [VXLAN Tunnel Egress QoS Policy, on page 61](#)
- [Verifying the VXLAN Configuration, on page 63](#)
- [Example of VXLAN Bridging Configuration, on page 65](#)

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

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

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

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)/6(1)

ACL Direction	ACL Type	VTEP Type	Port Type	Flow Direction	Traffic Type	Supported
Ingress	PACL	Ingress VTEP	L2 port	Access to Network [GROUP:encap direction]	Native L2 traffic [GROUP:inner]	YES (works only for base port PO)
Egress	PACL	Egress VTEP	L2 port	Network to Access [GROUP:decap direction]	Native L2 traffic [GROUP:inner]	NO
Ingress	VACL	Ingress VTEP	VLAN	Access to Network [GROUP:encap direction]	Native L2 traffic [GROUP:inner]	YES
Egress	VACL	Egress VTEP	VLAN	Network to Access [GROUP:decap direction]	Native L2 traffic [GROUP:inner]	YES

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

- Non-blocking Multicast (NBM) running on a VXLAN enabled switch is not supported. Feature nbm may disrupt VXLAN underlay multicast forwarding.
- 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.
- IP Unnumbered for VXLAN underlay is supported starting with Cisco NX-OS Release 7.0(3)I7(2). Only single unnumbered link between same devices (for example, spine - leaf) is supported. If multiple physical links are connecting the same leaf and spine, you must use the single L3 port-channel with unnumbered link.

- 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-OS Multicast 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-OS 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:

```
interface port-channel10
  switchport
  switchport mode trunk
  switchport trunk allowed vlan 1001-1200
  spanning-tree port type edge trunk
  spanning-tree bpduguard 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.
- To remove configurations from an NVE interface, we recommend manually removing each configuration rather than using the **default interface nve** command.
- 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.
- RACLs are not supported on Layer 3 uplinks for VXLAN traffic. Egress VACLs support is not available for de-capsulated packets in the network to access direction on the inner payload.
As a best practice, use PACLS/VACLs for the access to the network direction.
- QoS classification is not supported for VXLAN traffic in the network to access direction on the Layer 3 uplink interface.
- The QoS buffer-boost feature is not applicable for VXLAN traffic.
- 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 supported for a VNI only if the VTEP hosts the First-Hop Gateway (Distributed Anycast Gateway) for this VNI. The VTEP and SVI for this VLAN must be properly configured for the Distributed Anycast Gateway operation (for example, global anycast gateway MAC address configured and anycast gateway with the virtual IP address on the SVI).
- ARP suppression is a per-L2VNI fabric-wide setting in the VXLAN fabric. Enable or disable this feature consistently across all VTEPs in the fabric. Inconsistent ARP suppression configuration across VTEPs is not supported.
- 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.

- If multiple VTEPs use the same multicast group address for underlay multicast but have different VNIs, the VTEPs should have at least one VNI in common. Doing so ensures that NVE peer discovery occurs and underlay multicast traffic is forwarded correctly. For example, leafs L1 and L4 could have VNI 10 and leafs L2 and L3 could have VNI 20, and both VNIs could share the same group address. When leaf L1 sends traffic to leaf L4, the traffic could pass through leaf L2 or L3. Because NVE peer L1 is not learned on leaf L2 or L3, the traffic is dropped. Therefore, VTEPs that share a group address need to have at least one VNI in common so that peer learning occurs and traffic is not dropped. This requirement applies to VXLAN bud-node topologies.
- NVE source interface loopback for VTEP should only be IPv4 address. Use of IPv6 address for NVE source interface is not supported.
- Next hop address in overlay (in bgp l2vpn evpn address family updates) should be resolved in underlay URIB to the same address family. For example, the use of VTEP (NVE source loopback) IPv4 addresses in fabric should only have BGP l2vpn evpn peering over IPv4 addresses.
- 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 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:

```
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, encap routes are installed for multicast.

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

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

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

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



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

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



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

- When the vPC domain is shut, the loopback interface used by NVE on the VTEP with shutdown vPC domain is brought down and the status is Admin Shut. This is done so that the route to the loopback is withdrawn on the upstream and that the upstream can divert all traffic to the other vPC VTEP.
- When peer-link is no-shut, the NVE loopback address is brought up again and the route is advertised upstream, attracting traffic.
- For vPC, the loopback interface has 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 this 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:

```
switchch# sh ru int vlan 2

interface Vlan2
  description backup1_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.



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.
- Using the **ip forward** command enables the VTEP to forward the VXLAN de-capsulated packet destined to its router IP to the SUP/CPU.
- Before configuring it as an SVI, the backup VLAN needs to be configured on Cisco Nexus 9200, 9300-EX, 9300-FX, and 9300-FX2 platform switches as an infra-VLAN with the **system nve infra-vlans** command.

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

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. *
- 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).



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



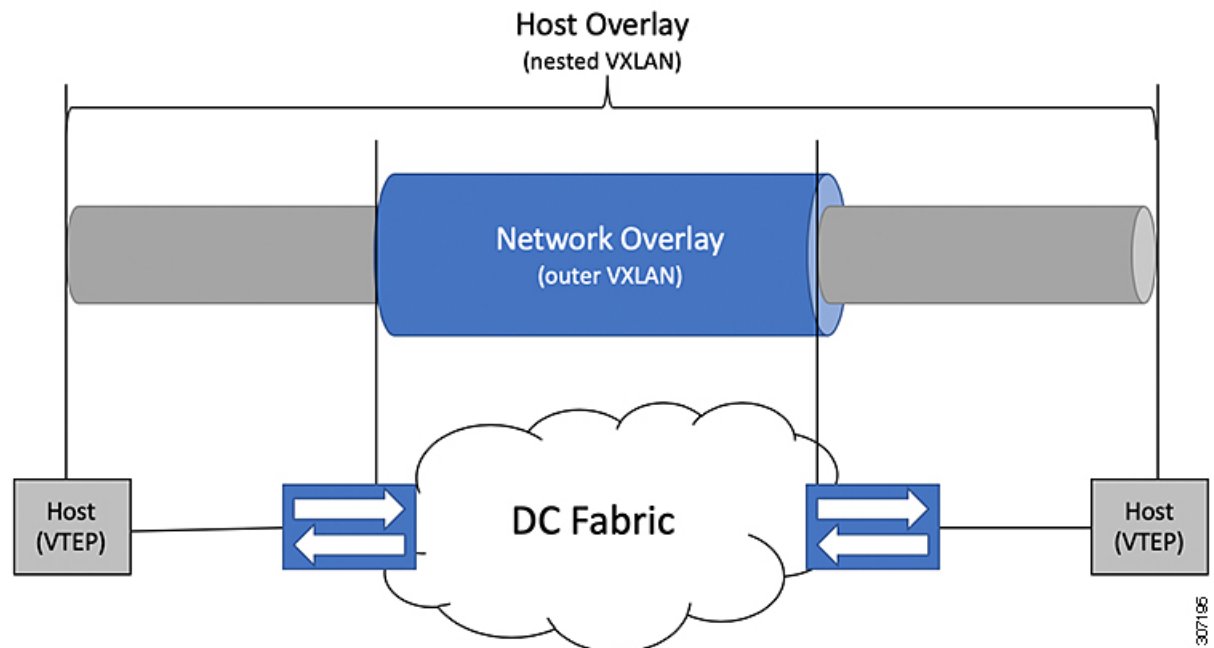
Note * Not required for static ingress replication or BGP EVPN ingress replication.

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) on the Cisco Nexus 9200, 9300-EX, 9300-FX, 9300-FX2, 9500-EX, and 9500-FX platform switches.

Figure 2: Host Overlay



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

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

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

The following attachment modes are supported for Nested VXLAN:

- Untagged traffic (in native VLAN on a trunk port or on an access port)
- Tagged traffic (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

Configuring VXLAN

Enabling VXLANs

Procedure

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

Mapping VLAN to VXLAN VNI

Procedure

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

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 NFE 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:

```
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.

```
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

	Command or Action	Purpose
Step 1	configure terminal	Enters global configuration mode.
Step 2	interface <i>type port</i>	Enters interface configuration mode.
Step 3	[no] switchport vlan mapping enable	Enables VLAN translation on the switch port. VLAN translation is disabled by default. Note Use the no form of this command to disable VLAN translation.
Step 4	[no] switchport vlan mapping <i>vlan-id translated-vlan-id</i>	Translates a VLAN to another VLAN. <ul style="list-style-type: none"> • The range for both the <i>vlan-id</i> and <i>translated-vlan-id</i> arguments is from 1 to 4094. • You can configure VLAN translation between the ingress (incoming) VLAN and a local (translated) VLAN on a port. For the traffic arriving on the interface where VLAN translation is enabled, the incoming

	Command or Action	Purpose
		<p>VLAN is mapped to a translated VLAN that is VXLAN enabled.</p> <p>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.</p> <p>Note Use the no form of this command to clear the mappings between a pair of VLANs.</p>
Step 5	[no] switchport vlan mapping all	Removes all VLAN mappings configured on the interface.
Step 6	(Optional) copy running-config startup-config	<p>Copies the running configuration to the startup configuration.</p> <p>Note The VLAN translation configuration does not become effective until the switch port becomes an operational trunk port</p>
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

	Command or Action	Purpose
Step 1	configure terminal	Enters global configuration mode.
Step 2	interface <i>type port</i>	Enters interface configuration mode.
Step 3	[no] switchport mode trunk	Enters trunk configuration mode.

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

Example

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

```
switch# 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

VLAN	MAC Address	Type	age	Secure	NTFY	Ports
* 111	0000.0092.0001	dynamic	0	F	F	nve1(100.100.100.254)
* 111	0000.0940.0001	dynamic	0	F	F	Eth1/1

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:

Procedure

	Command or Action	Purpose
Step 1	configure terminal	Enters global configuration mode.
Step 2	interface nve <i>x</i>	Creates a VXLAN overlay interface that terminates VXLAN tunnels. Note Only 1 NVE interface is allowed on the switch.
Step 3	source-interface <i>src-if</i>	The source interface must be a loopback interface that is configured on the switch with a valid /32 IP address. This /32 IP address must be known by the transient devices in the transport network and the remote VTEPs. This is accomplished by advertising it through a dynamic routing protocol in the transport network.
Step 4	member vni <i>vni</i>	Associate VXLAN VNIs (Virtual Network Identifiers) with the NVE interface.
Step 5	mcast-group <i>start-address</i> [<i>end-address</i>]	Assign a multicast group to the VNIs. Note used only for BUM traffic

Configuring Static MAC for VXLAN VTEP

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



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

Procedure

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

	Command or Action	Purpose
Step 5	(Optional) show mac address-table static interface nve x	Displays the static MAC addresses pointing to the remote VTEP.

Example

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

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

Legend:

* - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
age - seconds since last seen, + - primary entry using vPC Peer-Link,
(T) - True, (F) - False

VLAN	MAC Address	Type	age	Secure	NTFY	Ports
* 501	0047.1200.0000	static	-	F	F	nve1(33.1.1.3)
* 601	0049.1200.0000	static	-	F	F	nve1(33.1.1.4)

Disabling VXLANs

Procedure

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

Configuring BGP EVPN Ingress Replication

The following enables BGP EVPN with ingress replication for peers.

Procedure

	Command or Action	Purpose
Step 1	configure terminal	Enters global configuration mode.
Step 2	interface nve x	Creates a VXLAN overlay interface that terminates VXLAN tunnels. Note Only 1 NVE interface is allowed on the switch.

	Command or Action	Purpose
Step 3	source-interface <i>src-if</i>	The source interface must be a loopback interface that is configured on the switch with a valid /32 IP address. This /32 IP address must be known by the transient devices in the transport network and the remote VTEPs. This is accomplished by advertising it through a dynamic routing protocol in the transport network.
Step 4	member vni <i>vni</i>	Associate VXLAN VNIs (Virtual Network Identifiers) with the NVE interface.
Step 5	ingress-replication protocol bgp	Enables BGP EVPN with ingress replication for the VNI.

Configuring Static Ingress Replication

The following enables static ingress replication for peers.

Procedure

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

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
!Time: Wed Jul 20 23:26:25 2016
```

```
version 7.0(3u)I4(2u)
```

```
interface nve1
  no shutdown
  source-interface loopback0
  host-reachability protocol bgp
  member vni 900001 associate-vrf
  member vni 2000980
  mcast-group 225.4.0.1
```

- 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

	Command or Action	Purpose
Step 1	configure terminal	Enters global configuration mode.
Step 2	interface <i>type port</i>	Enters interface configuration mode.
Step 3	switchport mode dot1q-tunnel	Creates a 802.1Q tunnel on the port.
Step 4	switchport access vlan <i>vlan-id</i>	Specifies the port assigned to a VLAN.

	Command or Action	Purpose
Step 5	spanning-tree bpdudfilter enable	Enables BPDU Filtering for the specified spanning tree edge interface. By default, BPDU Filtering is disabled.
Step 6	interface nve x	Creates a VXLAN overlay interface that terminates VXLAN tunnels. Note This step is required for NX-OS 7.0(3)I2(2) and earlier releases. This step is not required for NX-OS 7.0(3)I3(1) and later releases.
Step 7	overlay-encapsulation vxlan-with-tag	Enables Q-in-VNI. Note This step is required for NX-OS 7.0(3)I2(2) and earlier releases: This step is not required for NX-OS 7.0(3)I3(1) and later releases. 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).

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 bpdudfilter 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
switch(config-if)#
```

Configuring Selective Q-in-VNI

Selective Q-in-VNI is a VXLAN tunneling feature that allows a user specific range of customer VLANs on a port to be associated with one specific provider VLAN. Packets that come in with a VLAN tag that matches any of the configured customer VLANs on the port are tunneled across the VXLAN fabric using the properties of the service provider VNI. The VXLAN encapsulated packet carries the customer VLAN tag as part of the L2 header of the inner packet.

The packets that come in with a VLAN tag that is not present in the range of the configured customer VLANs on a selective Q-in-VNI configured port are dropped. This includes the packets that come in with a VLAN tag that matches the native VLAN on the port. Packets coming untagged or with a native VLAN tag are L3 routed using the native VLAN's SVI that is configured on the selective Q-in-VNI port (no VXLAN).

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.

```
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 bpdupfilter 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:

```
vlan 50
  vn-segment 10050
```

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

```
member vni 10050
  ingress-replication protocol static
  peer-ip 100.1.1.3
  peer-ip 100.1.1.5
  peer-ip 100.1.1.10
```

- See the following example for the interface nve configuration:

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

- See the following example for the native VLAN configuration:

```
vlan 150
interface vlan150
  no shutdown
  ip address 150.1.150.6/24
  ip pim sparse-mode
```

- See the following example for configuring selective Q-in-VNI on a port. In this example, native VLAN 150 is used for routing the untagged packets. Customer VLANs 200-700 are carried across the dot1q tunnel. The native VLAN 150 and the provider VLAN 50 are the only VLANs allowed.

```
switch# config terminal
switch(config)#interface Ethernet 1/31
switch(config-if)#switchport
switch(config-if)#switchport mode trunk
switch(config-if)#switchport trunk native vlan 150
switch(config-if)#switchport vlan mapping 200-700 dot1q-tunnel 50
switch(config-if)#switchport trunk allowed vlan 50,150
switch(config-if)#no shutdown
```

Configuring Q-in-VNI with LACP Tunneling

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

Procedure

	Command or Action	Purpose
Step 1	configure terminal	Enters global configuration mode.
Step 2	interface <i>type port</i>	Enters interface configuration mode.
Step 3	switchport mode dot1q-tunnel	Enables dot1q-tunnel mode.
Step 4	switchport access vlan <i>vlan-id</i>	Specifies the port assigned to a VLAN.
Step 5	interface nve <i>x</i>	Creates a VXLAN overlay interface that terminates VXLAN tunnels.
Step 6	overlay-encapsulation vxlan-with-tag tunnel-control-frames lacp	<p>Enables Q-in-VNI for LACP tunneling.</p> <p>Note Use this form of the command for NX-OS 7.0(3)I3(1) and later releases.</p> <p>For NX-OS 7.0(3)I2(2) and earlier releases, use the overlay-encapsulation vxlan-with-tag tunnel-control-frames command.</p>

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

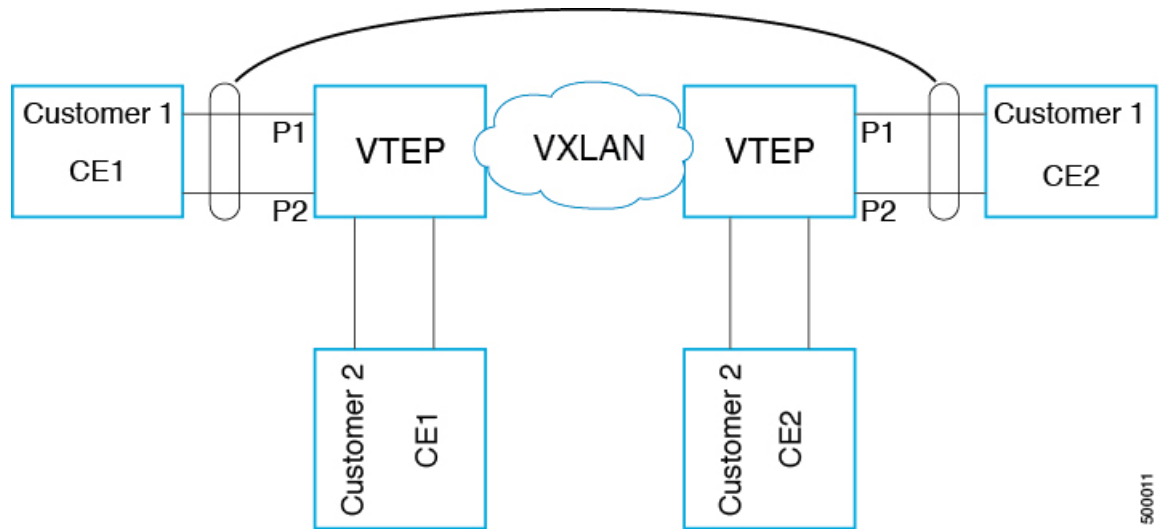


Note

- 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 3: LACP Tunneling Over VXLAN P2P Tunnels



500011

**Note**

- Q-in-VNI can be configured to tunnel LACP packets. (Able to provide port-channel connectivity across data-centers.)
 - Gives impression of L1 connectivity and co-location across data-centers.
 - Exactly two sites. Traffic coming from P1 of CE1 goes out of P1 of CE2. If P1 of CE1 goes down, LACP provides coverage (over time) to redirect traffic to P2.
- Uses static ingress replication with VXLAN with flood and learn. Each port of the port channel is configured with Q-in-VNI. There are multiple VNIs for each member of a port-channel and each port is pinned to specific VNI.
 - To avoid saturating the MAC, you should turn off/disable learning of VLANs.
- Configuring Q-in-VNI to tunnel LACP packets is not supported for VXLAN EVPN.
- The number of port-channel members supported is the number of ports supported by the VTEP.

Configuring QinQ-QinVNI

Overview for QinQ-QinVNI

- QinQ-QinVNI is a VXLAN tunneling feature that allows you to configure a trunk port as a multi-tag port to preserve the customer VLANs that are carried across the network.
- On a port that is configured as multi-tag, packets are expected with multiple-tags or at least one tag. When multi-tag packets ingress on this port, the outer-most or first tag is treated as provider-tag or provider-vlan. The remaining tags are treated as customer-tag or customer-vlan.

- This feature is supported on both vPC and non-vPC ports.
- Ensure that the **switchport trunk allow-multi-tag** command is configured on both of the vPC-peers. It is a type 1 consistency check.
- This feature is supported with VXLAN Flood and Learn and VXLAN EVPN.
- 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 is not supported when snooping is enabled on the VXLAN VLAN.
- You need at least one multi-tag trunk port allowing the provider VLANs in **up** state on both 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

	Command or Action	Purpose
Step 1	configure terminal Example: <pre>switch# configure terminal</pre>	Enters global configuration mode.

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

Example

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

Removing a VNI

Use this procedure to remove a VNI.

Procedure

-
- | | |
|---------------|--|
| Step 1 | Remove the VNI under NVE. |
| Step 2 | Remove the VRF from BGP (applicable when decommissioning for Layer 3 VNI). |
| Step 3 | Delete the SVI. |
| Step 4 | Delete the VLAN and VNI. |
-

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 4: FHRP over VXLAN Leafs as Layer 3 Gateway

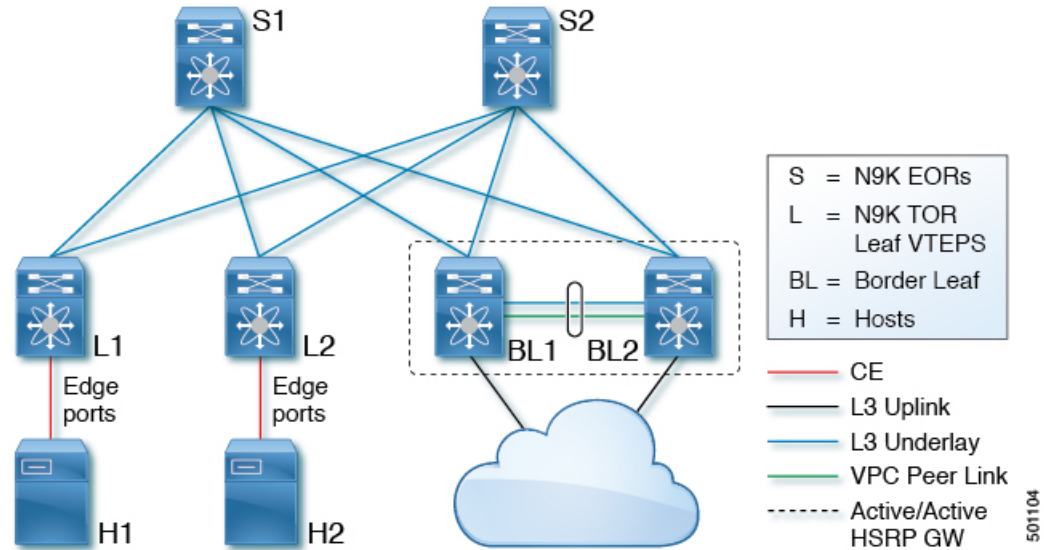
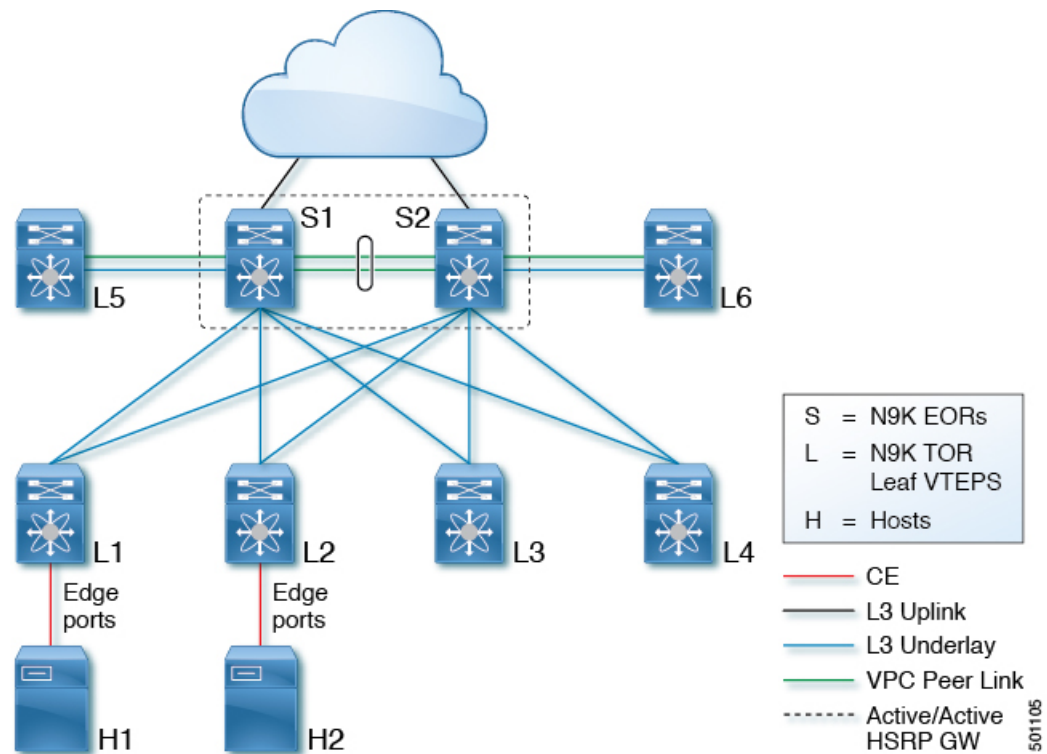


Figure 5: FHRP over VXLAN Spine as Layer 3 Gateway



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
```



Note 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:

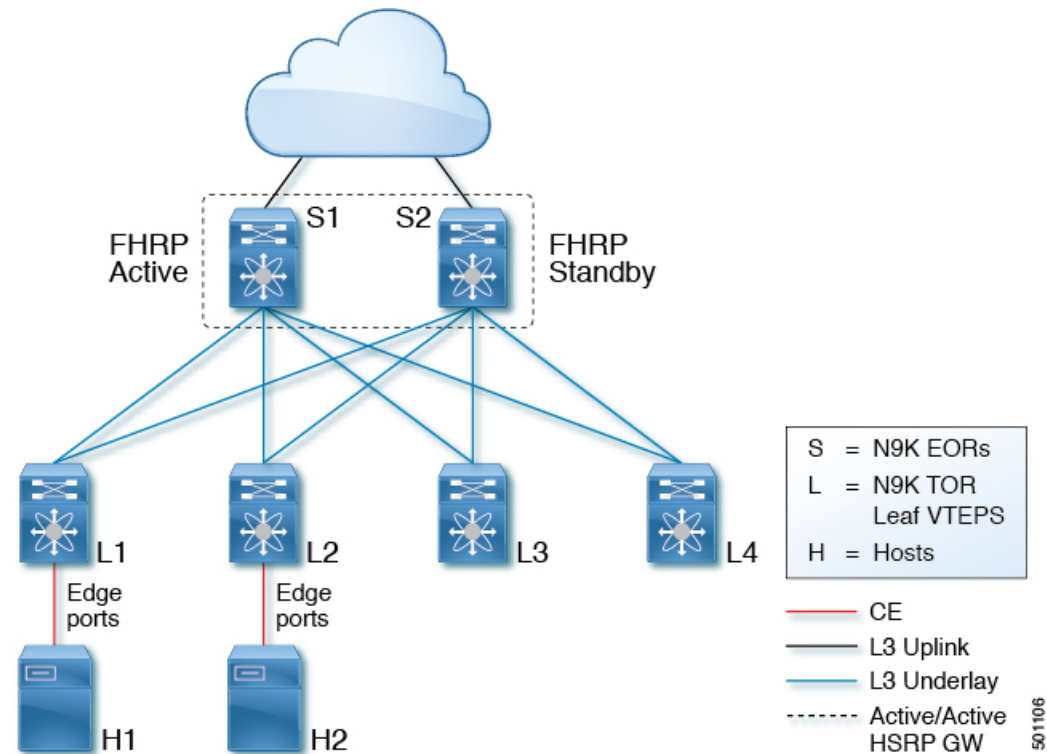
- Cisco Nexus 9300 Series switches
- N9K-X9536PQ line cards
- N9k-X9564TX line cards
- N9K-X9564PX line cards



Note 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:

Figure 6: Configuring FHRP Over VXLAN on the Spine Layer



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 operates in an active/standby state 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
```



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

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)I7(6), IGMP snooping on VXLAN VLANs is supported on N9K-C9364C, N9K-C93180-FX, and N9K-C9336C-FX2 platform switches.
- 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

	Command or Action	Purpose
Step 1	switch(config)# ip igmp snooping vxlan	Enables IGMP snooping for VXLAN VLANs. You have to explicitly configure this command to enable snooping for VXLAN VLANs.
Step 2	switch(config)# ip igmp snooping disable-nve-static-router-port	Configures IGMP snooping over VXLAN to not include NVE as static mrouter port using this global CLI command. IGMP snooping over VXLAN has the NVE interface as mrouter port by default.
Step 3	switch(config)# system nve ipmc global index-size ? Example: switch(config)# system nve ipmc global index-size ? <1000-7000> Ipmc allowed size	Configures the VXLAN global IPMC index size. IGMP snooping over VXLAN uses the IPMC indexes from the NVE global range on the Cisco Nexus 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.

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: <pre>switch# configure terminal</pre>	Enters global configuration mode.
Step 2	hardware profile [vxlan mpls] module {module all} Example: <pre>switch(config)# hardware profile vxlan module all</pre>	Configures VXLAN on all line cards. Note All line cards must be either VXLAN or MLPS. They cannot be mixed.
Step 3	<pre>switch(config)# reload</pre> Example: <pre>switch(config)# reload</pre>	Reloads the Cisco NX-OS software.
Step 4	<pre>switch(config)# show hardware profile module [module all]</pre> Example: <pre>switch(config)# show hardware profile module all</pre>	Displays the line cards that are configured with VXLAN.

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.

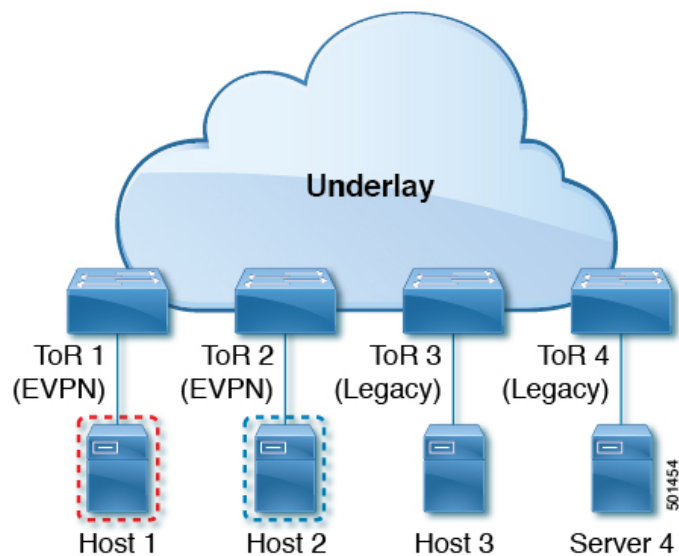
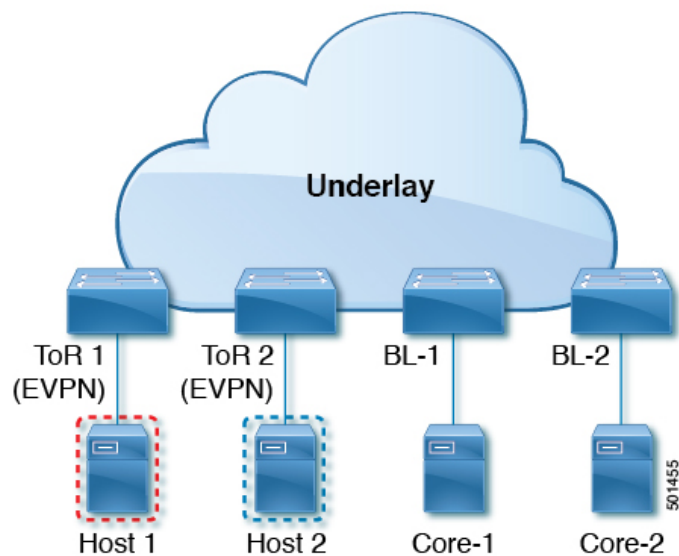
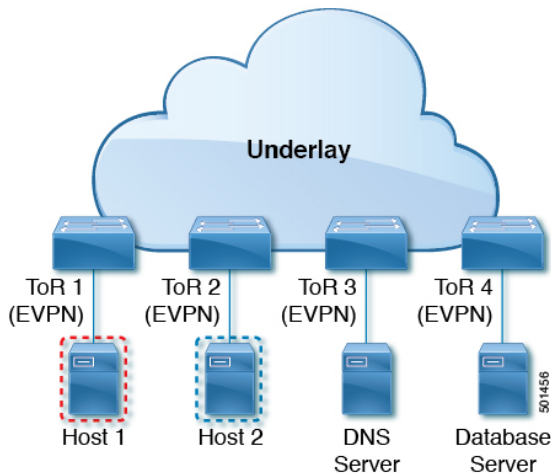
Figure 7: EVPN Brown-field Deployment*Figure 8: Border Leaf Connection to Core / Internet via Default-VRF*

Figure 9: Common Services



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 $O(N \times M)$ 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.

```
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
    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.

```
router bgp 1001
  vrf vrf120
    address-family ipv4 unicast
      network 0.0.0.0/0 // advertises default route to host leaf VTEPs
      advertise l2vpn evpn
      redistribute hmm route-map permitall
      maximum-paths 64
      maximum-paths ibgp 64
    address-family ipv6 unicast
      network 0::/0 // advertises default route to host leaf VTEPs
      advertise l2vpn 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.

```
.

switch# sh run vrf vrf150
!Command: show running-config vrf vrf150
!Time: Thu Aug 3 16:54:57 2017
version 7.0(3)I7(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 65535:150 evpn
route-target import 65535:250 //import VRF250 routes
route-target import 65535:250 evpn //import VRF250 routes
route-target export 65535:150
route-target export 65535:150 evpn
address-family ipv6 unicast
route-target import 65535:150
route-target import 65535:150 evpn
route-target import 65535:250 //import VRF250 routes
route-target import 65535:250 evpn //import VRF250 routes
route-target export 65535:150
route-target export 65535:150 evpn
router bgp 1001
vrf vrf150
address-family ipv4 unicast
advertise l2vpn 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 l2vpn 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 65535:250
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 65535:250
route-target export 65535:250 evpn
router bgp 1001
vrf vrf250
address-family ipv4 unicast
advertise l2vpn 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

```

```

advertise l2vpn evpn
redistribute hmm route-map permitall
aggregate-address 2001:0:11:50::/63
aggregate-address 2001:0:21:50::/63
maximum-paths 64
maximum-paths ibgp 64

```

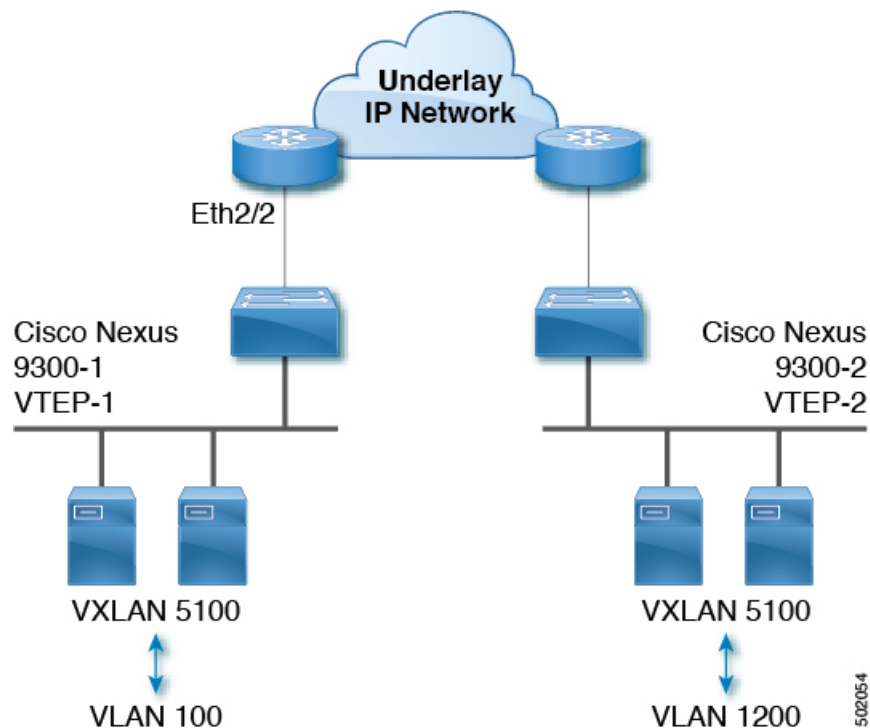
VXLAN Tunnel Egress QoS Policy

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.

Figure 10: An Example VXLAN Fabric



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

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

	Command or Action	Purpose
	<code>switch(config-if) # source-interface loopback1</code>	a valid /32 IP address. This /32 IP address must be known by the transient devices in the transport network and the remote VTEPs. This is accomplished by advertising it through a dynamic routing protocol in the transport network.
Step 8	member vni vni Example: <code>switch(config-if) # member vni 10101-10102</code>	Associate VXLAN VNIs (Virtual Network Identifiers) with the NVE interface.
Step 9	suppress-arp Example: <code>switch(config-if) # suppress-arp</code>	Configure to suppress ARP under Layer 2 VNI.

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)/1(1))

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

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

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

Command	Purpose
show tech-support vxlan [platform]	Displays related VXLAN tech-support information.
show interface {ethernet <i>slot/port</i> port-channel <i>port</i>} vlan mapping	Displays VLAN mapping information for a specific interface or port channel.
show logging level nve	Displays logging level.
show tech-support nve	Displays related NVE tech-support information.
show run interface nve <i>x</i>	Displays NVE overlay interface configuration.
show nve interface	Displays NVE overlay interface status.
show nve peers	Displays NVE peer status.
show nve peers <i>peer_IP_address</i> interface <i>interface_ID</i> counters	Displays per NVE peer statistics.
clear nve peers <i>peer_IP_address</i> interface <i>interface_ID</i> counters	Clears per NVE peer statistics.
clear nve peer-ip <i>peer-ip-address</i>	Clears stale NVE peers. Stale NVE peers are peers that do not have MAC addresses learnt behind them.
show nve vni	Displays VXLAN VNI status.
show nve vni ingress-replication	Displays the mapping of VNI to ingress-replication peer list and uptime for each peer.
show nve vni <i>vni_number</i> counters	Displays per VNI statistics.
clear nve vni <i>vni_number</i> counters	Clears per VNI statistics.
show nve vxlan-params	Displays VXLAN parameters, such as VXLAN destination or UDP port.
show mac address-table static interface nve 1	Displays static MAC information.

Command	Purpose
<code>show vxlan interface</code>	Displays VXLAN interface status for 9200 platform switches. .
<code>show vxlan interface count</code>	<p>Displays VXLAN VLAN logical port VP count.</p> <p>Note A VP is allocated on a per-port per-VLAN basis. The sum of all VPs across all VXLAN-enabled Layer 2 ports gives the total logical port VP count. For example, if there are 10 Layer 2 trunk interfaces, each with 10 VXLAN VLANs, then the total VXLAN VLAN logical port VP count is $10 \times 10 = 100$.</p>

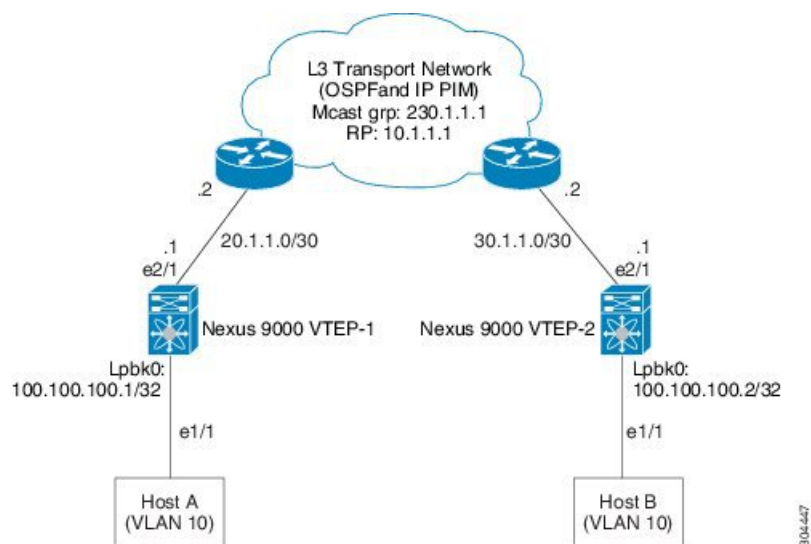
Table 6: Display VXLAN configuration information (Release 7.0(3)I2(2) and later)

Command	Purpose
<code>show run track</code>	Displays tracking information for running-config.
<code>show track</code>	<p>Displays tracking information for IP prefix for an endpoint.</p> <p>Note Assists tracking IPv4 routes with route-type HMM information.</p>

Example of VXLAN Bridging Configuration

- An example of a loopback interface configuration and routing protocol configuration:

Figure 11: VXLAN topology for VTEP



- 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 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-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

```

```

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 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

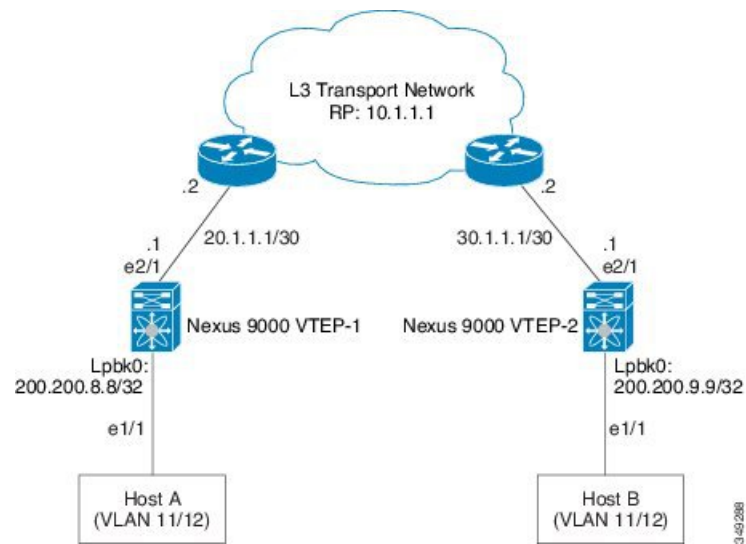
switch-vtep-2(config)# feature nv overlay
switch-vtep-2(config)# feature vn-segment-vlan-based
switch-vtep-2(config)# interface e1/1
switch-vtep-2(config-if)# switchport
switch-vtep-2(config-if)# switchport access vlan 10
switch-vtep-2(config-if)# no shutdown
switch-vtep-2(config)# interface nve1
switch-vtep-2(config-if)# no shutdown
switch-vtep-2(config-if)# source-interface loopback0

switch-vtep-2(config-if)# member vni 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-vlan)# exit

```

- An example of an ingress replication topology:

Figure 12: Ingress Replication topology



- Nexus 9000 VTEP-1 configuration:

```

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)# 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

```

```

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:

```

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
```

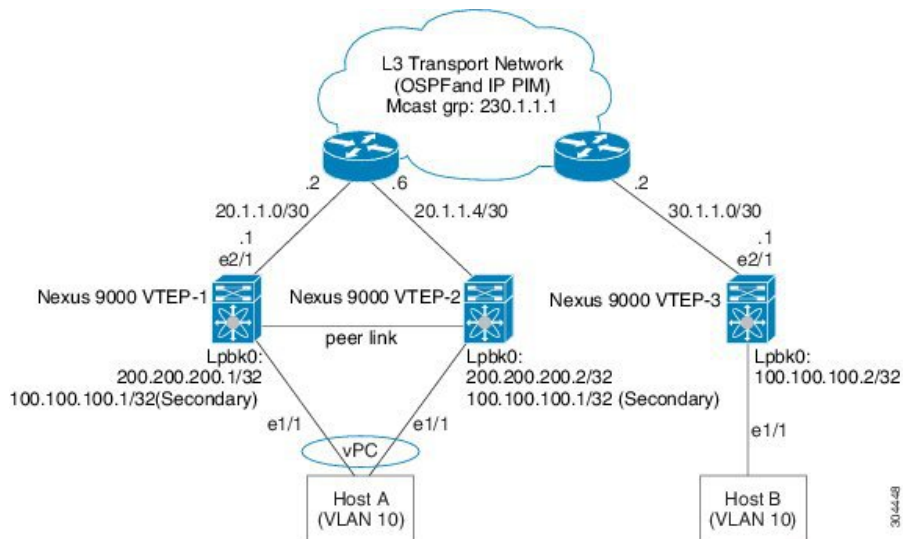
```
switch-vtep-2# show nve vni ingress-replication
```

Interface	VNI	Replication List	Up Time
nve1	10011	200.200.8.8 200.200.10.10	07:42:23 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 13: 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 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)# 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)# 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-vlan)# 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)# 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)# interface e1/1
switch-vtep-2(config-if)# channel-group 10 mode active
switch-vtep-2(config-if)# no shutdown

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 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-vlan)# 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.



CHAPTER 4

Configuring VXLAN BGP EVPN

This chapter contains the following sections:

- [Information About VXLAN BGP EVPN, on page 73](#)
- [Configuring VXLAN BGP EVPN, on page 85](#)
- [Verifying the VXLAN BGP EVPN Configuration, on page 98](#)
- [Example of VXLAN BGP EVPN \(EBGP\), on page 99](#)
- [Example of VXLAN BGP EVPN \(IBGP\), on page 108](#)
- [Example Show Commands, on page 117](#)

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.
- ACLs 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.
 - On Cisco Nexus 9000 PX/TX/PQ switches configured as VXLAN VTEPs, if any ALE 40G port is used as a VXLAN underlay port, configuring subinterfaces on either this or any other 40G port is not allowed and could lead to VXLAN traffic loss.
 - For Cisco NX-OS 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.
 - You must bind NVE to a loopback address that is separate from other loopback addresses that are required by Layer 3 protocols. NVE and other Layer 3 protocols using the same loopback is not supported.
 - 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, and 9300-FX/FX2 platform switches
 - Cisco Nexus 9300 platform switches
 - Cisco Nexus 9500 platform switches with X9500 line cards
 - Cisco Nexus 9500 platform switches with -EX and -FX 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 7.0(3)I5(1), the "System Routing Mode: template-vxlan-scale" is required on the following hardware platforms:
 - Cisco Nexus 9300-EX platform switches
 - Cisco Nexus 9500 platform switches with -EX line cards
- Changing the "System Routing Mode" requires a reload of the switch.
- For Cisco NX-OS Release 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 platform switch with -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, 9300-EX, and 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 Cisco NX-OS Release 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 Cisco NX-OS 7.0(3)I2(3) release, the VXLAN network identifier (VNID) 16777215 is reserved and should not be configured explicitly.
- For Cisco NX-OS Release 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).
- The **vpc orphan-ports suspend** command must be enabled for orphan ports that are connected to Cisco Nexus 9000 vPC VTEPs.
- 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.



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

Considerations for VXLAN BGP EVPN Deployment

- A loopback address is required when using the **source-interface config** command. The loopback address represents the local VTEP IP.
- During boot-up of a switch (7.0(3)I2(2) and later), you can use the **source-interface hold-down-time hold-down-time** command to suppress advertisement of the NVE loopback address until the overlay has converged. The range for the *hold-down-time* is 0 - 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 anycast 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 Cisco NX-OS Release 7.0(3)I1(2) through 7.0(3)I5(2)
- Cisco Nexus 9300-EX, 9300-FX, and 9300-FX2 platform switches beginning with Cisco NX-OS Release 7.0(3)I1(2) through 7.0(3)I5(2)
- Cisco Nexus 9300 platform switches beginning with Cisco NX-OS Release 7.0(3)I1(2)



Note Beginning with Cisco NX-OS Release 7.0(3)I5(2), the Cisco Nexus 9200, 9300-EX, and 9300-FX/FX2 platform switches 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, and 9300-FX/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.



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 the vPC domain is shut, the loopback interface used by NVE on the VTEP with shutdown vPC domain is brought down and the status is **Admin Shut**. This is done so that the route to the loopback is withdrawn on the upstream and that the upstream can divert all traffic to the other vPC VTEP.
- When peer-link is no-shut, the NVE loopback address is brought up again and the route is advertised upstream, attracting traffic.
- For vPC, the loopback interface has 2 IP addresses: the primary IP address and the secondary IP address.

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

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

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

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

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

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

```
switch# sh ru vpc

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

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

The following is an example of SVI with PIM enabled:

```
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.

```
switchch# 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. *
 - 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.*



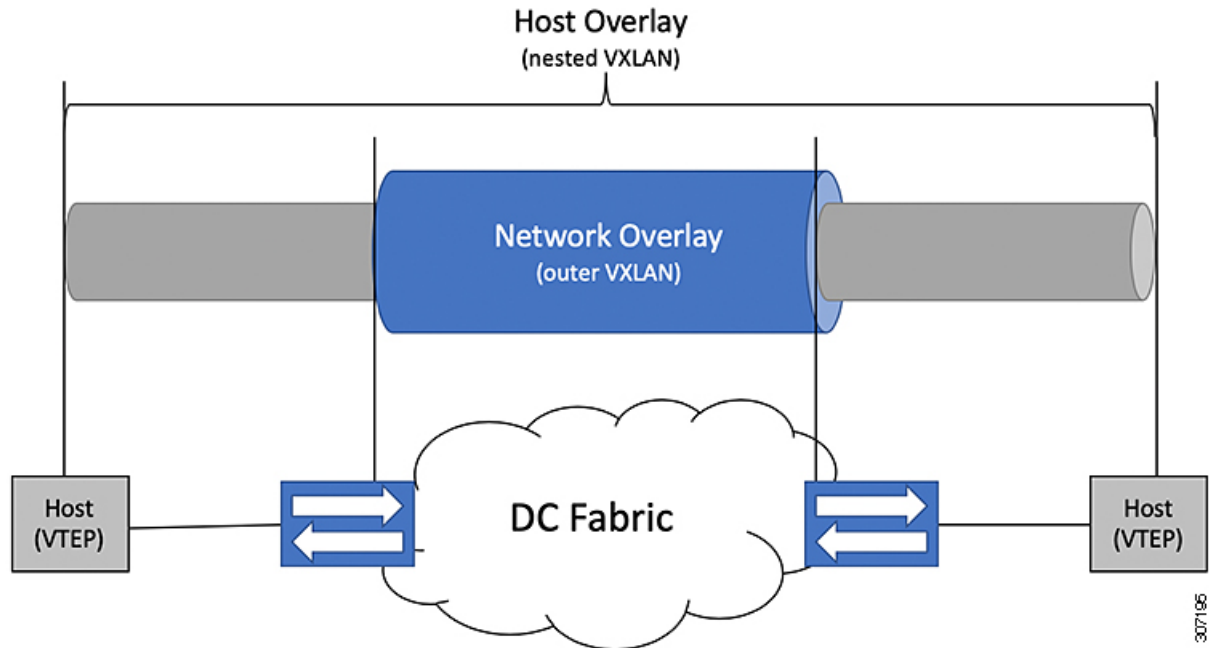
Note * Not required for static ingress replication or BGP EVPN ingress replication.

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 14: 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.

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

Command	Description
system vlan nve-overlay id <i>range</i>	<p>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.</p> <p>This is required to optimize the VXLAN resource allocation to scale the following platforms:</p> <ul style="list-style-type: none"> • 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. <p>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.</p> <p>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).</p>

Command	Description
suppress mac-route	<p>Suppresses the BGP MAC route so that BGP only sends the MAC/IP route for a host.</p> <p>Under NVE, the MAC updates for all VNIs are suppressed.</p> <p>Note</p> <ul style="list-style-type: none"> • Receive-side — Suppressing the MAC route depends upon the capability of the remote EVPN peer to derive a MAC route from the MAC/IP route (7.0(3)I2(2) and later). Avoid using the “suppress mac-route” command if devices in the network are running an earlier NX-OS release. • Send-side — Suppressing the MAC route means that the sender has a MAC/IP route. If your configuration has pure-Layer 2 VNIs (such as no corresponding VRF or Layer3-VNI), then there is no corresponding MAC/IP and you should avoid using the “suppress mac-route” command.

Configuring VXLAN BGP EVPN

Enabling VXLAN

Enable VXLAN and the EVPN.

Procedure

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

Configuring VLAN and VXLAN VNI

Procedure

	Command or Action	Purpose
Step 1	<code>vlan <i>number</i></code>	Specify VLAN.
Step 2	<code>vn-segment <i>number</i></code>	Map VLAN to VXLAN VNI to configure Layer 2 VNI under VXLAN VLAN.

Configuring VRF for VXLAN Routing

Configure the tenant VRF.

Procedure

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

	Command or Action	Purpose
Step 9	route-target both auto evpn	<p>Note Specifying the auto option is applicable only for IBGP.</p> <p>Manually configured route targets are required for EBGp.</p>

About RD Auto

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

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

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

Example auto-derived Route Distinguisher (RD)

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

About Route-Target Auto

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

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 SVI for Hosts for VXLAN Routing

Configure the SVI for hosts.

Procedure

	Command or Action	Purpose
Step 1	vlan <i>number</i>	Specify VLAN
Step 2	interface <i>vlan-number</i>	Specify VLAN interface.
Step 3	vrf member <i>vxlان-number</i>	Configure SVI for host.
Step 4	ip address <i>address</i>	Specify IP address.

Configuring VRF Overlay VLAN for VXLAN Routing

Procedure

	Command or Action	Purpose
Step 1	vlan <i>number</i>	Specify VLAN.
Step 2	vn-segment <i>number</i>	Specify vn-segment.

Configuring VNI Under VRF for VXLAN Routing

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

Procedure

	Command or Action	Purpose
Step 1	vrf context <i>vxlان</i>	Create a VXLAN Tenant VRF
Step 2	vni <i>number</i>	Configure Layer 3 VNI under VRF.

Configuring Anycast Gateway for VXLAN Routing

Procedure

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

Configuring the NVE Interface and VNIs

Procedure

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

Configuring BGP on the VTEP

Procedure

	Command or Action	Purpose
Step 1	router bgp <i>number</i>	Configure BGP.
Step 2	router-id <i>address</i>	Specify router address.
Step 3	neighbor <i>address</i> remote-as <i>number</i>	Define MP-BGP neighbors. Under each neighbor define l2vpn evpn.

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

Configuring RD and Route Targets for VXLAN Bridging

Procedure

	Command or Action	Purpose
Step 1	evpn	Configure VRF.
Step 2	vni number l2	Note Only Layer 2 VNIs need to be specified.
Step 3	rd auto	Define VRF RD (route distinguisher) to configure VRF context.
Step 4	route-target import auto	Define VRF Route Target and import policies.

	Command or Action	Purpose
Step 5	route-target export auto	Define VRF Route Target and export policies.

About RD Auto

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

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

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

Example auto-derived Route Distinguisher (RD)

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

About Route-Target Auto

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

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.



Note VXLAN EVPN ingress replication is supported on:

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

Before you begin

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

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

Procedure

	Command or Action	Purpose
Step 1	interface <i>nve-interface</i>	Configure the NVE interface.
Step 2	host-reachability protocol bgp	This defines BGP as the mechanism for host reachability advertisement
Step 3	member vni <i>vni</i> associate-vrf	Add Layer-3 VNIs, one per tenant VRF, to the overlay. Note Required for VXLAN routing only.
Step 4	member vni <i>vni</i>	Add Layer 2 VNIs to the tunnel interface.
Step 5	ingress-replication protocol bgp	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.

	Command or Action	Purpose
		Note Using ingress-replication protocol bgp avoids the need for any multicast configurations that might have been required for configuring the underlay.

Configuring BGP for EVPN on the Spine

Procedure

	Command or Action	Purpose
Step 1	route-map permitall permit 10	Configure route-map. Note The route-map keeps the next-hop unchanged for EVPN routes. <ul style="list-style-type: none"> • Required for eBGP. • Optional for iBGP.
Step 2	set ip next-hop unchanged	Set next-hop address. Note The route-map keeps the next-hop unchanged for EVPN routes. <ul style="list-style-type: none"> • Required for eBGP. • Optional for iBGP. Note When two next hops are enabled, next hop ordering is not maintained. If one of the next hops is a VXLAN next hop and the other next hop is local reachable via FIB/AM/Hmm, the local next hop reachable via FIB/AM/Hmm is always taken irrespective of the order. Directly/locally connected next hops are always given priority over remotely connected next hops.
Step 3	router bgp <i>autonomous system number</i>	Specify BGP.

	Command or Action	Purpose
Step 4	address-family l2vpn evpn	Configure address family Layer 2 VPN EVPN under the BGP neighbor.
Step 5	retain route-target all	Configure retain route-target all under address-family Layer 2 VPN EVPN [global]. Note Required for eBGP. Allows the spine to retain and advertise all EVPN routes when there are no local VNI configured with matching import route targets.
Step 6	neighbor address remote-as number	Define neighbor.
Step 7	address-family l2vpn evpn	Configure address family Layer 2 VPN EVPN under the BGP neighbor.
Step 8	disable-peer-as-check	Disables checking the peer AS number during route advertisement. Configure this parameter on the spine for eBGP when all leafs are using the same AS but the spines have a different AS than leafs. Note Required for eBGP.
Step 9	send-community extended	Configures community for BGP neighbors.
Step 10	route-map permitall out	Applies route-map to keep the next-hop unchanged. Note Required for eBGP.

Suppressing ARP

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



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

	Command or Action	Purpose
Step 1	hardware access-list tcam region arp-ether size double-wide	Configure TCAM region to suppress ARP.

	Command or Action	Purpose
		<p><i>tcam-size</i>—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.</p> <p>Note Reload is required for the TCAM configuration to be in effect.</p> <p>Note Configuring the hardware access-list tcam region arp-ether size double-wide is not required on Cisco Nexus 9200 Series switches.</p>
Step 2	interface nve 1	Create the network virtualization endpoint (NVE) interface.
Step 3	member vni <i>vni-id</i>	Specify VNI ID.
Step 4	suppress-arp	Configure to suppress ARP under Layer 2 VNI.
Step 5	copy running-config start-up-config	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Disabling VXLANs

Procedure

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

Duplicate Detection for IP and MAC Addresses

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

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

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

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

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

Command	Description
switch(config)# fabric forwarding ? anycast-gateway-mac dup-host-ip-addr-detection	Available sub-commands: <ul style="list-style-type: none"> • Anycast gateway MAC of the switch. • To detect duplicate host addresses in n seconds.
switch(config)# fabric forwarding dup-host-ip-addr-detection ? <1-1000>	The number of host moves allowed in n seconds. The range is 1 to 1000 moves; default is 5 moves.
switch(config)# fabric forwarding dup-host-ip-addr-detection 100 ? <2-36000>	The duplicate detection timeout in seconds for the number of host moves. The range is 2 to 36000 seconds; default is 180 seconds.

Command	Description
switch(config)# fabric forwarding dup-host-ip-addr-detection 100 10	Detects duplicate host addresses (limited to 100 moves) in a period of 10 seconds.

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

Command	Description
switch(config)# l2rib dup-host-mac-detection ? <1-1000> default	Available sub-commands for L2RIB: <ul style="list-style-type: none"> • The number of host moves allowed in n seconds. The range is 1 to 1000 moves. • Default setting (5 moves in 180 in seconds).
switch(config)# l2rib dup-host-mac-detection 100 ? <2-36000>	The duplicate detection timeout in seconds for the number of host moves. The range is 2 to 36000 seconds; default is 180 seconds.
switch(config)# l2rib dup-host-mac-detection 100 10	Detects duplicate host addresses (limited to 100 moves) in a period of 10 seconds.

Enabling Nuage Controller Interoperability

The following steps enable Nuage controller interoperability.

Procedure

	Command or Action	Purpose
Step 1	nuage controller interop	Global command to enable interoperability mode.
Step 2	router bgp <i>number</i>	Configure BGP.
Step 3	address-family l2vpn evpn	Configure address family Layer 2 VPN EVPN under the BGP neighbor.
Step 4	advertise-system-mac	Enable Nuage interoperability mode for BGP.
Step 5	allow-vni-in-ethertag	Enable Nuage interoperability mode for BGP.
Step 6	route-map permitall permit 10	Configure route-map to permit all.
Step 7	router bgp <i>number</i>	Configure BGP.
Step 8	vrf <i>vrf-name</i>	Specify tenant VRF.

	Command or Action	Purpose
Step 9	address-family ipv4 unicast	Configure address family for IPv4.
Step 10	advertise l2vpn evpn	Enable advertising EVPN routes.
Step 11	redistribute hmm route-map permitall	Enables advertise host tenant routes as evpn type-5 routes for interoperability.

Example

The following is an example to enable Nuage controller interoperability:

```

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

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

/*** Advertise host tenant routes as evpn type-5 routes for interoperability. ***/
switch(config)# route-map permitall permit 10
switch(config)# router bgp 1001
switch(config-router)# vrf vni-491830
switch(config-router-vrf)# address-family ipv4 unicast
switch(config-router-vrf-af)# advertise l2vpn evpn
switch(config-router-vrf-af)# redistribute hmm route-map permitall

```

Verifying the VXLAN BGP EVPN Configuration

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

Command	Purpose
show nve vrf	Displays VRFs and associated VNIs
show bgp l2vpn evpn	Displays routing table information.
show ip arp suppression-cache [detail summary vlan <i>vlan</i> statistics]	Displays ARP suppression information.
show vxlan interface	Displays VXLAN interface status.

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

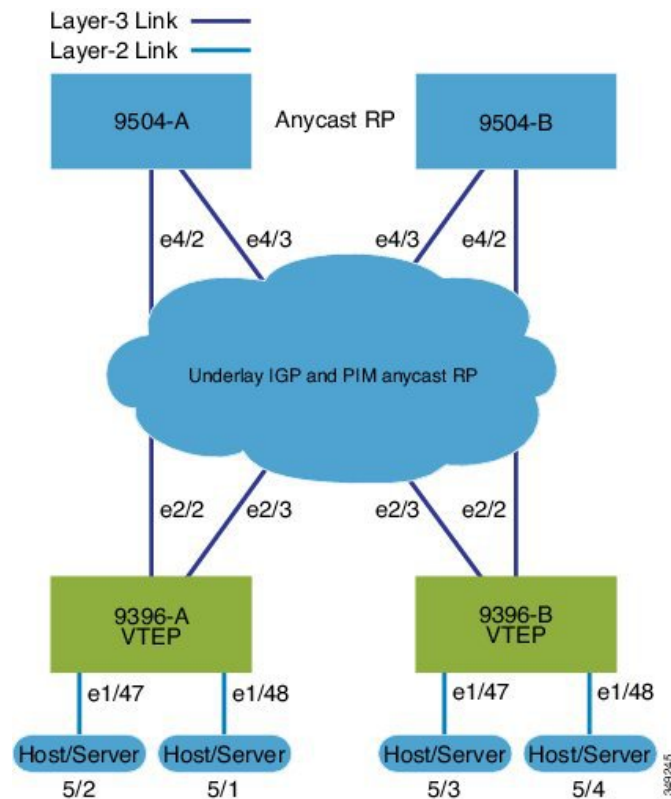


Note Although the **show ip bgp** command is available for verifying a BGP configuration, as a best practice, it is preferable to use the **show bgp** command instead.

Example of VXLAN BGP EVPN (EBGP)

An example of a VXLAN BGP EVPN (EBGP):

Figure 15: VXLAN BGP EVPN Topology (EBGP)



EBGP between Spine and Leaf

• Spine (9504-A)

- Enable the EVPN control plane

```
nv overlay evpn
```

- Enable the relevant protocols

```
feature bgp
feature pim
```

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

- Configure the BGP underlay.

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

- Spine (9504-B)

- Enable the EVPN control plane and the relevant protocols

```
nv overlay evpn
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 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 192.168.4.42/24
  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

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

- Configure Loopback for Anycast RP

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

- Configure the BGP overlay for the EVPN address family.

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

- Configure the BGP underlay.


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

- Leaf (9396-A)

- Enable the EVPN control plane

```
nv overlay evpn
```

- Enable the relevant protocols

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

- 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 4:1:0:1::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 platform switches.

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

  advertise l2vpn evpn

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 evpn
route-target export 65535:101 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 platform 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
```

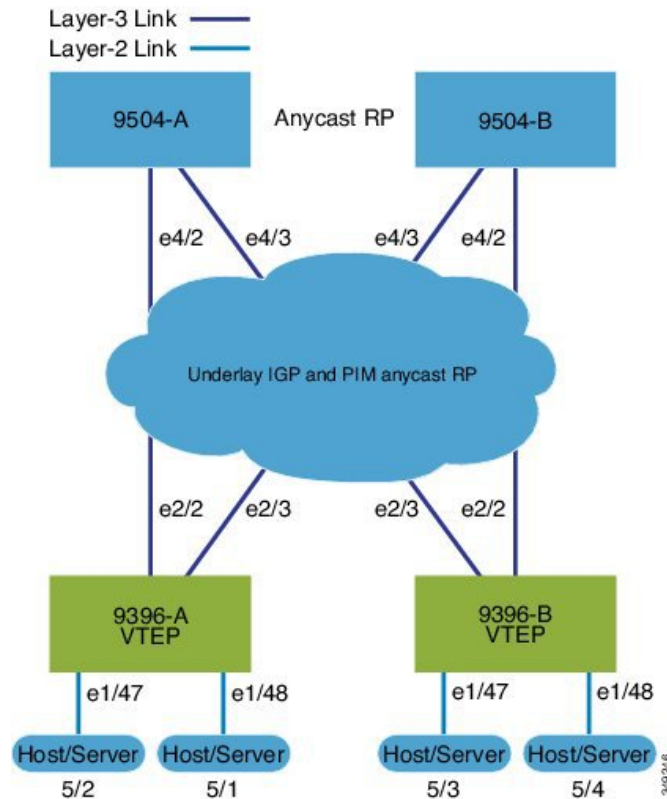
- Configure BGP

```
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 16: VXLAN BGP EVPN Topology (IBGP)



IBGP between Spine and Leaf

• Spine (9504-A)

- Enable the EVPN control plane

```
nv overlay evpn
```

- Enable the relevant protocols

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

- Configure Loopback for local VTEP IP, and BGP

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

- Configure Loopback for Anycast RP

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

- Configure Anycast RP

```
ip pim rp-address 100.1.1.1 group-list 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 platform switches.

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

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

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

```

Example Show Commands

• show nve peers

```

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

```

• show nve vni

```

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

```

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

• show ip arp suppression-cache detail

```

9396-B# show ip arp suppression-cache detail

```

Example Show Commands

```

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

```

Ip Address	Age	Mac Address	Vlan	Physical-ifindex	Flags
4.1.1.54	00:06:41	0054.0000.0000	1001	Ethernet1/48	L
4.1.1.51	00:20:33	0051.0000.0000	1001	(null)	R
4.2.2.53	00:06:41	0053.0000.0000	1002	Ethernet1/47	L
4.2.2.52	00:20:33	0052.0000.0000	1002	(null)	R

• show vxlan interface

```

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

```

• show bgp l2vpn evpn summary

```

9396-B# show bgp l2vpn evpn summary
BGP summary information for VRF default, address family L2VPN EVPN
BGP router identifier 40.1.1.1, local AS number 65535
BGP table version is 27, L2VPN EVPN config peers 2, capable peers 2
14 network entries and 18 paths using 2984 bytes of memory
BGP attribute entries [14/2240], BGP AS path entries [0/0]
BGP community entries [0/0], BGP clusterlist entries [2/8]

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

```

• show bgp l2vpn evpn

```

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

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

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

```



```

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

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

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

```

• show l2route evpn mac all

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

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

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

• show l2route evpn mac-ip all

9396-B# **show l2route evpn mac-ip all**

Flags -(Rmac):Router MAC (Stt):Static (L):Local (R):Remote (V):vPC link
 (Dup):Duplicate (Spl):Split (Rcv):Recv (D):Del Pending (S):Stale (C):Clear
 (Ps):Peer Sync (Ro):Re-Originated

Topology	Mac Address	Prod	Flags	Seq No	Host IP	Next-Hops
1001	d8b1.9071.e903	BGP	--	0	4.1.1.12	30.1.1.1
1001	f8c2.8890.2a45	HMM	--	0	4.1.1.122	Local
1002	d8b1.9071.e903	BGP	--	0	4.2.2.11	30.1.1.1
1002	f8c2.8890.2a45	HMM	--	0	4.2.2.111	Local



CHAPTER 5

Configuring VXLAN OAM

This chapter contains the following sections:

- [VXLAN OAM Overview, on page 121](#)
- [Loopback \(Ping\) Message, on page 122](#)
- [Traceroute or Pathtrace Message, on page 123](#)
- [Configuring VXLAN OAM, on page 125](#)
- [Configuring NGOAM Profile, on page 128](#)
- [NGOAM Authentication, on page 129](#)

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

Category	Tools
Fault Verification	Loopback Message

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

Loopback (Ping) Message

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

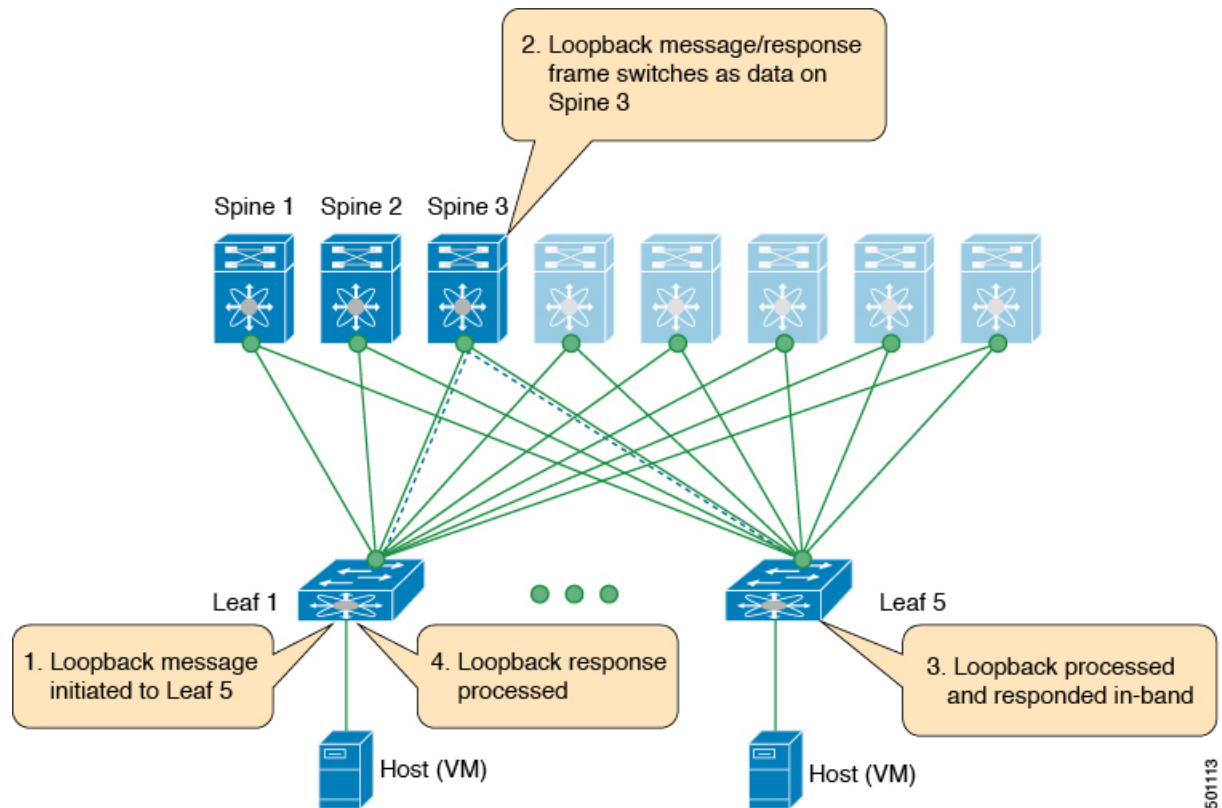
The loopback (ping) message can be destined to VM or to the (VTEP on) leaf switch. This ping message can use different OAM channels. If the ICMP channel is used, the loopback message can reach all the way to the VM if the VM's IP address is specified. If NVO3 draft Tissa channel is used, this loopback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

Ping

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

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

Figure 17: Loopback Message



501113

Traceroute or Pathtrace Message

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

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

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

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

Traceroute

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

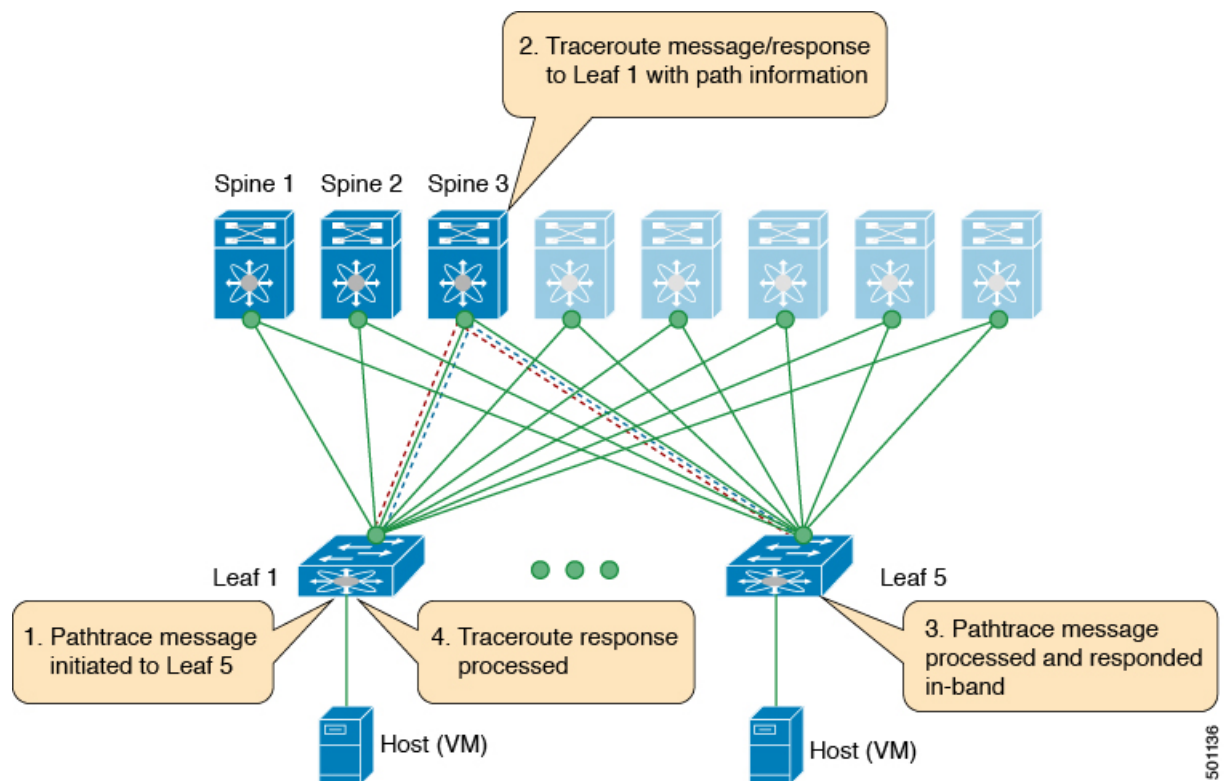
- Traceroute uses the ICMP packets (channel-1), encapsulated in the VXLAN encapsulation to reach the host

Pathtrace

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

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

Figure 18: Traceroute Message



501136

Configuring VXLAN OAM

Before you begin

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

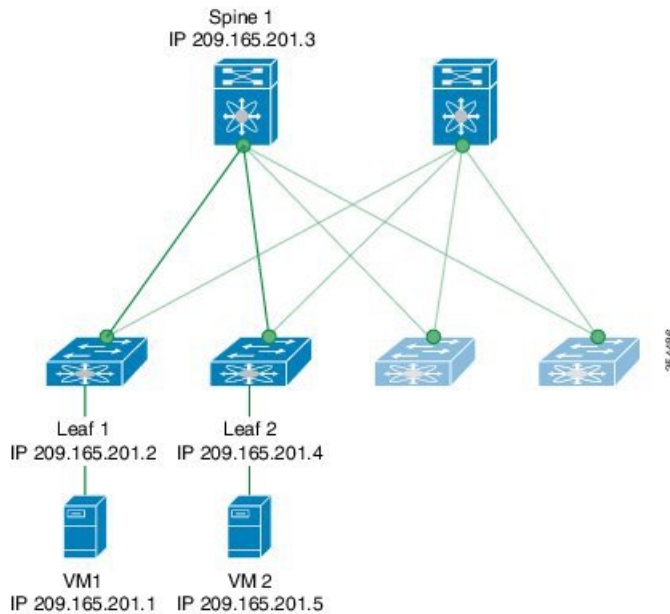
Procedure

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

Example

See the following examples of the configuration topology.

Figure 19: VXLAN Network



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

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

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

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

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



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

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

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



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

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

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

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

```
Path trace Request to peer ip 209.165.201.4 source ip 209.165.201.2
```

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

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

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

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

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

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

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

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

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

```
Path trace Request to peer ip 209.165.201.4 source ip 209.165.201.2
Sender handle: 46
TTL  Code  Reply                               IngressI/f    EgressI/f    State
=====
```

```

1 !Reply from 209.165.201.3, Eth5/5/1 Eth5/5/2 UP/UP
2 !Reply from 209.165.201.4, Eth1/3 Unknown UP/DOWN

```

Configuring NGOAM Profile

Complete the following steps to configure NGOAM profile.

Procedure

	Command or Action	Purpose
Step 1	switch(config)#[no] feature ngoam	Enables or disables NGOAM feature
Step 2	switch(config)#[no] ngoam profile <profile-id>	Configures OAM profile. The range for the profile-id is <1 – 1023>. This command does not have a default value. Enters the config-ngoam-profile submode to configure NGOAM specific commands. Note All profiles have default values and the show run all CLI command displays them. The default values are not visible through the show run CLI command.
Step 3	switch(config-ng-oam-profile)# ? Example: switch(config-ng-oam-profile)# ? description Configure description of the profile dot1q Encapsulation dot1q/bd flow Configure ngoam flow hop Configure ngoam hop count interface Configure ngoam egress interface no Negate a command or set its defaults oam-channel Oam-channel used payload Configure ngoam payload sport Configure ngoam Udp source port range	Displays the options for configuring NGOAM profile.

Example

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

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

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

NGOAM Authentication

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

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

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

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

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

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

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

```

bcast:10408 discards:0 errors:0 bandwidth:42949672970000000
  2 !Reply from 12.0.22.1, Eth1/7 Unknown UP / DOWN
    Input Stats: PktRate:0 ByteRate:0 Load:0 Bytes:4213416 unicast:275 mcast:4366 bcast:3
discards:0 errors:0 unknown:0 bandwidth:42949672970000000
switch# conf t
switch(config)# no ngoam authentication-key 123456789
switch(config)# end

```

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

```

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

```



CHAPTER 6

Configuring VXLAN EVPN Multihoming

This chapter contains the following sections:

- [VXLAN EVPN Multihoming Overview](#) , on page 131
- [Configuring VXLAN EVPN Multihoming](#), on page 135
- [Configuring Layer 2 Gateway STP](#), on page 137
- [Configuring VXLAN EVPN Multihoming Traffic Flows](#), on page 141
- [Configuring VLAN Consistency Checking](#), on page 153
- [Configuring ESI ARP Suppression](#), on page 156

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 an Network Layer Reachability Information (NLRI) with no MAC addresses in the route key.

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

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

To summarize the new implementation concepts for the EVPN multihoming:

- EAD/ES: Ethernet Auto Discovery Route per ES that is also referred to as type-1 route. This route is used to converge the traffic faster during access failure scenarios. This route has Ethernet Tag of 0xFFFFFFFF.

- EAD/EVI: Ethernet Auto Discovery Route per EVI that is also referred to as type-1 route. This route is used for aliasing and load balancing when the traffic only hashes to one of the switches. This route cannot have Ethernet Tag value of 0xFFFFFFFF 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.

- EVPN 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.
- EVPN Multihoming requires that all switches in a given network must be EVPN Multihoming capable.. Mixing platforms with and without EVPN Multihoming is not supported.
- EVPN multihoming is not supported on FEX.
- Beginning with Cisco NX-OS Release 7.0(3)I5(2), ARP suppression is supported with EVPN multihoming.
- EVPN Multihoming is supported with multihoming to two switches only.
- To enable EVPN 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

	Command or Action	Purpose
Step 1	evpn esi multihoming	Enables EVPN multihoming globally.
Step 2	address-family l2vpn evpn maximum-paths <>maximum-paths ibgp <> Example: <pre>address-family l2vpn evpn maximum-paths 64 maximum-paths ibgp 64</pre>	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.
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 <> Example:	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.

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

VXLAN EVPN Multihoming Configuration Examples

See the sample VXLAN EVPN multihoming configuration on the switches:

Switch 1 (L1)

```
evpn esi multihoming
```

```

router bgp 1001
  address-family l2vpn evpn
  maximum-paths ibgp 2

```

```

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

```

```

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

```

```

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

```

Switch 2 (L2)

```
evpn esi multihoming
```

```

router bgp 1001
  address-family l2vpn evpn
  maximum-paths ibgp 2

```

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

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

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

Configuring Layer 2 Gateway STP

Layer 2 Gateway STP Overview

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

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

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

Guidelines for Moving to Layer 2 Gateway STP

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

- With Layer 2 Gateway STP, root guard is enabled by default on all the access ports.
- With Layer 2 Gateway STP enabled, the VXLAN fabric (all VTEPs) emulates a single pseudo-root switch for the customer access switches.
- All access ports from VTEPs connecting to the customer access switches are in the **Desg FWD** state by default.
- All ports on customer access switches connecting to VTEPs are either in the root-port FWD or Altn BLK state.
- Root guard is activated if superior spanning-tree information is received from the customer access switches. This process puts the ports in **BLK L2GW_Inc** state to secure the root on the VXLAN fabric and prevent a loop.
- Explicit domain ID configuration is needed to enable spanning-tree BPDU tunneling across the fabric.
- As a best practice, you should configure all VTEPs with the lowest spanning-tree priority of all switches in the spanning-tree domain to which they are attached. By setting all the VTEPs as the root bridge, the entire VXLAN fabric appears to be one virtual bridge.
- ESI interfaces should not be enabled in spanning-tree edge mode to allow Layer 2 Gateway STP to run across the VTEP and access layer.
- You can continue to use ESIs or orphans (single-homed hosts) in spanning-tree edge mode if they directly connect to hosts or servers that do not run Spanning Tree Protocol and are end hosts.
- Configure all VTEPs that are connected by a common customer access layer in the same Layer 2 Gateway STP domain. Ideally, all VTEPs on the fabric on which the hosts reside and to which the hosts can move.
- The Layer 2 Gateway STP domain scope is global, and all ESIs on a given VTEP can participate in only one domain.
- Mappings between Multiple Spanning Tree (MST) instances and VLANs must be consistent across the VTEPs in a given Layer 2 Gateway STP domain.
- Ensure that the root of an STP domain local to the VXLAN fabric is a VTEP or placed within the fabric.
- Non-Layer 2 Gateway STP enabled VTEPs cannot be directly connected to Layer 2 Gateway STP-enabled VTEPs. Performing this action results in conflicts and disputes because the non-Layer 2 Gateway STP VTEP keeps sending BPDUs and it can steer the root outside.
- Keep the current edge and the BPDU filter configurations on both the Cisco Nexus switches and the access switches after upgrading to the latest build.
- Enable Layer 2 Gateway STP on all the switches with a recommended priority and the *mst* instance mapping as needed. Use the commands **spanning-tree domain enable** and **spanning-tree mst <instance-id's> priority 8192**.
- Remove the BPDU filter configurations on the switch side first.
- Remove the BPDU filter configurations and the edge on the customer access switch.

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

Enabling Layer 2 Gateway STP on a Switch

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

Procedure

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

Example

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

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

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

```

switch# show spanning-tree vlan 1001

MST0000
  Spanning tree enabled protocol mstp

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

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

```

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

```

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

```

```

switch# show spanning-tree

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

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

```

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

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

```

switch(config)# spanning-tree domain disable

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

```

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

```

switch# show spanning-tree vlan 1001

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

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

```

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

Configuring VXLAN EVPN Multihoming Traffic Flows

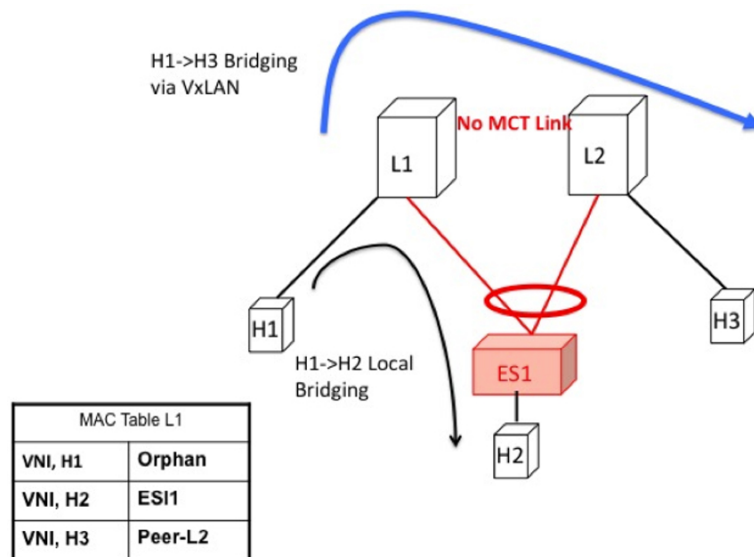
EVPN Multihoming Local Traffic Flows

All switches that are a part of the same redundancy group (as defined by the ESI) act as a single virtual switch with respect to the access switch/host. However, there is no 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.

Figure 20: Local Bridging at L1. H1->H3 bridging via VXLAN. In vPC, H1->H3 will be via Peer Link.



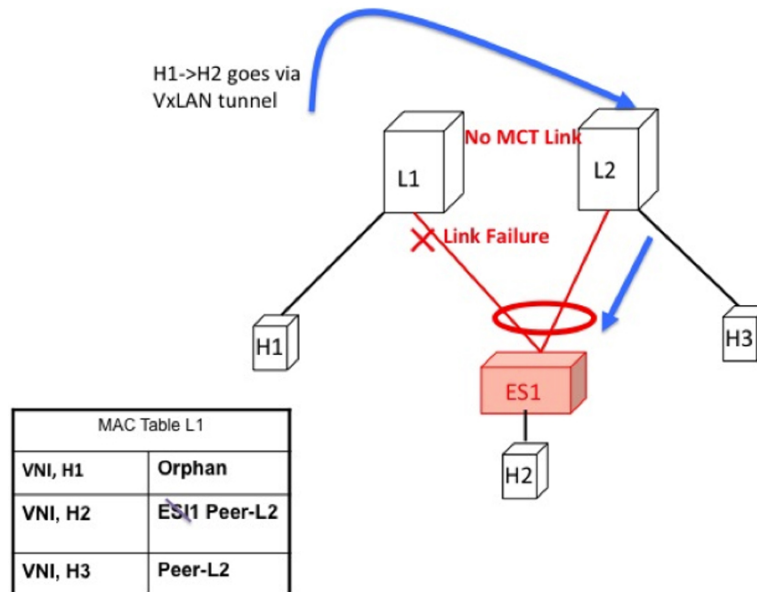
Access Failure for Locally Bridged Traffic

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



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

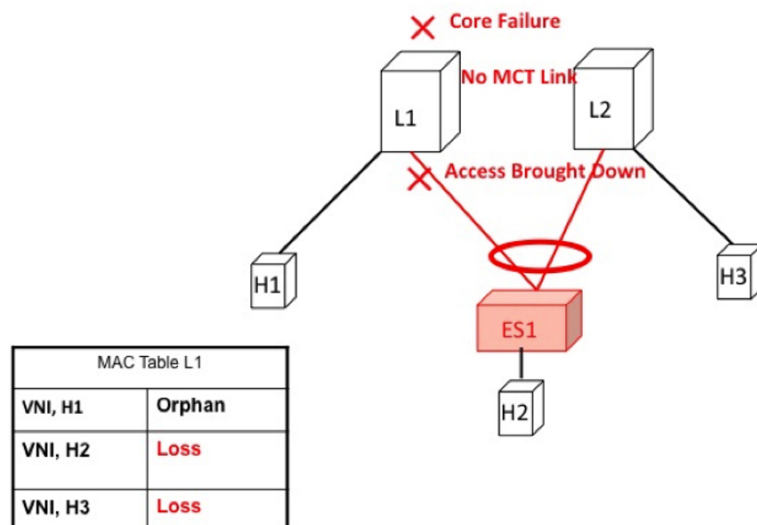
Figure 21: ES1 failure on L1. H1->H2 is now bridged over VXLAN tunnel.



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.

Figure 22: Core failure on L1. H1->H2 loses all connectivity as there is no Peer Link.



Locally Routed Traffic

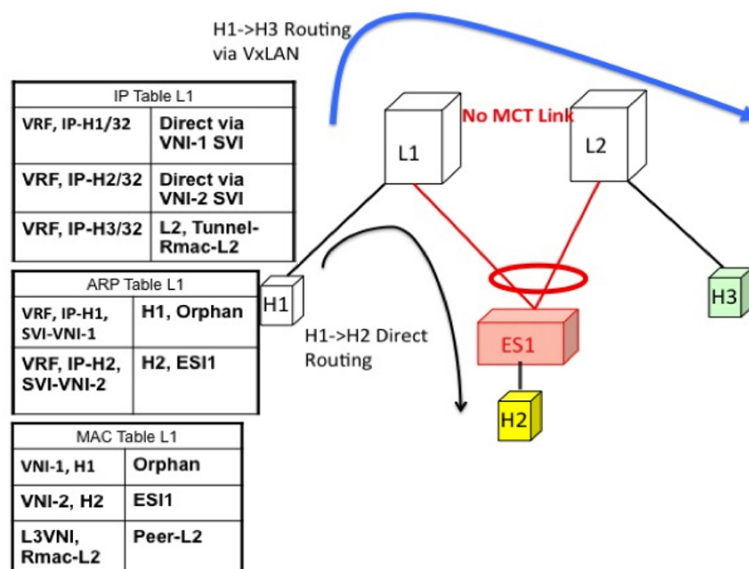
Consider H1, H2, and H3 being in different subnets and L1/L2 being distributed anycast gateways.

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.

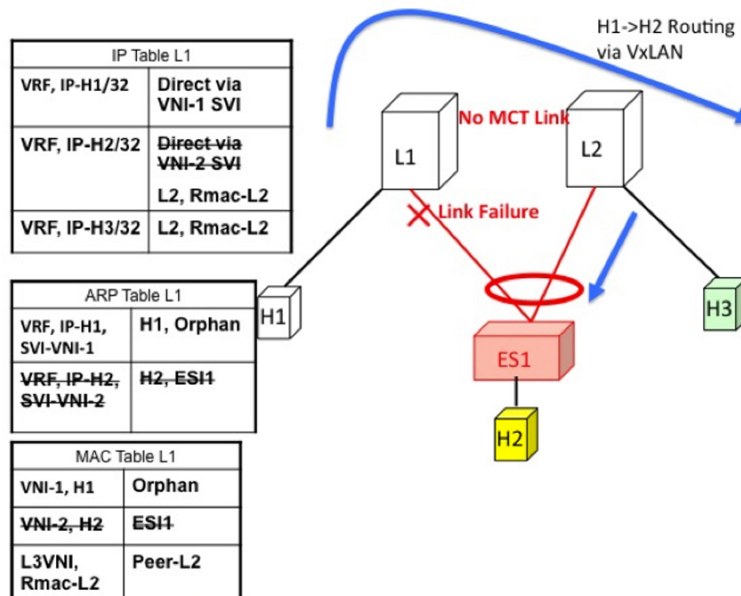
Figure 23: L1 is Distributed Anycast Gateway. H1, H2, and H3 are in different VLANs. H1->H3 routing happens via VXLAN tunnel encapsulation. In vPC, H3 ARP would have been synced via Peer Link and direct routing.



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.

Figure 24: H1, H2, and H3 are in different VLANs. ES1 fails on L1. H1->H2 routing happens via VXLAN tunnel encapsulation.

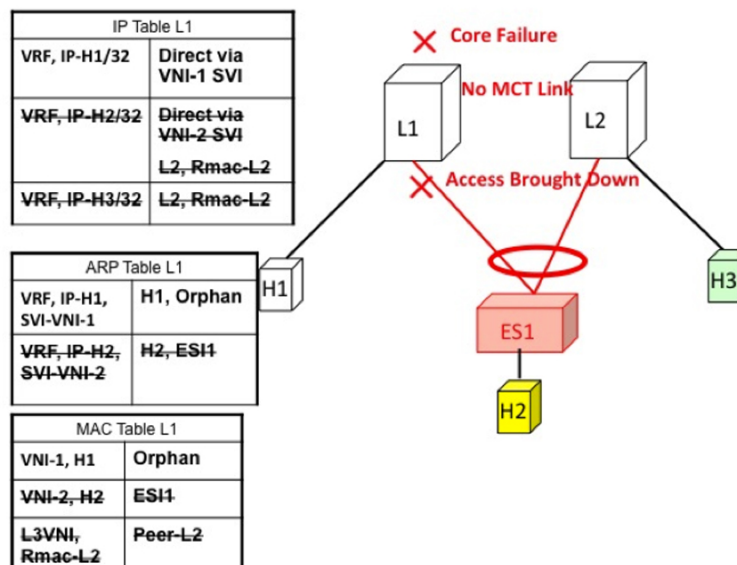


Core Failure for Locally Routed Traffic

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

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

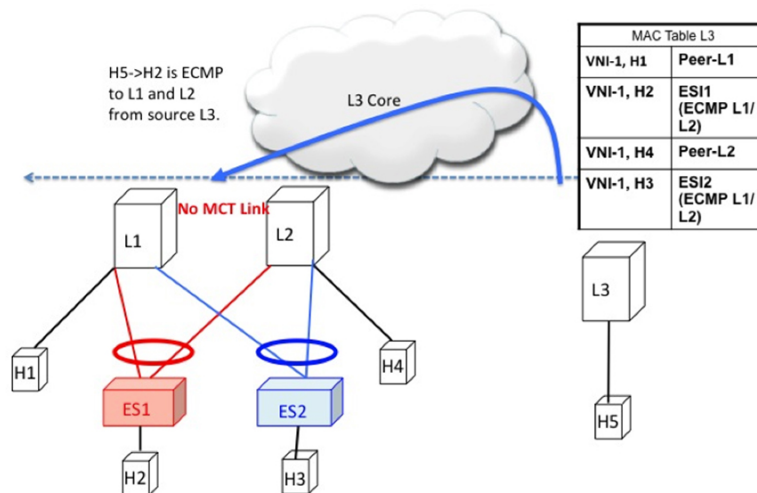
Figure 25: H1, H2, and H3 are in different VLANs. Core fails on L1. Access is brought down. H1 loses all connectivity.



EVPN Multihoming Remote Traffic Flows

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

Figure 26: Layer 2 VXLAN Gateway. L3 performs MAC ECMP to L1/L2.



Remote Bridged Traffic

Consider a remote host H5 that wants to bridge traffic to host H2 that is positioned behind the EVPN MH Complex (L1, L2). Host H2 builds an ECMP list in accordance to the rules defined in RFC 7432. The MAC table at switch 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		
NLRI	Route Type	Ethernet Segment (Type 1)
	Route Distinguisher	Router-ID: Segment-ID (VNID << 8)
	ESI	<Type: 1B><MAC: 6B><LD: 3B>
	Ethernet Tag	MAX-ET
	MPLS Label	0
ATTRS	ESI Label Extended Community ESI Label = 0	Single Active = False
	Next-Hop	NVE Loopback IP
	Route Target	Subset of List of RTs of MAC-VRFs associated to all the EVIs active on the ES

MAC-IP Route (Type 2)

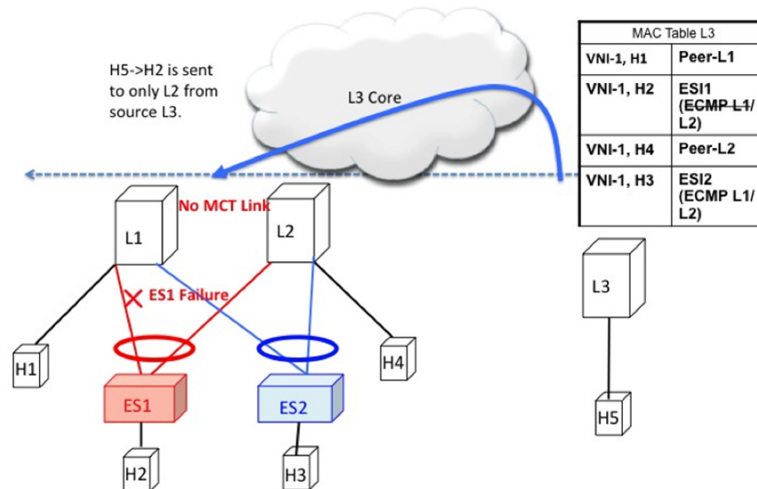
MAC-IP Route remains the same as used in the current vPC multihoming and NX-OS single-homing solutions. However, now it has a non-zero ESI field that indicates that this is a multihomed host and it is a candidate for ECMP Path Resolution.

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

Access Failure for Remote Bridged Traffic

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

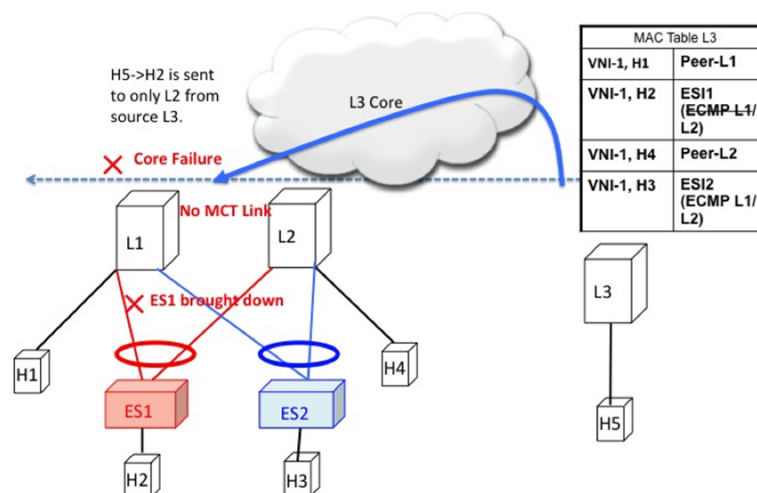
Figure 27: Layer 2 VXLAN Gateway. ESI failure on L1. L3 withdraws L1 from MAC ECMP list. This will happen due to EAD/ES mass withdrawal from L1.



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 28: 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.

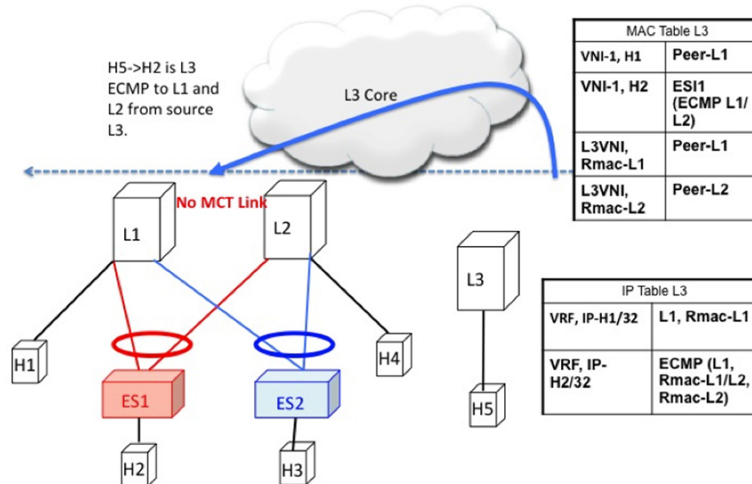


Remote Routed Traffic

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



Access Failure for Remote Routed Traffic

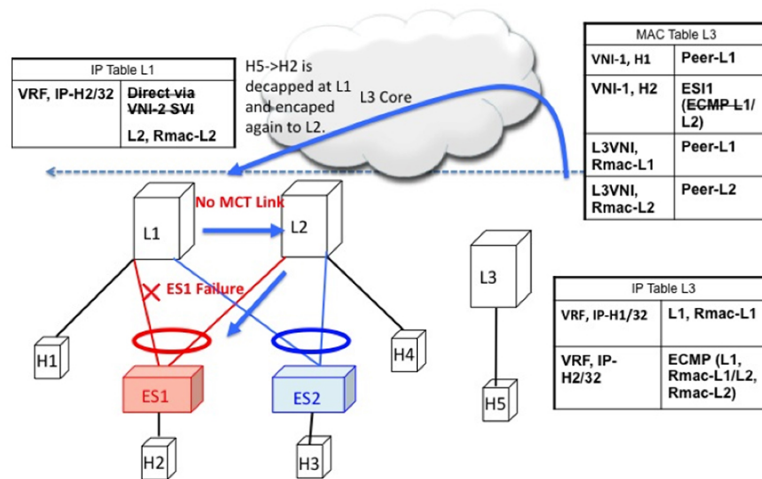
If the access link pointing to ES1 goes down on L1, the mass withdrawal route is sent in the form of EAD/ES and that causes L3 to remove L1 from the MAC ECMP PathList, leading the intra-subnet (L2) traffic to converge quickly. L1 now treats H2 as a remote route reachable via VXLAN Overlay as it is no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

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

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

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

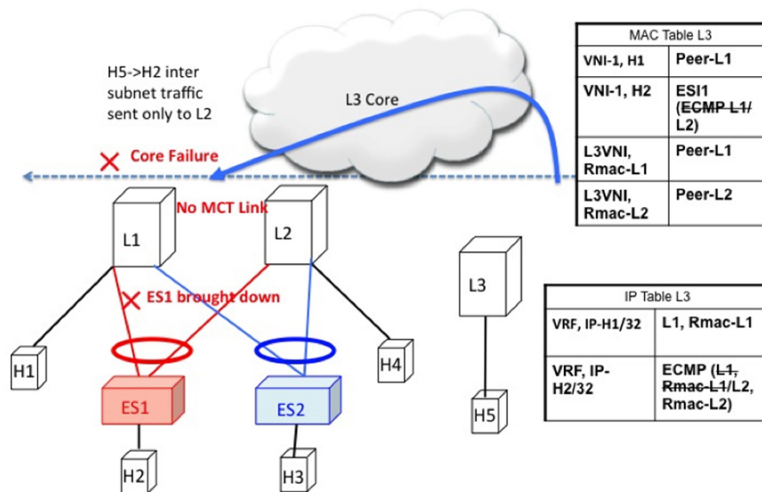
Figure 30: Layer 3 VXLAN Gateway. ESI failure causes ES mass withdrawal that only impacts L2 ECMP. L3 ECMP continues until Type2 is withdrawn. L3 traffic reaches H2 via suboptimal path L3->L1->L2 until then.



Core Failure for Remote Routed Traffic

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

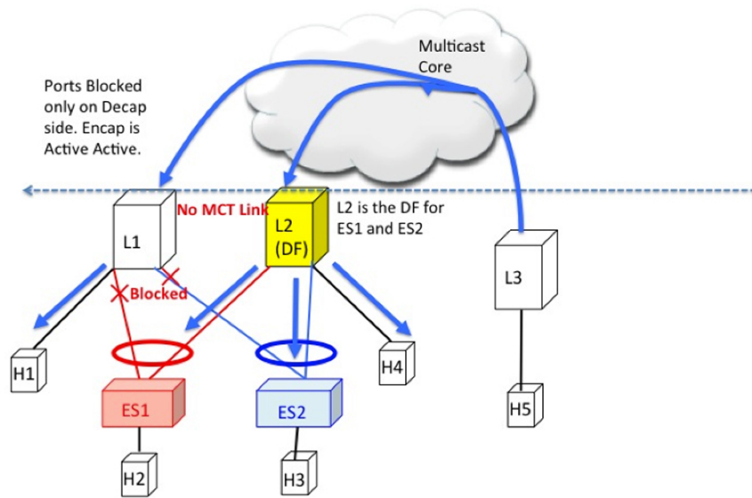
Figure 31: Layer 3 VXLAN Gateway. Core failure. All L3 ECMP paths to L1 are withdrawn at L3 due to route reachability going away.



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.

Figure 32: BUM traffic originating at L3. L2 is the DF for ES1 and ES2. L2 decapsulates and forwards to ES1, ES2 and orphan. L1 decapsulates and only forwards to orphan.



Designated Forwarder

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

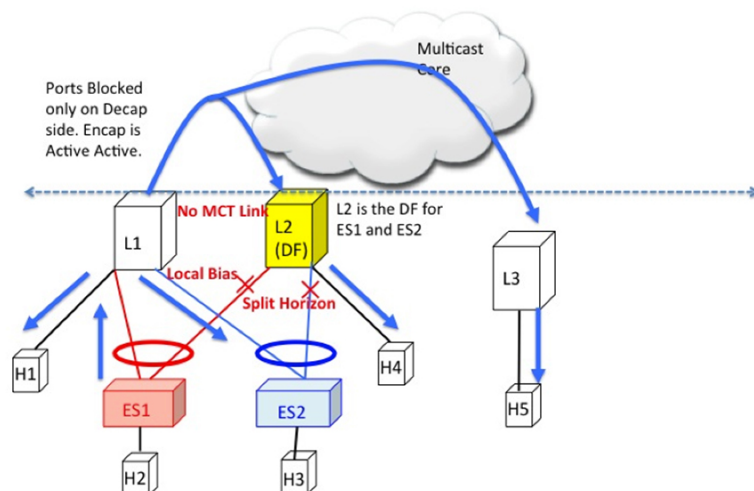
- DF Election is per (ES, VLAN) basis. There can be a different DF for ES1 and ES2 for a given VLAN.
- DF election result only applies to BUM traffic on the RX side for decap.
- Every switch must decap BUM traffic to forward it to singly homed or orphan links.
- Duplication of DF role leads to duplicate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

Split Horizon and Local Bias

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

Remote switches decap and forward it to their ESI and orphan links based on the DF state. However, this packet is also received at L2 that belongs to the same redundancy group as the originating switch L1. L2 must decap the packet to send it to orphan ports. However, even though 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 33: BUM traffic originating at L1. L2 is the DF for ES1 and ES2. However, L2 must perform split horizon check here as it shares ES1 and ES2 with L1. L2 however



Ethernet Segment Route (Type 4)

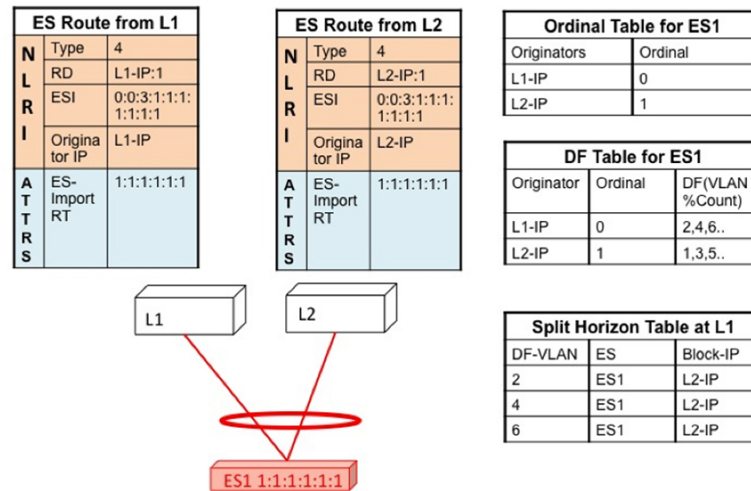
The Ethernet Segment Route is used to elect the Designated Forwarder and to apply Split Horizon Filtering. All the switches that are configured with an Ethernet Segment originate from this route. Ethernet Segment Route is exported and imported when ESI is locally configured under the PC.

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

DF Election and VLAN Carving

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

Figure 34: 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 **cfs ipv4 mcast-address** <mcast address> CLI command. To enable CFSoIP, the **cfs ipv4 distribute** CLI command should be used.

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

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

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

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

VLAN consistency checking runs for the following events:

- Global VLAN configuration: Add, delete, shut, or no shut events.
- Port channel VLAN configuration: Trunk allowed VLANs added or removed or access VLAN changed.
- CFS events: CFS peer added or deleted or CFSv4 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 CFSv4. 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.
- CFSv4 and CFSv6 should not be used in the same device.
- CFSv4 should not be used in devices that are not used for VLAN consistency checking.
- If CFSv4 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 CFSv4.

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
```

```
switch# sh run | i vlan-consistency
evpn esi multihoming
vlan-consistency-check
```

Displaying Show command Output for VLAN Consistency Checking

See the following show commands output for VLAN consistency checking.

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

```
switch# sh cfs peers name nve

Scope      : Physical-ip
-----
Switch WWN      IP Address
-----
20:00:f8:c2:88:23:19:47 172.31.202.228 [Local]
                        Switch
20:00:f8:c2:88:90:c6:21 172.31.201.172 [Not Merged]
20:00:f8:c2:88:23:22:8f 172.31.203.38  [Not Merged]
20:00:f8:c2:88:23:1d:e1 172.31.150.132 [Not Merged]
20:00:f8:c2:88:23:1b:37 172.31.202.233 [Not Merged]
20:00:f8:c2:88:23:05:1d 172.31.150.134 [Not Merged]
```

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

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

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

See the following Syslog output:

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

After Fixing configuration

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

Configuring ESI ARP Suppression

Overview of ESI ARP Suppression

Ethernet Segment Identifier (ESI) ARP suppression is an extension of the ARP suppression solution in VXLAN EVPN. It optimizes the ESI multihoming feature by significantly decreasing ARP broadcasts in the data center.

The host normally floods the VLAN with ARP requests. You can minimize this flooding by maintaining an ARP cache locally on the leaf switch. The ARP cache is built by:

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

With ESI ARP suppression, the initial ARP requests are broadcast to all sites. However, subsequent ARP requests are suppressed at the first-hop leaf switch and answered locally if possible. In this way, ESI ARP suppression significantly reduces ARP traffic across the overlay. If the cache lookup fails and the response cannot be generated locally, the ARP request can be flooded, which helps with the detection of silent hosts.

ESI ARP suppression is a per-VNI (L2 VNI) feature and is supported only with VXLAN EVPN (distributed gateway). This feature is supported in L2 (no SVI) and L3 modes for releases prior to Cisco NX-OS Release 7.0(3)I5(2) but only in L3 mode for later releases.

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 team region arp-ether 256** CLI command.

```
Interface nve1
no shutdown
source-interface loopback1
host-reachability protocol bgp
```

```

member vni 10000
  suppress-arp
mcast-group 224.1.1.10

```

Displaying Show Commands for ESI ARP Suppression

See the following Show commands output for ESI ARP suppression:

```

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

```

```
switch# show ip arp suppression-cache local
```

```

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

```

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

```
switch# show ip arp suppression-cache remote
```

```

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

```

Remote Vtep Adrrs	Ip Address	Age	Mac Address	Vlan	Physical-ifindex	Flags
61.1.1.40	00:48:37	0000.0610.0040	610	(null)		R

VTEP1, VTEP2.. VTEPn

```
switch# show ip arp suppression-cache detail
```

```

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

```

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

VTEP1, VTEP2.. VTEPn

```

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

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

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

switch#

```




CHAPTER 7

Configuring VIP/PIP

This chapter contains the following sections:

- [Advertising Primary IP Address, on page 159](#)
- [BorderPE Switches in a vPC Setup, on page 160](#)
- [DHCP Configuration in a vPC Setup, on page 160](#)
- [IP Prefix Advertisement in vPC Setup, on page 160](#)

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:

```
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 l2vpn evpn
    advertise-pip
  interface nve 1
    advertise virtual-rmac
```

IP Prefix Advertisement in vPC Setup

There are 3 types of Layer-3 routes that can be advertised by BGP EVPN. They are:

- Local host routes—These routes are learned from the attached servers or hosts.
- Prefix routes—These routes are learned via other routing protocol at the leaf, border leaf and border spine switches.
- Leaf switch generated routes—These routes include interface routes and static routes.

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**.

When enabling PIP or VIP you need to perform a shut/no shut on the NVE interface so that there is a NVE interface flap. This will avoid advertising unknown IP address to VTEPS in a spine-leaf topology.

The following is a sample configuration:

```
switch(config)# router bgp 100
  address-family l2vpn 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 163](#)
- [Guidelines and Limitations for VXLAN EVPN Multi-Site, on page 164](#)
- [Enabling VXLAN EVPN Multi-Site, on page 165](#)
- [Configuring VNI Dual Mode, on page 166](#)
- [Configuring Fabric/DCI Link Tracking, on page 167](#)
- [Configuring Fabric External Neighbors, on page 168](#)

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-originate 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.

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)



Note 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.
- PIM is not supported on multisite VXLAN DCI links.
- iBGP EVPN Peering between border gateways of different fabrics/sites is not supported.
- The **peer-type fabric-external** command configuration is required only for VXLAN Multi-site BGWs (this command must not be used when peering with non-Cisco equipment).



Note The **peer-type fabric-external** command configuration is not required for pseudo BGWs.

- If different Anycast Gateway MAC addresses are configured across sites, ARP suppression must be enabled for all VLANs that have been extended.
- Bind NVE to a loopback address that is separate from loopback addresses that are required by Layer 3 protocols. A best practice is to use a dedicated loopback address for the NVE source interface (PIP VTEP) and Multi-Site source interface (anycast and virtual IP VTEP).

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

	Command or Action	Purpose
Step 1	configure terminal Example: <code>switch# configure terminal</code>	Enters global configuration mode.
Step 2	evpn multisite border-gateway <i>ms-id</i> Example: <code>switch(config)# evpn multisite border-gateway 100</code>	Configure the site ID for a site/fabric. The range of values for <i>ms-id</i> is 1 to 2,814,749,767,110,655. The <i>ms-id</i> must be the same in all border gateways within the same fabric/site.
Step 3	interface nve 1 Example: <code>switch(config-evpn-msite-bgw)# interface nve 1</code>	Creates a VXLAN overlay interface that terminates VXLAN tunnels. Note Only 1 NVE interface is allowed on the switch
Step 4	source-interface loopback <i>src-if</i> Example: <code>switch(config-if-nve)# source-interface loopback 0</code>	The source interface must be a loopback interface that is configured on the switch with a valid /32 IP address. This /32 IP address must be known by the transient devices in the transport network and the remote VTEPs. This is accomplished by advertising it through a dynamic routing protocol in the transport network.
Step 5	host-reachability protocol bgp Example:	Defines BGP as the mechanism for host reachability advertisement.

	Command or Action	Purpose
	<code>switch(config-if-nve)# host-reachability protocol bgp</code>	
Step 6	multisite border-gateway interface loopback <i>vi-num</i> Example: <code>switch(config-if-nve)# multisite border-gateway interface loopback 100</code>	Defines the loopback interface used for the border gateway virtual IP address (VIP). The border-gateway interface must be a loopback interface that is configured on the switch with a valid /32 IP address. This /32 IP address must be known by the transient devices in the transport network and the remote VTEPs. This is accomplished by advertising it through a dynamic routing protocol in the transport network. This loopback must be different than the source interface loopback. The range of <i>vi-num</i> is from 0 to 1023.
Step 7	no shutdown Example: <code>switch(config-if-nve)# no shutdown</code>	Negate shutdown command.
Step 8	exit Example: <code>switch(config-if-nve)# exit</code>	Exits the NVE configuration mode.
Step 9	interface loopback <i>loopback_number</i> Example: <code>switch(config)# interface loopback 0</code>	Configure the loopback interface.
Step 10	ip address <i>ip-address</i> Example: <code>switch(config-if)# ip address 198.0.2.0/32</code>	Configures the IP address for the loopback interface.

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.



Note If you have multiple VRFs and only one is extended to ALL leaf switches, you can add a dummy loopback to that one extended VRF and advertise through BGP. Otherwise, you'll need to check how many VRFs are extended and to which switches, and then add a dummy loopback to the respective VRFs and advertise them as well. Therefore, use the **advertise-pip** command to prevent potential user errors in the future.

For more information about configuring the mcast-group (or ingress-replication protocol bgp) for a large number of VNIs, see [Example of VXLAN BGP EVPN \(EBGP\), on page 99](#).

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: switch# configure terminal	Enters global configuration mode.
Step 2	interface nve 1 Example: switch(config)# interface nve 1	Creates a VXLAN overlay interface that terminates VXLAN tunnels. Note Only one NVE interface is allowed on the switch.
Step 3	member vni vni-range Example: switch(config-if-nve)# member vni 200	Configure the virtual network identifier (VNI). The range for <i>vni-range</i> is from 1 to 16,777,214. The value of <i>vni-range</i> can be a single value like 5000 or a range like 5001-5008. Note Enter one of the Step 4 or Step 5 commands.
Step 4	mcast-group ip-addr Example: switch(config-if-nve-vni)# mcast-group 255.0.4.1	Configure the NVE Multicast group IP prefix within the fabric.
Step 5	ingress-replication protocol bgp Example: switch(config-if-nve-vni)# ingress-replication protocol bgp	Enables BGP EVPN with ingress replication for the VNI within the fabric.
Step 6	multisite ingress-replication Example: switch(config-if-nve-vni)# multisite ingress-replication	Defines the Multi-Site BUM replication method. Per-VNI knob for extending Layer 2 VNI.

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

	Command or Action	Purpose
Step 1	configure terminal Example:	Enters global configuration mode.

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

Configuring Fabric External Neighbors

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

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: <code>switch# configure terminal</code>	Enters global configuration mode.
Step 2	router bgp <i>as-num</i> Example:	Configure the autonomous system number. The range for <i>as-num</i> is from 1 to 4,294,967,295.

	Command or Action	Purpose
	<code>switch(config)# router bgp 100</code>	
Step 3	neighbor <i>ip-addr</i> Example: <code>switch(config-router)# neighbor 100.0.0.1</code>	Configure a BGP neighbor.
Step 4	peer-type fabric-external Example: <code>switch(config-router-neighbor)# peer-type fabric-external</code>	<p>Enables the next hop rewrite for multi-site. Defines site external BGP neighbors for EVPN exchange. The default for peer-type is fabric-internal.</p> <p>Note The peer-type fabric-external command is required only for VXLAN Multi-Site border gateways. It is not required for pseudo border gateways.</p>
Step 5	address-family l2vpn evpn Example: <code>switch(config-router-neighbor)# address-family l2vpn evpn</code>	
Step 6	rewrite-evpn-rt-asn Example: <code>switch(config-router-neighbor)# rewrite-evpn-rt-asn</code>	<p>Rewrites the route target (RT) information to simplify the MAC-VRF and IP-VRF configuration. BGP receives a route, and as it processes the RT attributes, it checks if the AS value matches the peer AS that is sending that route and replaces it. Specifically, this command changes the incoming route target's AS number to match the BGP-configured neighbor's remote AS number. You can see the modified RT value in the receiver router.</p>



CHAPTER 9

Configuring Tenant Routed Multicast

This chapter contains the following sections:

- [About Tenant Routed Multicast, on page 171](#)
- [About Tenant Routed Multicast Mixed Mode, on page 173](#)
- [Guidelines and Limitations for Tenant Routed Multicast, on page 173](#)
- [Guidelines and Limitations for Layer 3 Tenant Routed Multicast, on page 174](#)
- [Guidelines and Limitations for Layer 2/Layer 3 Tenant Routed Multicast \(Mixed Mode\), on page 174](#)
- [Rendezvous Point for Tenant Routed Multicast, on page 175](#)
- [Configuring a Rendezvous Point for Tenant Routed Multicast, on page 175](#)
- [Configuring RP Everywhere with PIM Anycast, on page 179](#)
- [Configuring RP Everywhere with MSDP Peering, on page 184](#)
- [Configuring Layer 3 Tenant Routed Multicast, on page 190](#)
- [Configuring TRM on the VXLAN EVPN Spine, on page 194](#)
- [Configuring Tenant Routed Multicast in Layer 2/Layer 3 Mixed Mode, on page 196](#)
- [Configuring Layer 2 Tenant Routed Multicast, on page 201](#)

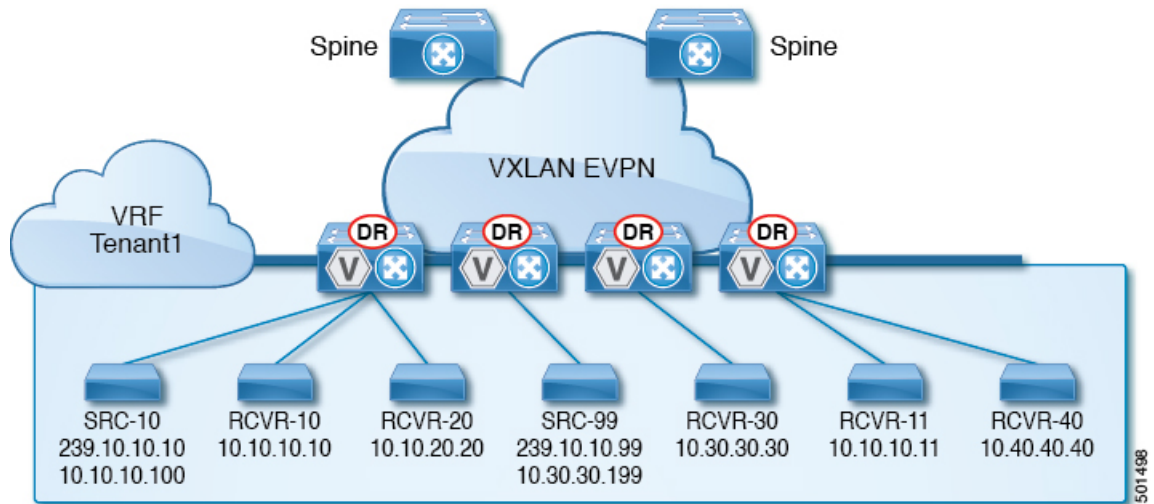
About Tenant Routed Multicast

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

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

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

Figure 35: VXLAN EVPN TRM

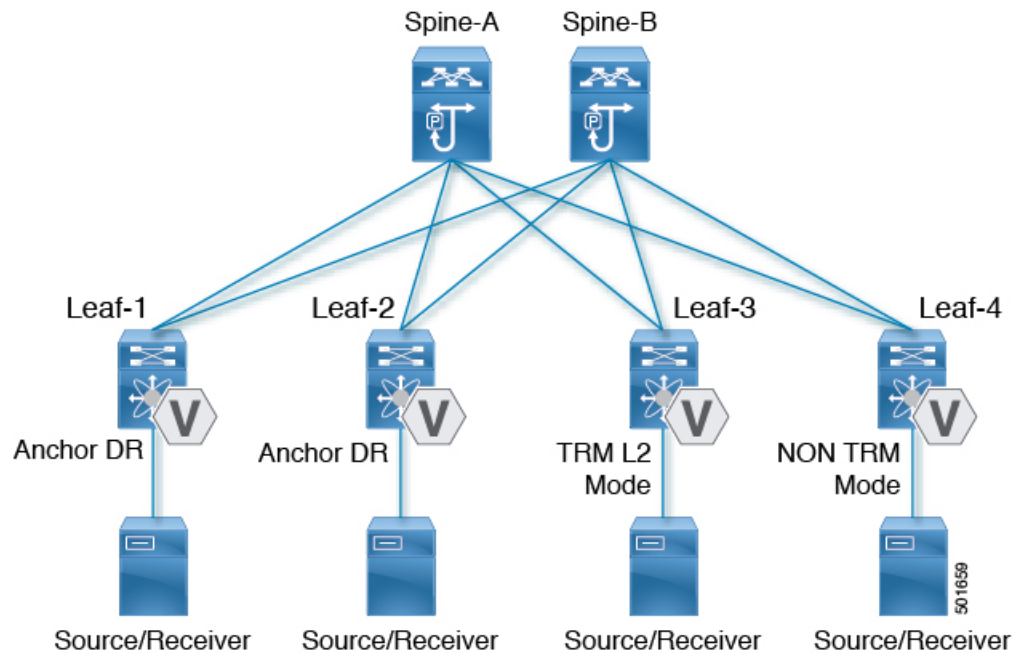


With TRM enabled, multicast forwarding in the underlay is leveraged to replicate VXLAN encapsulated routed multicast traffic. A Default Multicast Distribution Tree (Default-MDT) is built per-VRF. This is an addition to the existing multicast groups for Layer-2 VNI Broadcast, Unknown Unicast, and Layer-2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

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

About Tenant Routed Multicast Mixed Mode

Figure 36: TRM Layer2/Layer 3 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 15](#) 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, 9364C, 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.

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

Configuring a Rendezvous Point for Tenant Routed Multicast

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

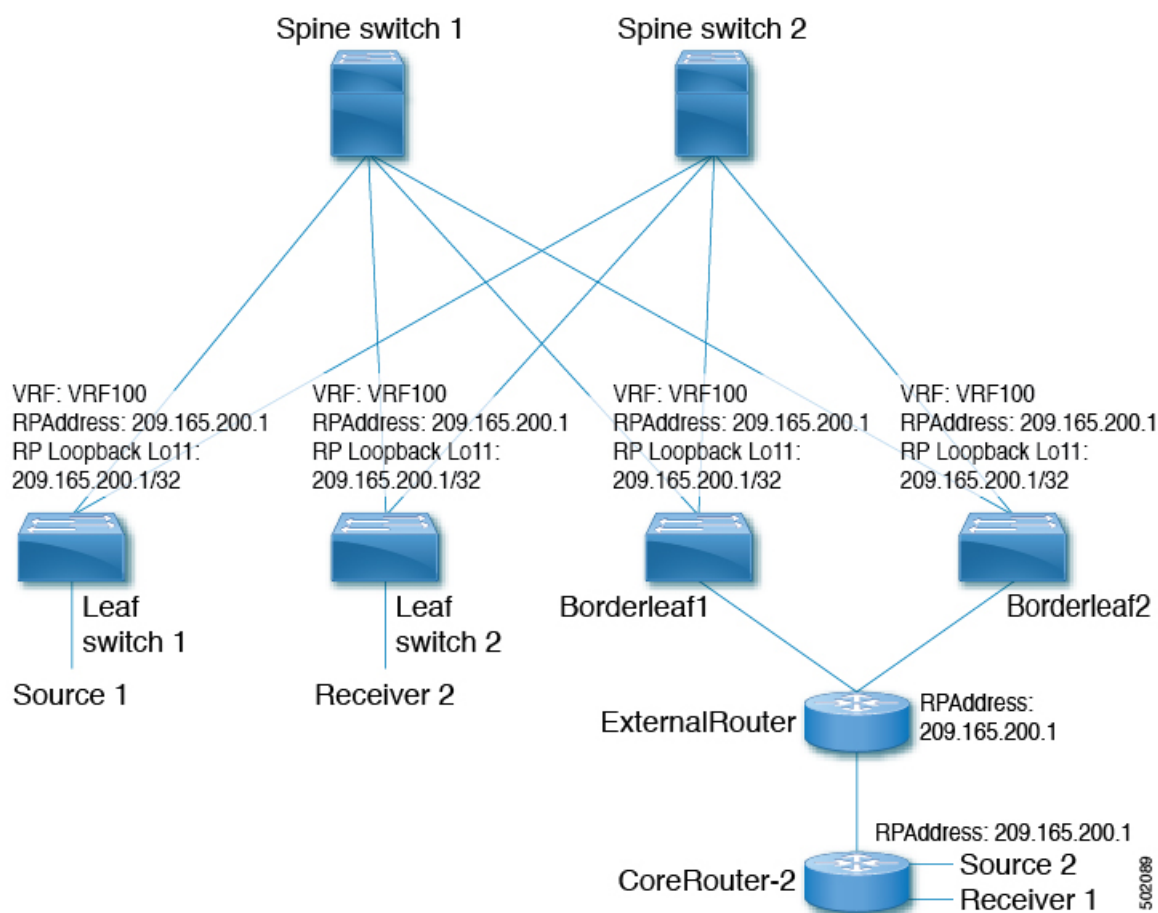
- [Configuring a Rendezvous Point Inside the VXLAN Fabric, on page 176](#)
- [Configuring an External Rendezvous Point, on page 177](#)
- [Configuring RP Everywhere with PIM Anycast, on page 179](#)
 - [Configuring a TRM Leaf Node for RP Everywhere with PIM Anycast, on page 180](#)
 - [Configuring a TRM Border Leaf Node for RP Everywhere with PIM Anycast, on page 180](#)

- [Configuring an External Router for RP Everywhere with PIM Anycast, on page 182](#)
- [Configuring RP Everywhere with MSDP Peering, on page 184](#)
 - [Configuring a TRM Leaf Node for RP Everywhere with MSDP Peering, on page 185](#)
 - [Configuring a TRM Border Leaf Node for RP Everywhere with MSDP Peering, on page 186](#)
 - [Configuring an External Router for RP Everywhere with MSDP Peering, on page 188](#)

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 37:

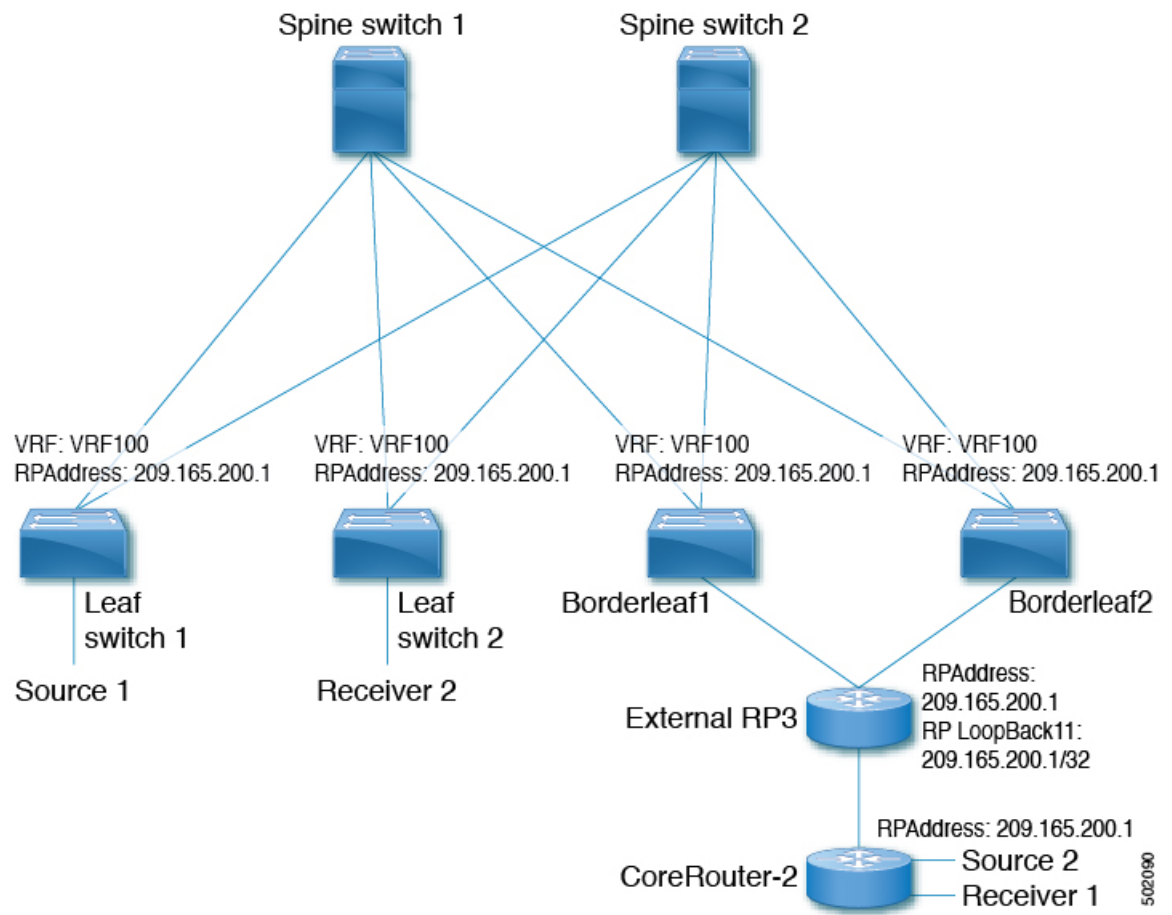


Procedure

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

Configuring an External Rendezvous Point

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

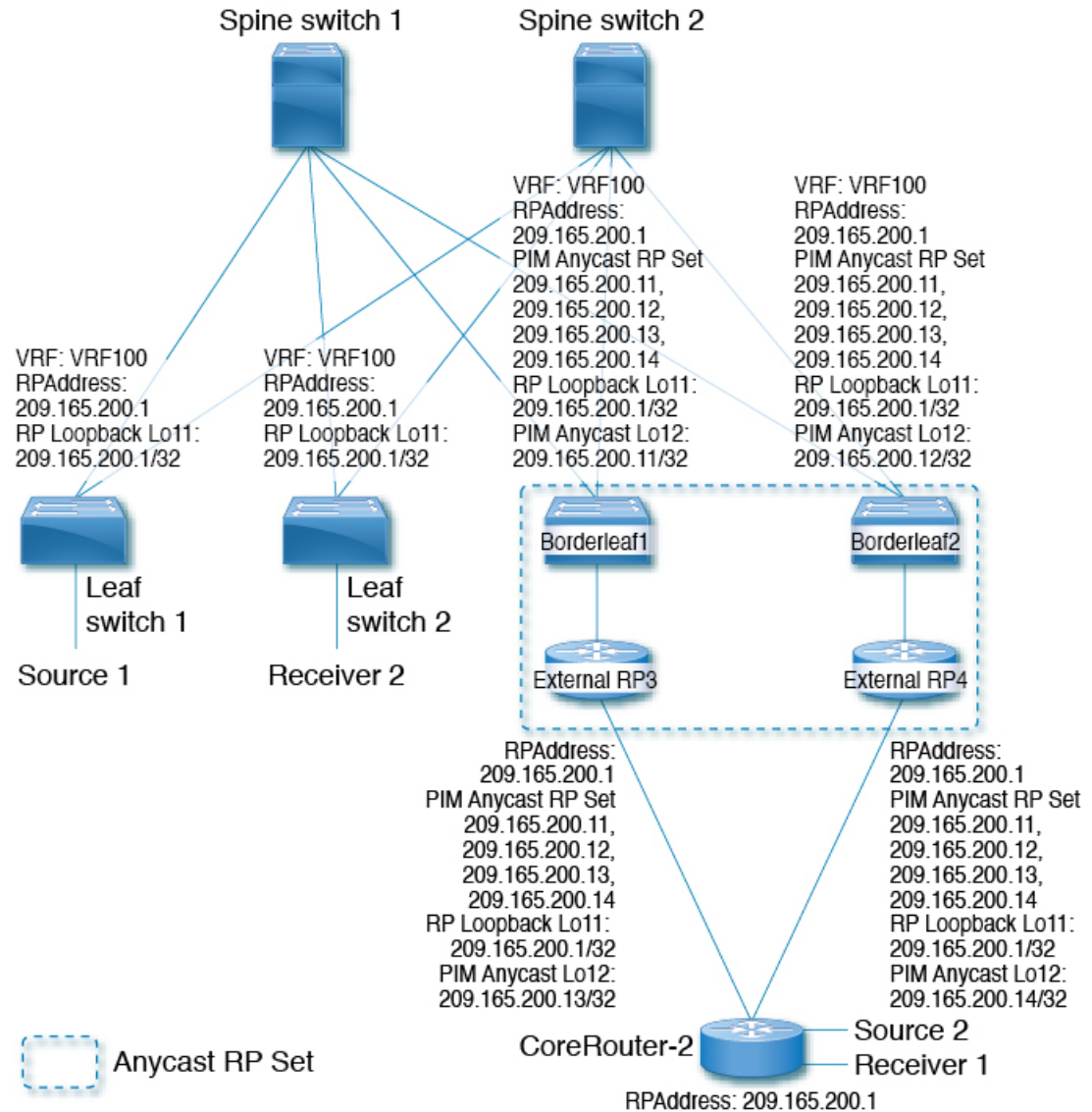


Procedure

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

Configuring RP Everywhere with PIM Anycast

RP Everywhere configuration with PIM Anycast solution.



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

- [Configuring a TRM Leaf Node for RP Everywhere with PIM Anycast, on page 180](#)
- [Configuring a TRM Border Leaf Node for RP Everywhere with PIM Anycast, on page 180](#)
- [Configuring an External Router for RP Everywhere with PIM Anycast, on page 182](#)

Configuring a TRM Leaf Node for RP Everywhere with PIM Anycast

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

Procedure

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

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

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

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: switch# configure terminal	Enter configuration mode.
Step 2	{ip ipv6} pim evpn-border-leaf Example: switch(config)# ipv6 pim evpn-border-leaf	Configure VXLAN VTEP as TRM border leaf node,
Step 3	interface loopback loopback_number Example: switch(config)# interface loopback 11	Configure the loopback interface on all VXLAN VTEP devices.
Step 4	vrf member vrf-name Example: switch(config-if)# vrf member vrf100	Configure VRF name.
Step 5	ip address ip-address Example: switch(config-if)# ip address 209.165.200.1/32	Specify IP address.
Step 6	ipv6 pim sparse-mode Example: switch(config-if)# ipv6 pim sparse-mode	Configure sparse-mode PIM on an interface.
Step 7	interface loopback loopback_number Example: switch(config)# interface loopback 12	Configure the PIM Anycast set RP loopback interface.
Step 8	vrf member vxlan-number Example: switch(config-if)# vrf member vxlan-number	Configure VRF name.
Step 9	ipv6 address ipv6-address Example: switch(config-if)# ip address 209.165.200.11/32	Specify IP address.
Step 10	ipv6 pim sparse-mode Example: switch(config-if)# ipv6 pim sparse-mode	Configure sparse-mode PIM on an interface.

	Command or Action	Purpose
Step 11	vrf context <i>vrf-name</i> Example: switch(config-if)# vrf context vrf100	Create a VXLAN tenant VRF.
Step 12	ipv6 pim rp-address <i>ipv6-address-of-router</i> group-list <i>group-range-prefix</i> Example: switch(config-vrf)# ipv6 pim rp-address 2090:165:200::1 group fflle::/16	The value of the <i>ip-address-of-router</i> parameters is that of the RP. The same IP address must be on all the edge devices (VTEPs) for a fully distributed RP.
Step 13	ipv6 pim anycast-rp <i>anycast-rp-address</i> <i>address-of-rp</i> Example: switch(config-vrf)# ipv6 pim anycast-rp 2090:165:2000::1 2090:165:2000::11	Configure PIM Anycast RP set.
Step 14	ipv6 pim anycast-rp <i>anycast-rp-address</i> <i>address-of-rp</i> Example: switch(config-vrf)# ipv6 pim anycast-rp 2090:165:2000::1 2090:165:2000::12	Configure PIM Anycast RP set.
Step 15	ipv6 pim anycast-rp <i>anycast-rp-address</i> <i>address-of-rp</i> Example: switch(config-vrf)# ipv6 pim anycast-rp 2090:165:2000::1 2090:165:2000::13	Configure PIM Anycast RP set.
Step 16	ipv6 pim anycast-rp <i>anycast-rp-address</i> <i>address-of-rp</i> Example: switch(config-vrf)# ipv6 pim anycast-rp 2090:165:2000::1 2090:165:2000::14	Configure PIM Anycast RP set.

Configuring an External Router for RP Everywhere with PIM Anycast

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

Procedure

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

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

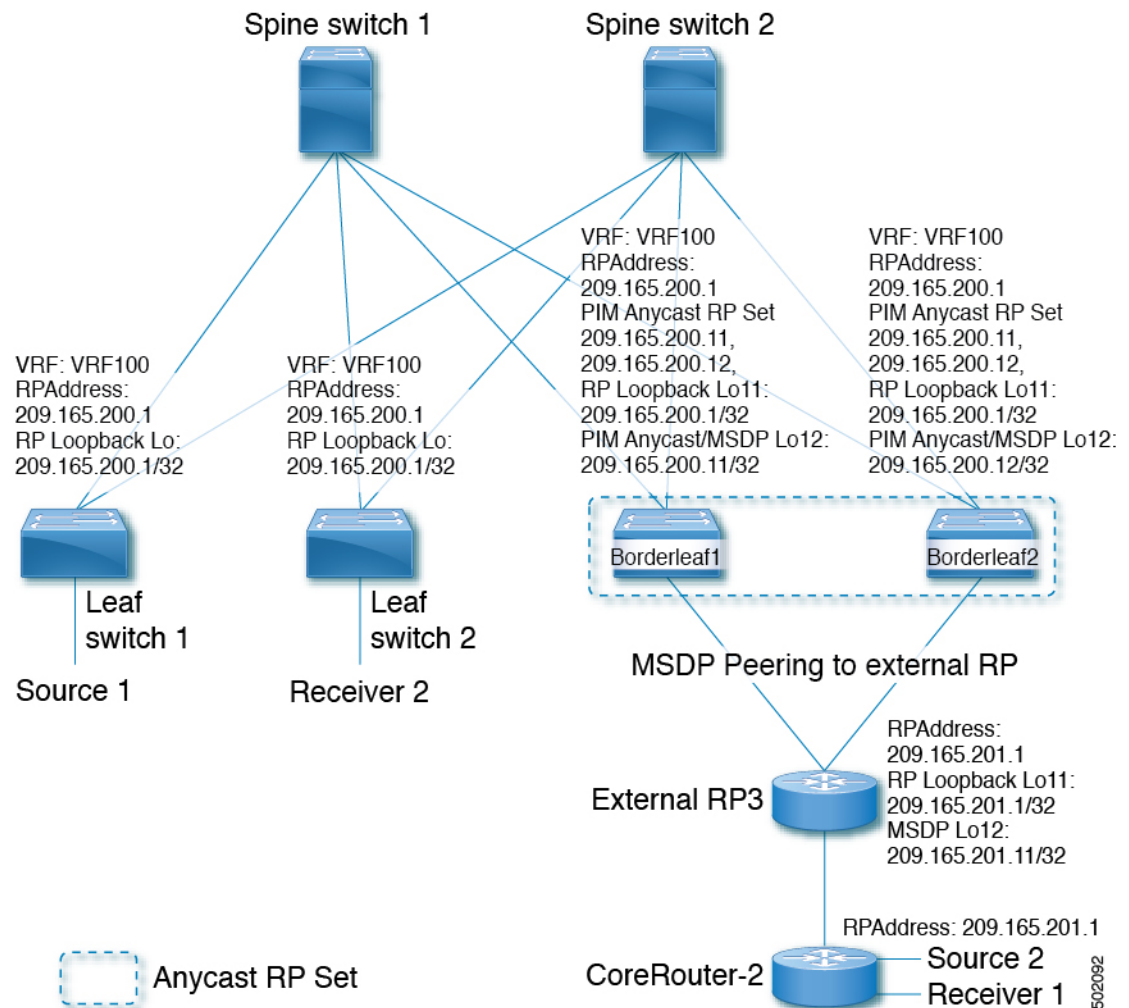
	Command or Action	Purpose
Step 12	ip pim anycast-rp <i>anycast-rp-address</i> <i>address-of-rp</i> Example: <pre>switch(config-vrf)# ip pim anycast-rp 209.165.200.1 209.165.200.11</pre>	Configure PIM Anycast RP set.
Step 13	ip pim anycast-rp <i>anycast-rp-address</i> <i>address-of-rp</i> Example: <pre>switch(config-vrf)#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

Configuring RP Everywhere with MSDP Peering

The following figure represents the RP Everywhere configuration with MSDP RP solution.

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

- [Configuring a TRM Leaf Node for RP Everywhere with MSDP Peering, on page 185](#)
- [Configuring a TRM Border Leaf Node for RP Everywhere with MSDP Peering, on page 186](#)
- [Configuring an External Router for RP Everywhere with MSDP Peering, on page 188](#)



Configuring a TRM Leaf Node for RP Everywhere with MSDP Peering

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

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: <pre>switch# configure terminal</pre>	Enter configuration mode.
Step 2	interface loopback loopback_number Example: <pre>switch(config)# interface loopback 11</pre>	Configure the loopback interface on all VXLAN VTEP devices.

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

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

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

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: switch# configure terminal	Enter configuration mode.
Step 2	feature msdp Example: switch(config)# feature msdp	Enable feature MSDP.
Step 3	ip pim evpn-border-leaf Example: switch(config)# ip pim evpn-border-leaf	Configure VXLAN VTEP as TRM border leaf node,
Step 4	interface loopback <i>loopback_number</i> Example:	Configure the loopback interface on all VXLAN VTEP devices.

	Command or Action	Purpose
	<code>switch(config)# interface loopback 11</code>	
Step 5	vrf member <i>vrf-name</i> Example: <code>switch(config-if)# vrf member vrf100</code>	Configure VRF name.
Step 6	ip address <i>ip-address</i> Example: <code>switch(config-if)# ip address 209.165.200.1/32</code>	Specify IP address.
Step 7	ip pim sparse-mode Example: <code>switch(config-if)# ip pim sparse-mode</code>	Configure sparse-mode PIM on an interface.
Step 8	interface loopback <i>loopback_number</i> Example: <code>switch(config)# interface loopback 12</code>	Configure the PIM Anycast set RP loopback interface.
Step 9	vrf member <i>vrf-name</i> Example: <code>switch(config-if)# vrf member vrf100</code>	Configure VRF name.
Step 10	ip address <i>ip-address</i> Example: <code>switch(config-if)# ip address 209.165.200.11/32</code>	Specify IP address.
Step 11	ip pim sparse-mode Example: <code>switch(config-if)# ip pim sparse-mode</code>	Configure sparse-mode PIM on an interface.
Step 12	vrf context <i>vrf-name</i> Example: <code>switch(config-if)# vrf context vrf100</code>	Create a VXLAN tenant VRF.
Step 13	ip pim rp-address <i>ip-address-of-router</i> group-list <i>group-range-prefix</i> Example: <code>switch(config-vrf)# ip pim rp-address 209.165.200.1 group-list 224.0.0.0/4</code>	The value of the <i>ip-address-of-router</i> parameter is that of the RP. The same IP address must be on all the edge devices (VTEPs) for a fully distributed RP.
Step 14	ip pim anycast-rp <i>anycast-rp-address</i> <i>address-of-rp</i> Example:	Configure PIM Anycast RP set.

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

Configuring an External Router for RP Everywhere with MSDP Peering

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: <code>switch# configure terminal</code>	Enter configuration mode.
Step 2	feature msdp Example: <code>switch(config)# feature msdp</code>	Enable feature MSDP.
Step 3	interface loopback <i>loopback_number</i> Example: <code>switch(config)# interface loopback 11</code>	Configure the loopback interface on all VXLAN VTEP devices.
Step 4	vrf member <i>vrf-name</i> Example: <code>switch(config-if)# vrf member vrf100</code>	Configure VRF name.
Step 5	ip address <i>ip-address</i> Example: <code>switch(config-if)# ip address 209.165.201.1/32</code>	Specify IP address.

	Command or Action	Purpose
Step 6	ip pim sparse-mode Example: switch(config-if)# ip pim sparse-mode	Configure sparse-mode PIM on an interface.
Step 7	interface loopback <i>loopback_number</i> Example: switch(config)# interface loopback 12	Configure the PIM Anycast set RP loopback interface.
Step 8	vrf member <i>vrf-name</i> Example: switch(config-if)# vrf member vrf100	Configure VRF name.
Step 9	ip address <i>ip-address</i> Example: switch(config-if)# ip address 209.165.201.11/32	Specify IP address.
Step 10	ip pim sparse-mode Example: switch(config-if)# ip pim sparse-mode	Configure sparse-mode PIM on an interface.
Step 11	vrf context <i>vrf-name</i> Example: switch(config-if)# vrf context vrf100	Create a VXLAN tenant VRF.
Step 12	ip pim rp-address <i>ip-address-of-router</i> group-list <i>group-range-prefix</i> Example: switch(config-vrf)# ip pim rp-address 209.165.201.1 group-list 224.0.0.0/4	The value of the <i>ip-address-of-router</i> parameters is that of the RP. The same IP address must be on all the edge devices (VTEPs) for a fully distributed RP.
Step 13	ip msdp originator-id loopback12 Example: switch(config-vrf)# ip msdp originator-id loopback12	Configure MSDP originator ID.
Step 14	ip msdp peer <i>ip-address</i> connect-source loopback12 Example: switch(config-vrf)# ip msdp peer 209.165.200.11 connect-source loopback12	Configure MSDP peering between external RP router and all TRM border nodes.

Configuring Layer 3 Tenant Routed Multicast

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

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



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

Before you begin

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

The rendezvous point (RP) must be configured.

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: <code>switch# configure terminal</code>	Enter configuration mode.
Step 2	feature ngmvpn Example: <code>switch(config)# feature ngmvpn</code>	Enables the Next-Generation Multicast VPN (ngMVPN) control plane. New address family commands become available in BGP.
Step 3	ip igmp snooping vxlan Example: <code>switch(config)# ip igmp snooping vxlan</code>	Configure IGMP snooping for VXLANs.
Step 4	interface nve1 Example: <code>switch(config)# interface nve 1</code>	Configure the NVE interface.
Step 5	member vni vni-range associate-vrf Example: <code>switch(config-if-nve)# member vni 200100 associate-vrf</code>	Configure the virtual network identifier. The range of <i>vni-range</i> is from 1 to 16,777,214.
Step 6	mcast-group ip-prefix Example:	Builds the default multicast distribution tree for the VRF VNI (Layer 3 VNI).

	Command or Action	Purpose
	<pre>switch(config-if-nve-vni) # mcast-group 225.3.3.3</pre>	<p>The multicast group is used in the underlay (core) for all multicast routing within the associated Layer 3 VNI (VRF).</p> <p>Note We recommend that underlay multicast groups for Layer 2 VNI, default MDT, and data MDT not be shared. Use separate, non-overlapping groups.</p>
Step 7	<p>exit</p> <p>Example:</p> <pre>switch(config-if-nve-vni) # exit</pre>	Exits command mode.
Step 8	<p>exit</p> <p>Example:</p> <pre>switch(config-if) # exit</pre>	Exits command mode.
Step 9	<p>router bgp 100</p> <p>Example:</p> <pre>switch(config) # router bgp 100</pre>	Set autonomous system number.
Step 10	<p>exit</p> <p>Example:</p> <pre>switch(config-router) # exit</pre>	Exits command mode.
Step 11	<p>neighbor ip-addr</p> <p>Example:</p> <pre>switch(config-router) # neighbor 1.1.1.1</pre>	Configure IP address of the neighbor.
Step 12	<p>address-family ipv4 mvpn</p> <p>Example:</p> <pre>switch(config-router-neighbor) # address-family ipv4 mvpn</pre>	Configure multicast VPN.
Step 13	<p>send-community extended</p> <p>Example:</p> <pre>switch(config-router-neighbor-af) # send-community extended</pre>	Enables ngMVPN for address family signalization. The send community both command ensures both standard and extended communities are exchanged for this address family.
Step 14	<p>exit</p> <p>Example:</p> <pre>switch(config-router-neighbor-af) # exit</pre>	Exits command mode.

	Command or Action	Purpose
Step 15	exit Example: <code>switch(config-router)# exit</code>	Exits command mode.
Step 16	vrf context <i>vrf_name</i> Example: <code>switch(config-router)#vrf context vrf100</code>	Configure VRF name.
Step 17	ip pim rp-address <i>ip-address-of-router</i> group-list <i>group-range-prefix</i> Example: <code>switch(config-vrf)# ip pim rp-address 99.99.99.1 group-list 226.0.0.0/8</code>	<p>The value of the <i>ip-address-of-router</i> parameter is that of the RP. The same IP address must be on all of the edge devices (VTEPs) for a fully distributed RP.</p> <p>For overlay RP placement options, see the Configuring a Rendezvous Point for Tenant Routed Multicast, on page 175 section.</p>
Step 18	address-family ipv4 unicast Example: <code>switch(config-vrf)# address-family ipv4 unicast</code>	Configure unicast address family.
Step 19	route-target both auto mvpn Example: <code>switch(config-vrf-af-ipv4)# route-target both auto mvpn</code>	<p>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).</p> <p>Auto route targets are constructed by the 2-byte Autonomous System Number (ASN) and Layer 3 VNI.</p>
Step 20	ip multicast overlay-spt-only Example: <code>switch(config)# ip multicast overlay-spt-only</code>	Add the ip multicast overlay-spt-only command on all MVPN-enabled Cisco Nexus 9000 Series switches (typically leaf node).
Step 21	interface <i>vlan_id</i> Example: <code>switch(config)# interface vlan11</code>	Configures the first-hop gateway (distributed anycast gateway for the Layer 2 VNI. No router PIM peering must ever happen with this interface.
Step 22	no shutdown Example: <code>switch(config-if)# no shutdown</code>	Disables an interface.
Step 23	vrf member <i>vrf-num</i> Example: <code>switch(config-if)# vrf member vrf100</code>	Configure VRF name.

	Command or Action	Purpose
Step 24	ip address <i>ip_address</i> Example: <pre>switch(config-if) # ip address 11.1.1.1/24</pre>	Configure IP address.
Step 25	ip pim sparse-mode Example: <pre>switch(config-if) # ip pim sparse-mode</pre>	Enables IGMP and PIM on the SVI. This is required if multicast sources and/or receivers exist in this VLAN.
Step 26	fabric forwarding mode anycast-gateway Example: <pre>switch(config-if) # fabric forwarding mode anycast-gateway</pre>	Configure Anycast Gateway Forwarding Mode.
Step 27	ip pim neighbor-policy NONE* Example: <pre>switch(config-if) # ip pim neighbor-policy NONE*</pre>	<p>Creates an IP PIM neighbor policy to avoid PIM neighborship with PIM routers within the VLAN. The none keyword is a configured route map to deny any ipv4 addresses to avoid establishing PIM neighborship policy using anycase IP.</p> <p>Note Do not use Distributed Anycast Gateway for PIM Peerings.</p>
Step 28	exit Example: <pre>switch(config-if) # exit</pre>	Exits command mode.
Step 29	interface <i>vlan_id</i> Example: <pre>switch(config) # interface vlan100</pre>	Configure Layer 3 VNI.
Step 30	no shutdown Example: <pre>switch(config-if) # no shutdown</pre>	Disable an interface.
Step 31	vrf member vrf100 Example: <pre>switch(config-if) # vrf member vrf100</pre>	Configure VRF name.
Step 32	ip forward Example: <pre>switch(config-if) # ip forward</pre>	Enable IP forwarding on interface.
Step 33	ip pim sparse-mode Example:	Configure sparse-mode PIM on interface. There is no PIM peering happening in the

	Command or Action	Purpose
	<code>switch(config-if)# ip pim sparse-mode</code>	Layer-3 VNI, but this command must be present for forwarding.

Configuring TRM on the VXLAN EVPN Spine

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

Before you begin

The VXLAN BGP EVPN spine must be configured. See [Configuring BGP for EVPN on the Spine, on page 93](#).

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: <code>switch# configure terminal</code>	Enter configuration mode.
Step 2	route-map permitall permit 10 Example: <code>switch(config)# route-map permitall permit 10</code>	Configure the route-map. Note The route-map keeps the next-hop unchanged for EVPN routes <ul style="list-style-type: none"> • Required for eBGP • Options for iBGP
Step 3	set ip next-hop unchanged Example: <code>switch(config-route-map)# set ip next-hop unchanged</code>	Set next hop address. Note The route-map keeps the next-hop unchanged for EVPN routes <ul style="list-style-type: none"> • Required for eBGP • Options for iBGP
Step 4	exit Example: <code>switch(config-route-map)# exit</code>	Return to exec mode.
Step 5	router bgp [autonomous system] number Example: <code>switch(config)# router bgp 65002</code>	Specify BGP.

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

	Command or Action	Purpose
Step 14	route-map permitall out Example: <pre>switch(config-router-neighbor-af) # route-map permitall out</pre>	Applies route-map to keep the next-hop unchanged. Note Required for eBGP.

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

	Command or Action	Purpose
Step 1	configure terminal Example: <pre>switch# configure terminal</pre>	Enter configuration mode.
Step 2	feature ngmvpn Example: <pre>switch(config)# feature ngmvpn</pre>	Enables the Next-Generation Multicast VPN (ngMVPN) control plane. New address family commands become available in BGP.
Step 3	advertise evpn multicast Example: <pre>switch(config)# advertise evpn multicast</pre>	Advertises IMET and SMET routes into BGP EVPN towards non-TRM capable switches.
Step 4	ip igmp snooping vxlan Example: <pre>switch(config)# ip igmp snooping vxlan</pre>	Configure IGMP snooping for VXLANs.
Step 5	ip multicast overlay-spt-only Example:	Gratuitously originate (S,A) route when source is locally connected.

	Command or Action	Purpose
	<code>switch(config)# ip multicast overlay-spt-only</code>	
Step 6	ip multicast overlay-distributed-dr Example: <code>switch(config)# ip multicast overlay-distributed-dr</code>	Enables distributed anchor DR function on this VTEP.
Step 7	interface nve1 Example: <code>switch(config)# interface nve 1</code>	Configure the NVE interface.
Step 8	member vni vni-range associate-vrf Example: <code>switch(config-if-nve)# member vni 200100 associate-vrf</code>	Configure the virtual network identifier. The range of <i>vni-range</i> is from 1 to 16,777,214.
Step 9	mcast-group ip-prefix Example: <code>switch(config-if-nve-vni)# mcast-group 225.3.3.3</code>	Configures the multicast group on distributed anchor DR.
Step 10	exit Example: <code>switch(config-if-nve-vni)# exit</code>	Exits command mode.
Step 11	interface loopback loopback_number Example: <code>switch(config-if-nve)# interface loopback 10</code>	Configure the loopback interface on all distributed anchor DR devices.
Step 12	ip address ip_address Example: <code>switch(config-if)# ip address 100.100.1.1/32</code>	Configure IP address. This IP address is the same on all distributed anchor DR.
Step 13	ip router ospf process-tag area ospf-id Example: <code>switch(config-if)# ip router ospf 100 area 0.0.0.0</code>	OSPF area ID in IP address format.
Step 14	ip pim sparse-mode Example: <code>switch(config-if)# ip pim sparse-mode</code>	Configure sparse-mode PIM on interface.
Step 15	interface nve1 Example:	Configure NVE interface.

	Command or Action	Purpose
	<code>switch(config-if)# interface nve1</code>	
Step 16	shutdown Example: <code>switch(config-if-nve)# shutdown</code>	Disable the interface.
Step 17	mcast-routing override source-interface loopback int-num Example: <code>switch(config-if-nve)# mcast-routing override source-interface loopback 10</code>	<p>Enables that TRM is using a different source-interface interface than the VTEPs default source-interface.</p> <p>The <i>loopback2</i> 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.</p>
Step 18	exit Example: <code>switch(config-if-nve)# exit</code>	Exits command mode.
Step 19	router bgp 100 Example: <code>switch(config)# router bgp 100</code>	Set autonomous system number.
Step 20	neighbor ip-addr Example: <code>switch(config-router)# neighbor 1.1.1.1</code>	Configure IP address of the neighbor.
Step 21	address-family ipv4 mvpn Example: <code>switch(config-router-neighbor)# address-family ipv4 mvpn</code>	Configure multicast VPN.
Step 22	send-community extended Example: <code>switch(config-router-neighbor-af)# send-community extended</code>	Send community attribute.
Step 23	exit Example: <code>switch(config-router-neighbor-af)# exit</code>	Exits command mode.
Step 24	exit Example: <code>switch(config-router)# exit</code>	Exits command mode.

	Command or Action	Purpose
Step 25	vrf <i>vrf_name</i> vrf100 Example: switch(config) # vrf context vrf100	Configure VRF name.
Step 26	ip pim rp-address <i>ip-address-of-router</i> group-list <i>group-range-prefix</i> Example: switch(config-vrf) # ip pim rp-address 99.99.99.1 group-list 226.0.0.0/8	<p>The value of the <i>ip-address-of-router</i> parameter is that of the RP. The same IP address must be on all of the edge devices (VTEPs) for a fully distributed RP.</p> <p>For overlay RP placement options, see the Configuring a Rendezvous Point for Tenant Routed Multicast, on page 175 - Internal RP section.</p>
Step 27	address-family ipv4 unicast Example: switch(config-vrf) # address-family ipv4 unicast	Configure unicast address family.
Step 28	route-target both auto mvpn Example: switch(config-vrf-af-ipv4) # route-target both auto mvpn	Specify target for mvpn routes.
Step 29	exit Example: switch(config-vrf-af-ipv4) # exit	Exits command mode.
Step 30	exit Example: switch(config-vrf) # exit	Exits command mode.
Step 31	interface <i>vlan_id</i> Example: switch(config) # interface vlan11	Configure Layer 2 VNI.
Step 32	no shutdown Example: switch(config-if) # no shutdown	Disable an interface.
Step 33	vrf member vrf100 Example: switch(config-if) # vrf member vrf100	Configure VRF name.
Step 34	ip address <i>ip_address</i> Example:	Configure IP address.

	Command or Action	Purpose
	<code>switch(config-if)# ip address 11.1.1.1/24</code>	
Step 35	ip pim sparse-mode Example: <code>e</code> <code>switch(config-if)# ip pim sparse-mode</code>	Configure sparse-mode PIM on interface.
Step 36	fabric forwarding mode anycast-gateway Example: <code>switch(config-if)# fabric forwarding mode anycast-gateway</code>	Configure Anycast Gateway Forwarding Mode.
Step 37	ip pim neighbor-policy NONE* Example: <code>switch(config-if)# ip pim neighbor-policy NONE*</code>	The none keyword is a configured route map to deny any IPv4 addresses to avoid establishing a PIM neighborship policy using anycase IP.
Step 38	exit Example: <code>switch(config-if)# exit</code>	Exits command mode.
Step 39	interface <i>vlan_id</i> Example: <code>switch(config)# interface vlan100</code>	Configure Layer 3 VNI.
Step 40	no shutdown Example: <code>switch(config-if)# no shutdown</code>	Disable an interface.
Step 41	vrf member vrf100 Example: <code>switch(config-if)# vrf member vrf100</code>	Configure VRF name.
Step 42	ip forward Example: <code>switch(config-if)# ip forward</code>	Enable IP forwarding on interface.
Step 43	ip pim sparse-mode Example: <code>switch(config-if)# ip pim sparse-mode</code>	Configure sparse-mode PIM on interface.

Configuring Layer 2 Tenant Routed Multicast

This procedure enables the Tenant Routed Multicast (TRM) feature. This enables Layer 2 multicast BGP signaling.

IGMP Snooping Querier must be configured per multicast-enabled VXLAN VLAN on all Layer-2 TRM leaf switches.

Before you begin

VXLAN EVPN must be configured.

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: <code>switch# configure terminal</code>	Enter configuration mode.
Step 2	feature ngmvpn Example: <code>switch(config)# feature ngmvpn</code>	Enables EVPN/MVPN feature.
Step 3	advertise evpn multicast Example: <code>switch(config)# advertise evpn multicast</code>	Advertise L2 multicast capability.
Step 4	ip igmp snooping vxlan Example: <code>switch(config)# ip igmp snooping vxlan</code>	Configure IGMP snooping for VXLANs.



CHAPTER 10

Configuring VXLAN QoS

This chapter contains the following sections:

- [Information About VXLAN QoS, on page 203](#)
- [Guidelines and Limitations for VXLAN QoS, on page 211](#)
- [Default Settings for VXLAN QoS, on page 212](#)
- [Configuring VXLAN QoS, on page 213](#)
- [Verifying the VXLAN QoS Configuration, on page 215](#)
- [VXLAN QoS Configuration Examples, on page 215](#)

Information About VXLAN QoS

VXLAN QoS enables you to provide Quality of Service (QoS) capabilities to traffic that is tunneled in VXLAN.

Traffic in the VXLAN overlay can be assigned to different QoS properties:

- Classification traffic to assign different properties.
- Including traffic marking with different priorities.
- Queuing traffic to enable priority for the protected traffic.
- Policing for misbehaving traffic.
- Shaping for traffic that limits speed per interface.
- Properties traffic sensitive to traffic drops.



Note

QoS allows you to classify the network traffic, police and prioritize the traffic flow, and provide congestion avoidance. For more information about QoS, see the [Cisco Nexus 9000 Series NX-OS Quality of Service Configuration Guide, Release 7.x](#).

VXLAN QoS Terminology

This section defines VXLAN QoS terminology.

Table 8: VXLAN QoS Terminology

Term	Definition
Frames	Carries traffic at Layer 2. Layer 2 frames carry Layer 3 packets.
Packets	Carries traffic at Layer 3.
VXLAN packet	Carries original frame, encapsulated in VXLAN IP/UDP header.
Original frame	A Layer 2 or Layer 2 frame that carries the Layer 3 packet before encapsulation in a VXLAN header.
Decapsulated frame	A Layer 2 or a Layer 2 frame that carries a Layer 3 packet after the VXLAN header is decapsulated.
Ingress VTEP	The point where traffic is encapsulated in the VXLAN header and enters the VXLAN tunnel.
Egress VTEP	The point where traffic is decapsulated from the VXLAN header and exits the VXLAN tunnel.
Class of Service (CoS)	Refers to the three bits in an 802.1Q header that are used to indicate the priority of the Ethernet frame as it passes through a switched network. The CoS bits in the 802.1Q header are commonly referred to as the 802.1p bits. 802.1Q is discarded prior to frame encapsulation in a VXLAN header, where CoS value is not present in VXLAN tunnel. To maintain QoS when a packet enters the VXLAN tunnel, the type of service (ToS) and CoS values map to each other.
IP precedence	The 3 most significant bits of the ToS byte in the IP header.
Differentiated Services Code Point (DSCP)	The first six bits of the ToS byte in the IP header. DSCP is only present in an IP packet.
Explicit Congestion Notification (ECN)	The last two bits of the ToS byte in the IP header. ECN is only present in an IP packet.
QoS tags	Prioritization values carried in Layer 3 packets and Layer 2 frames. A Layer 2 CoS label can have a value ranging between zero for low priority and seven for high priority. A Layer 3 IP precedence label can have a value ranging between zero for low priority and seven for high priority. IP precedence values are defined by the three most significant bits of the 1-byte ToS byte. A Layer 3 DSCP label can have a value between 0 and 63. DSCP values are defined by the six most significant bits of the 1-byte IP ToS field.

Term	Definition
Classification	The process used for selecting traffic for QoS
Marking	The process of setting: a Layer 2 COS value in a frame, Layer 3 DSCP value in a packet, and Layer 3 ECN value in a packet. Marking is also the process of choosing different values for the CoS, DSCP, ECN field to mark packets so that they have the priority that they require during periods of congestion.
Policing	Limiting bandwidth used by a flow of traffic. Policing can mark or drop traffic.
MQC	The Cisco Modular QoS command line interface (MQC) framework, which is a modular and highly extensible framework for deploying QoS.

VXLAN QoS Features

Trust Boundaries

The trust boundary forms a perimeter on your network. Your network trusts (and does not override) the markings on your switch. The existing ToS values are trusted when received on in the VXLAN fabric.

Classification

You use classification to partition traffic into classes. You classify the traffic based on the port characteristics or the packet header fields that include IP precedence, differentiated services code point (DSCP), Layer 3 to Layer 4 parameters, and the packet length.

The values used to classify traffic are called match criteria. When you define a traffic class, you can specify multiple match criteria, you can choose to not match on a particular criterion, or you can determine the traffic class by matching any or all criteria.

Traffic that fails to match any class is assigned to a default class of traffic called class-default.

Marking

Marking is the setting of QoS information that is related to a packet. Packet marking allows you to partition your network into multiple priority levels or classes of service. You can set the value of a standard QoS field for COS, IP precedence, and DSCP. You can also set the QoS field for internal labels (such as QoS groups) that can be used in subsequent actions. Marking QoS groups is used to identify the traffic type for queuing and scheduling traffic.

Policing

Policing causes traffic that exceeds the configured rate to be discarded or marked down to a higher drop precedence.

Single-rate policers monitor the specified committed information rate (CIR) of traffic. Dual-rate policers monitor both CIR and peak information rate (PIR) of traffic.

Queuing and Scheduling

The queuing and scheduling process allows you to control the queue usage and the bandwidth that is allocated to traffic classes. You can then achieve the desired trade-off between throughput and latency.

You can limit the size of the queues for a particular class of traffic by applying either static or dynamic limits.

You can apply weighted random early detection (WRED) to a class of traffic, which allows packets to be dropped based on the QoS group. The WRED algorithm allows you to perform proactive queue management to avoid traffic congestion.

ECN can be enabled along with WRED on a particular class of traffic to mark the congestion state instead of dropping the packets. ECN marking in the VXLAN tunnel is performed in the outer header, and at the Egress VTEP is copied to decapsulated frame.

Traffic Shaping

You can shape traffic by imposing a maximum data rate on a class of traffic so that excess packets are retained in a queue to smooth (constrain) the output rate. In addition, minimum bandwidth shaping can be configured to provide a minimum guaranteed bandwidth for a class of traffic.

Traffic shaping regulates and smooths out the packet flow by imposing a maximum traffic rate for each port's egress queue. Packets that exceed the threshold are placed in the queue and are transmitted later. Traffic shaping is similar to Traffic Policing, but the packets are not dropped. Because packets are buffered, traffic shaping minimizes packet loss (based on the queue length), which provides better traffic behavior for TCP traffic.

By using traffic shaping, you can control the following:

- Access to available bandwidth.
- Ensure that traffic conforms to the policies established for it.
- Regulate the flow of traffic to avoid congestion that can occur when the egress traffic exceeds the access speed of its remote, target interface.

For example, you can control access to the bandwidth when 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](#).

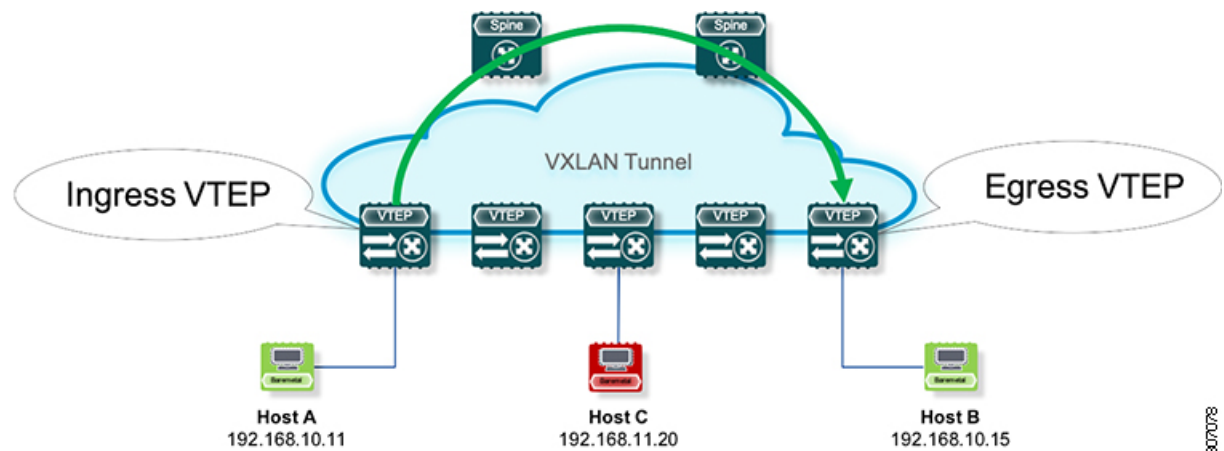
MQC CLI

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 38: VXLAN Network



The network is bidirectional, but in the previous image, traffic is moving left to right.

In the VLXAN 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 VLXAN encapsulated traffic is decapsulated and egresses the VTEP as classical Ethernet traffic.



Note Ingress and egress VTEPs are the boundary between the VXLAN tunnel and the IP network.

Ingress VTEP and Encapsulation in the VXLAN Tunnel

At the ingress VTEP, the VTEP processes packets as follows:

Procedure

- Step 1** Layer 2 or Layer 3 traffic enters the edge of the VXLAN network.
- Step 2** The switch receives the traffic from the input interface and uses the 802.1p bits or the DSCP value to perform any classification, marking, and policing. It also derives the outer DSCP value in the VXLAN header. For classification of incoming IP packets, the input service policy can also use access control lists (ACLs).

- Step 3** For each incoming packet, the switch performs a lookup of the IP address to determine the next hop.
 - Step 4** The packet is encapsulated in the VXLAN header. The encapsulated packet's VXLAN header is assigned a DSCP value that is based on QoS rules.
 - Step 5** The switch forwards the encapsulated packets to the appropriate output interface for processing.
 - Step 6** The encapsulated packets, marked by the DSCP value, are sent to the VXLAN tunnel output interface.
-

Transport Through the VXLAN Tunnel

In the transport through a VXLAN tunnel, the switch processes the VXLAN packets as follows:

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:

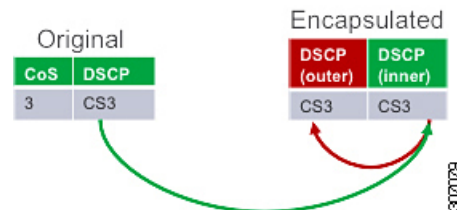
IP to VXLAN

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 39: 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 212](#). In this way, the original QoS attributes are preserved in the VXLAN tunnel. This behavior is illustrated in the following figure:

Figure 40: Copy of Priority from Layer-2 Frame to VXLAN Outer Header



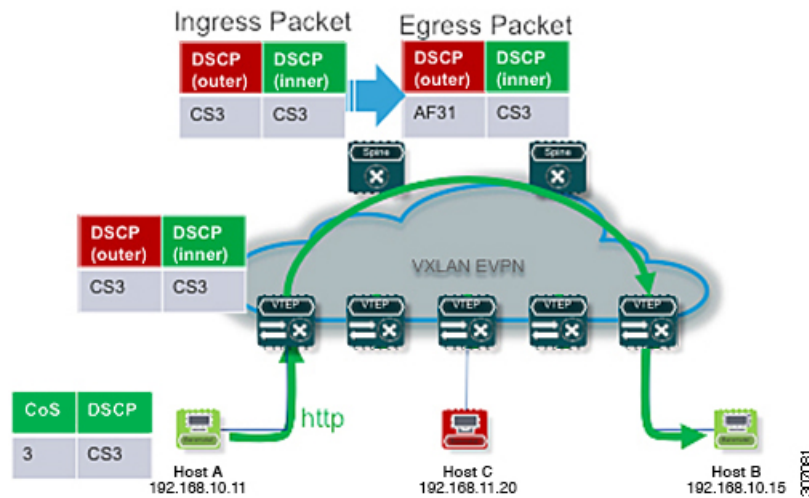
A Layer-2 frame, does not have a DSCP value present because the IP header is not present in the frame. After a Layer-2 frame is encapsulated, the original CoS value is not preserved in the VXLAN tunnel.

Inside the VXLAN Tunnel

Inside the VXLAN tunnel, traffic classification is based on the outer header DSCP value. Classification can be done matching the DSCP value or using ACLs for classification.

If VXLAN encapsulated traffic is crossing the trust boundary, marking can be changed in the packet to match QoS behavior in the tunnel. Marking can be performed inside of the VXLAN tunnel, where a new DSCP value is applied only on the outer header. The new DSCP value can influence different QoS behaviors inside the VXLAN tunnel. The original DSCP value is preserved in the inner header.

Figure 41: Marking Inside of the VXLAN Tunnel



VXLAN to IP

Classification at the egress VTEP is performed for traffic leaving the VXLAN tunnel. For classification at the egress VTEP, the inner header values are used. The inner DSCP value is used for priority-based classification. Classification can be performed using ACLs.

Classification is performed on the NVE interface for all VXLAN tunneled traffic.

Marking and policing can be performed on the NVE interface for tunneled traffic. If marking is configured, newly marked values are present in the decapsulated packet. Because the original CoS value is not preserved in the encapsulated packet, marking can be performed for decapsulated packets for any devices that expect an 802.1p field for QoS in the rest of the network.

Decapsulated Packet Priority Selection

At the egress VTEP, the VXLAN header is removed from the packet and the decapsulated packet egresses the switch with the DSCP value. The switch assigns the DSCP value of the decapsulated packet based on two modes:

- Uniform mode – the DSCP value from the outer header of the VXLAN packet is copied to the decapsulated packet. Any change of the DSCP value in the VXLAN tunnel is preserved and present in the decapsulated packet. Uniform mode is the default mode of decapsulated packet priority selection.
- Pipe mode – the original DSCP value is preserved at the VXLAN tunnel end. At the egress VTEP, the system copies the inner DSCP value to the decapsulated packet DSCP value. In this way, the original DSCP value is preserved at the end of the VXLAN tunnel.

Figure 42: Uniform Mode Outer DSCP Value is Copied to Decapsulated Packet DSCP Value for a Layer-3 Packet

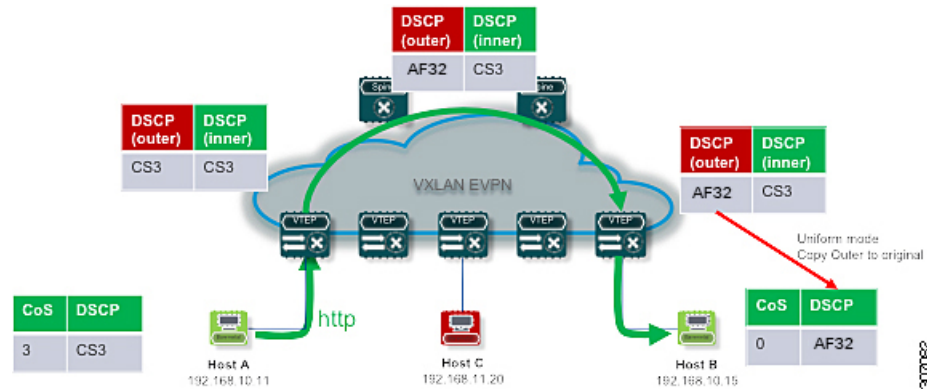
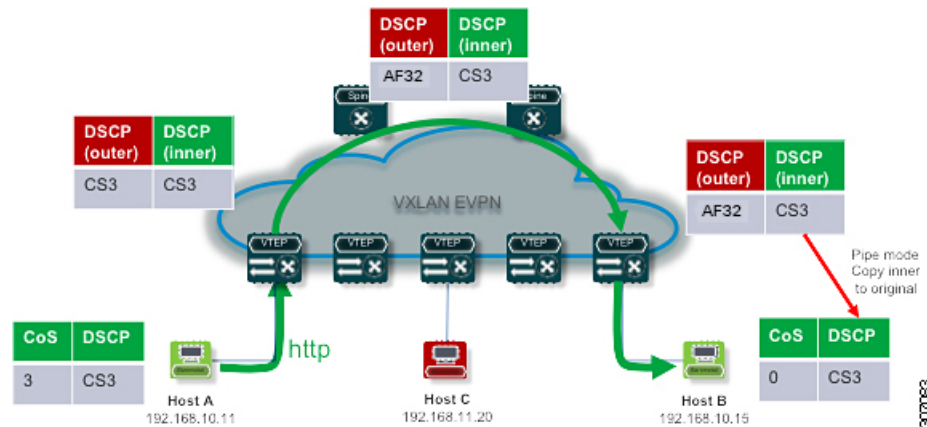


Figure 43: Pipe Mode Inner DSCP Value is Copied to Decapsulated Packet DSCP Value for Layer-3 Packet



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.
- The following limitations apply to the VXLAN QoS policies when using a Border Gateway (BGW) Spine:
 - If QoS policies are needed for intra-site BUM traffic for VNI with multicast underlay, and that multicast underlay group is also owned by a VNI defined on the BGW Spine, then the QoS policy must be applied to the NVE interface. QoS policies applied to fabric interfaces will not modify these flows since the NVE interface acts as an incoming interface.
 - If QoS policies are needed for intra-site BUM traffic for VNI with multicast underlay, and that multicast group is not owned by a VNI defined on the BGW Spine, then the QoS policy must be applied to a fabric interface. QoS policies applied to the NVE interface will not modify these flows since the NVE is not considered an incoming interface.
 - If the NVE interface of the BGW Spine owns a multicast group used for BUM traffic within the local fabric, QoS policies cannot be applied to both the fabric interfaces and NVE interface to differentiate treatment of intra-site and inter-site flows for that multicast group.

Default Settings for VXLAN QoS

The following table lists the default CoS to DSCP mapping in the ingress VTEP for Layer 2 frames:

Table 9: Default CoS to DSCP Mapping

CoS of Original Layer 2 Frame	DSCP of Outer VXLAN Header
0	0
1	8
2	16
3	26
4	32
5	46

CoS of Original Layer 2 Frame	DSCP of Outer VXLAN Header
6	48
7	56

Configuring VXLAN QoS

Configuration of VXLAN QoS is done using the MQC model. The same configuration that is used for the QoS configuration applies to VXLAN QoS. For more information about configuring QoS, see the [Cisco Nexus 9000 Series NX-OS Quality of Service Configuration Guide, Release 9.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

	Command or Action	Purpose
Step 1	configure terminal Example: <code>switch# configure terminal</code>	Enters global configuration mode.
Step 2	[no] class-map [type qos] [match-all] [match-any] class-map-name Example: <code>switch(config)# class-map type qos class1</code>	Creates or accesses the class map <i>class-map-name</i> and enters class-map mode. The <i>class-map-name</i> can contain alphabetic, hyphen, or underscore characters, and can be up to 40 characters. (match-any is the default when the no option is selected and multiple match statements are entered.)

	Command or Action	Purpose
Step 3	[no] match [access-group cos dscp precedence] {name 0-7 0-63 0-7} Example: <pre>switch(config-cmap-qos) # match dscp 26</pre>	Configures the traffic class by matching packets based on access-list, cos value, dscp values, or ip precedence value
Step 4	[no] policy-map type qos policy-map-name Example: <pre>switch(config-cmap-qos) # policy-map type qos policy</pre>	Creates or accesses the policy map named <i>policy-map-name</i> and then enters policy-map mode. The policy-map name can contain alphabetic, hyphen, or underscore characters, is case sensitive, and can be up to 40 characters.
Step 5	[no] class class-name Example: <pre>switch(config-pmap-qos) # class class1</pre>	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: <pre>switch(config-pmap-c-qos) # set qos-group 1</pre>	Sets the QoS group value to <i>qos-group-value</i> . 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: <pre>switch(config-pmap-c-qos) # exit</pre>	Exits class-map mode.
Step 8	[no] interface nve nve-interface-number Example: <pre>switch(config) # interface nve 1</pre>	Enters interface mode to configure the NVE interface.
Step 9	[no] service-policy type qos input policy-map-name Example: <pre>switch(config-if-nve) # service-policy type qos input policy1</pre>	Adds a service-policy <i>policy-map-name</i> 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: <pre>switch(config-if-nve) # qos-mode pipe</pre>	Selecting decapsulated packet priority selection and using pipe mode. Entering the no form of this command negates pipe mode and defaults to uniform mode.

Verifying the VXLAN QoS Configuration

Table 10: VXLAN QoS Verification Commands

Command	Purpose
show class map	Displays information about all configured class maps.
show policy-map	Displays information about all configured policy maps.
show running ipqos	Displays configured QoS configuration on the switch.

VXLAN QoS Configuration Examples

Ingress VTEP Classification and Marking

This example shows how to configure the **class-map type qos** command for classification matching traffic with an ACL. Enter the **policy-map type qos** command to put traffic in qos-group 1 and set the DSCP value. Enter the **service-policy type qos** command to attach to the ingress interface in the input direction to classify traffic matching the ACL.

```
access-list ACL_QOS_DSCP_CS3 permit ip any any eq 80

class-map type qos CM_QOS_DSCP_CS3
 match access-group name ACL_QOS_DSCP_CS3

policy-map type qos PM_QOS_MARKING
 class CM_QOS_DSCP_CS3
  set qos-group 1
  set dscp 24

interface ethernet1/1
 service-policy type qos input PM_QOS_MARKING
```

Transit Switch – Spine Classification

This example shows how to configure the **class-map type qos** command for classification matching DSCP 24 set on the ingress VTEP. Enter the **policy-map type qos** command to put traffic in qos-group 1. Enter the **service-policy type qos** command to attach to the ingress interface in the input direction to classify traffic matching criteria.

```
class-map type qos CM_QOS_DSCP_CS3
 match dscp 24

policy-map type qos PM_QOS_CLASS
 class CM_QOS_DSCP_CS3
  set qos-group 1

interface Ethernet 1/1
 service-policy type qos input PM_QOS_CLASS
```

Egress VTEP Classification and Marking

This example shows how to configure the **class-map type qos** command for classification matching traffic by DSCP value. Enter the **policy-map type qos** to place traffic in qos-group 1 and mark CoS value in outgoing frames. The **service-policy type qos** command is applied to the NVE interface in the input direction to classify traffic coming out of the VXLAN tunnel.

```
class-map type qos CM_QOS_DSCP_CS3
  match dscp 24

policy-map type qos PM_QOS_MARKING
  class CM_QOS_DSCP_CS3
    set qos-group 1
    set cos 3

interface nve 1
  service-policy type qos input PM_QOS_MARKING
```

Queuing

This example shows how to configure the **policy-map type queueing** command for traffic in qos-group 1. Assigning 50% of the available bandwidth to q1 mapped to qos-group 1 and attaching policy in the output direction to all ports using the **system qos** command.

```
policy-map type queueing PM_QUEUEING
class type queueing c-out-8q-q7
  priority level 1
class type queueing c-out-8q-q6
  bandwidth remaining percent 0
class type queueing c-out-8q-q5
  bandwidth remaining percent 0
class type queueing c-out-8q-q4
  bandwidth remaining percent 0
class type queueing c-out-8q-q3
  bandwidth remaining percent 0
class type queueing c-out-8q-q2
  bandwidth remaining percent 0
class type queueing c-out-8q-q1
  bandwidth remaining percent 50
class type queueing c-out-8q-q-default
  bandwidth remaining percent 50

system qos
  service-policy type queueing output PM_QUEUEING
```



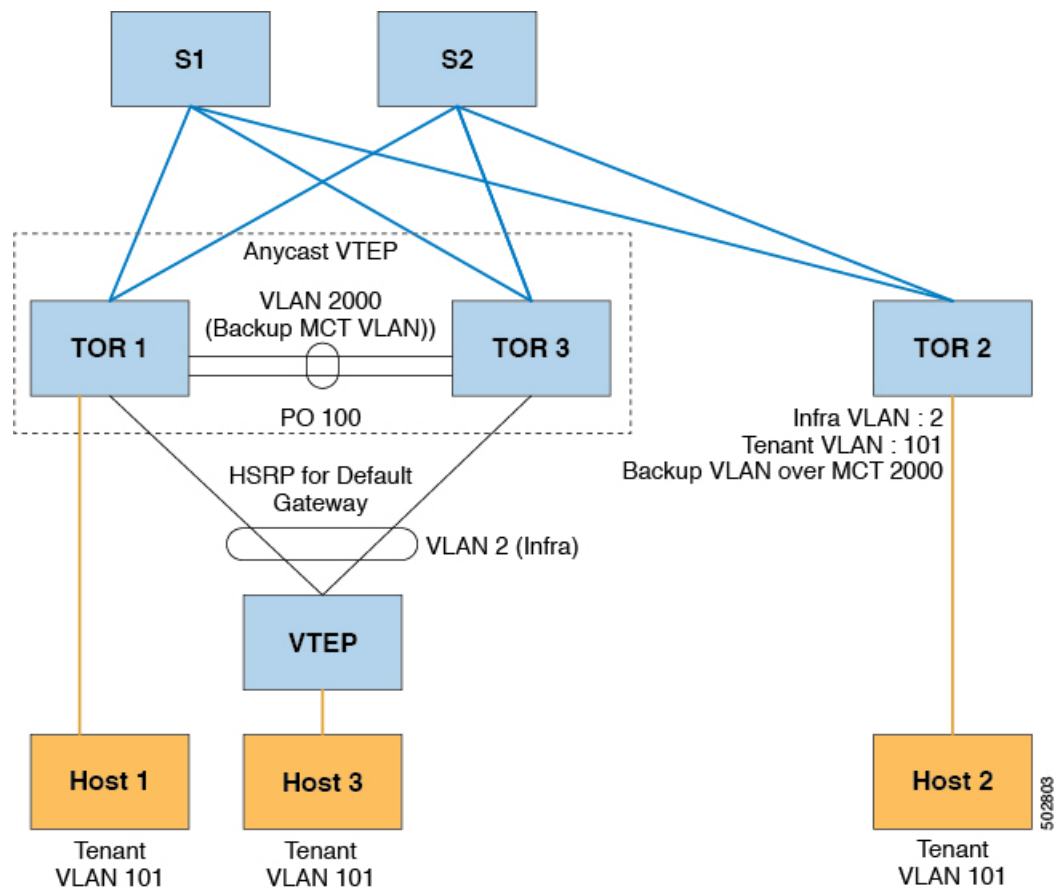
APPENDIX **A**

VXLAN Bud Node Over VPC

- [VXLAN Bud Node Over VPC Overview, on page 217](#)
- [VXLAN Bud Node Over vPC Topology Example, on page 218](#)

VXLAN Bud Node Over VPC Overview

Figure 44: Underlay Network Based on PIM-SM and OSPF





Note For bud-node topologies, the source IP of the VTEP behind VPC must be in the same subnet as the infra VLAN. This SVI should have proxy ARP enabled. For example:

```
Interface Vlan2
ip proxy-arp
```



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

For Cisco Nexus 9200, 9300-EX, and 9300-FX switches, use the **system nve infra-vlans** command to configure any VLANs that are used as infra-VLANs.

VXLAN Bud Node Over vPC Topology Example

- Enable the required features:

```
feature ospf
feature pim
feature interface-vlan
feature vn-segment-vlan-based
feature hsrp
feature lacp
feature vpc
feature nv overlay
```

- Configuration for PIM anycast RP.

In this example, 1.1.1.1 is the anycast RP address.

```
ip pim rp-address 1.1.1.1 group-list 225.0.0.0/8
```

- VLAN configuration

In this example, tenant VLANs 101-103 are mapped to vn-segments.

```
vlan 1-4,101-103,2000
vlan 101
  vn-segment 10001
vlan 102
  vn-segment 10002
vlan 103
  vn-segment 10003
```

- vPC configuration

```
vpc domain 1
 peer-switch
 peer-keepalive destination 172.31.144.213
 delay restore 180
 peer-gateway
 ipv6 nd synchronize
 ip arp synchronize
```

- Infra VLAN SVI configuration

```
interface Vlan2
 no shutdown
 no ip redirects
 ip proxy-arp
 ip address 10.200.1.252/24
 no ipv6 redirects
 ip router ospf 1 area 0.0.0.0
 ip pim sparse-mode
 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
```

Interface	VNI	Multicast-group	State	Mode	Type	[BD/VRF]	Flags
nve1	10001	225.1.1.1	Up	DP	L2	[101]	
nve1	10002	225.1.1.1	Up	DP	L2	[102]	
nve1	10003	225.1.1.1	Up	DP	L2	[103]	

```
tor1# sh nve peers
Interface Peer-IP      State LearnType Uptime  Router-Mac
-----
nve1      10.200.1.1            Up    DP           00:07:23 n/a
nve1      10.200.1.2            Up    DP           00:07:18 n/a
nve1      102.102.102.102      Up    DP           00:07:23 n/a
```

```
tor1# sh ip mroute 225.1.1.1
IP Multicast Routing Table for VRF "default"
```

```
(*, 225.1.1.1/32), uptime: 00:07:41, ip pim nve static igmp
Incoming interface: Ethernet2/1, RPF nbr: 10.1.5.2
```

```

Outgoing interface list: (count: 3)
  Vlan2, uptime: 00:07:23, igmp
  Vlan2000, uptime: 00:07:31, static
  nve1, uptime: 00:07:41, nve

(10.200.1.1/32, 225.1.1.1/32), uptime: 00:07:40, ip mrib pim nve
Incoming interface: Vlan2, RPF nbr: 10.200.1.1
Outgoing interface list: (count: 3)
  Vlan2, uptime: 00:07:23, mrib, (RPF)
  Vlan2000, uptime: 00:07:31, mrib
  nve1, uptime: 00:07:40, nve

(10.200.1.2/32, 225.1.1.1/32), uptime: 00:07:41, ip mrib pim nve
Incoming interface: Vlan2, RPF nbr: 10.200.1.2
Outgoing interface list: (count: 3)
  Vlan2, uptime: 00:07:23, mrib, (RPF)
  Vlan2000, uptime: 00:07:31, mrib
  nve1, uptime: 00:07:41, nve

(99.99.99.99/32, 225.1.1.1/32), uptime: 00:07:41, ip mrib pim nve
Incoming interface: loopback0, RPF nbr: 99.99.99.99
Outgoing interface list: (count: 3)
  Vlan2, uptime: 00:07:23, mrib
  Vlan2000, uptime: 00:07:31, mrib
  Ethernet2/5, uptime: 00:07:39, pim
(102.102.102.102/32, 225.1.1.1/32), uptime: 00:07:40, ip mrib pim nve
Incoming interface: Ethernet2/1, RPF nbr: 10.1.5.2
Outgoing interface list: (count: 1)
  nve1, uptime: 00:07:40, nve

```

tor1# sh vpc

Legend:

- local vPC is down, forwarding via vPC peer-link

```

vPC domain id          : 1
Peer status             : peer adjacency formed ok
vPC keep-alive status   : peer is alive
Configuration consistency status : success
Per-vlan consistency status : success
Type-2 consistency status : success
vPC role                : secondary, operational primary
Number of vPCs configured : 4
Peer Gateway            : Enabled
Dual-active excluded VLANs : -
Graceful Consistency Check : Enabled
Auto-recovery status     : Disabled
Delay-restore status     : Timer is off.(timeout = 180s)
Delay-restore SVI status : Timer is off.(timeout = 10s)

```

vPC Peer-link status

id	Port	Status	Active vlans
1	Po100	up	1-4,101-103,2000

vPC status

id	Port	Status	Consistency	Reason	Active vlans
1	Po1	up	success	success	2
2	Po2	up	success	success	2

```
tor1# sh vpc consistency-parameters global
```

Legend:

Type 1 : vPC will be suspended in case of mismatch

Name	Type	Local Value	Peer Value
Vlan to Vn-segment Map	1	3 Relevant Map(s)	3 Relevant Map(s)
STP Mode	1	Rapid-PVST	Rapid-PVST
STP Disabled	1	None	None
STP MST Region Name	1	""	""
STP MST Region Revision	1	0	0
STP MST Region Instance to	1		
VLAN Mapping			
STP Loopguard	1	Disabled	Disabled
STP Bridge Assurance	1	Enabled	Enabled
STP Port Type, Edge	1	Normal, Disabled,	Normal, Disabled,
BPDUFILTER, Edge BPDUGuard		Disabled	Disabled
STP MST Simulate PVST	1	Enabled	Enabled
Nve Oper State, Secondary	1	Up, 99.99.99.99, DP	Up, 99.99.99.99, DP
IP, Host Reach Mode			
Nve Vni Configuration	1	10001-10003	10001-10003
Interface-vlan admin up	2	2,2000	2,2000
Interface-vlan routing	2	1-4,2000	1-4,2000
capability			
Allowed VLANs	-	1-4,101-103,2000	1-4,101-103,2000
Local suspended VLANs	-	-	



APPENDIX **B**

DHCP Relay in VXLAN BGP EVPN

- [DHCP Relay in VXLAN BGP EVPN Overview, on page 223](#)
- [DHCP Relay in VXLAN BGP EVPN Example, on page 224](#)
- [Configuring VPC Peers Example, on page 240](#)
- [vPC VTEP DHCP Relay Configuration Example, on page 242](#)

DHCP Relay in VXLAN BGP EVPN Overview

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

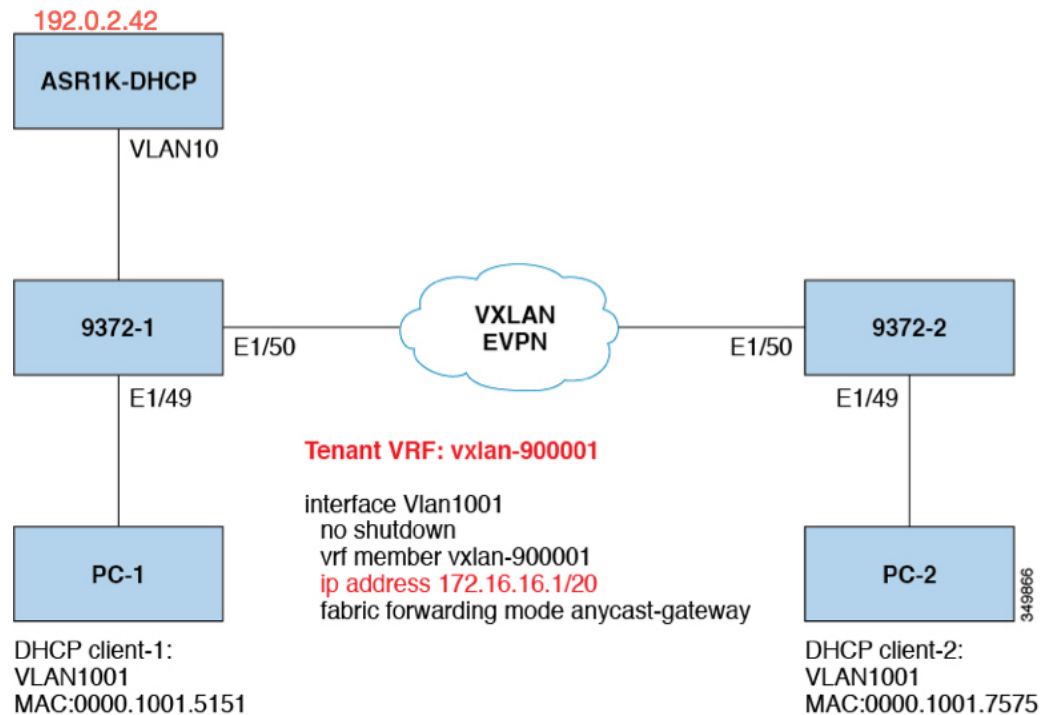
(Defined in RFC#3527.)

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 45: Example Topology



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

```
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
 defaultrouter 172.16.16.1
```

Basic VXLAN BGP EVPN Configuration

• 9372-1

```
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 mcde anycast-gateway

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 mcde 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 192.1.34.2/24
  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
```

```

log-neighbor-changes
neighbor 1.1.1.1 remote-as 65535
  update-source loopback0
  address-family l2vpn evpn
    send-community both
vrf vxlen-900001
  address-family ipv4 unicast
    network 22.22.22.22/32

    advertise l2vpn evpn
evpn
  vni 2001001 12

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

```

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
!Time: Mon Aug 24 07:51:16 2015

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)I1(3)
feature dhcp

service dhcp
ip dhcp relay
ip dhcp relay information option
ip dhcp relay information option vpn
ipv6 dhcp relay

interfaoe 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

```



Note Ethanalzyer 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

```



```

Length: 4
Link selection: 172.16.16.0 (172.16.16.0)

ASR1K-DHCP# sh ip dhcp bin
Bindings from all pools not associated with VRF:
IP address ClientID/ Lease expiration Type State Interface
      Hardware address/
      User name

Bindings from VRF pool vxlan900001:
IP address ClientID/ Lease expiration Type State Interface
      Hardware address/
      User name
172.16.16.10 0100.0010.0175.75 Aug 25 2015 09:21 AM Automatic Active GigabitEthernet2/1/0
172.16.16.11 0100.0010.0151.51 Aug 25 2015 08:54 AM Automatic Active GigabitEthernet2/1/0

9372-1# sh ip route vrf vxlan900001
IP Route Table for VRF "vxlan900001"
'*' denotes best ucast nexthop
'***' denotes best mcast nexthop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

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
22.22.22.22/32, 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
- 172.16.16.11 ping statistics -
1 packets transmitted, 1 packets received, 0.00% packet loss
round-trip min/avg/max = 0.846/0.845/0.846 ms

9372-1# ping 172.16.16.10 vrf vxlan900001 count 1
PING 172.16.16.10 (172.16.16.10): 56 data bytes
64 bytes from 172.16.16.10: icmp_seq=0 ttl=62 time=0.874 ms
- 172.16.16.10 ping statistics -
1 packets transmitted, 1 packets received, 0.00% packet loss
round-trip min/avg/max = 0.874/0.873/0.874 ms

```

Client on Tenant VRF (SVI X) and Server on the Same Tenant VRF (SVI Y)

Put DHCP server (192.1.42.3) 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 loopback1 and we need to make sure loopback1 is reachable from everywhere of VRF vxlan-900001.

```
9372-1# sh run int lo1

!Command: show running-config interface loopback1
!Time: Mon Aug 24 09:18:53 2015

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
!4ip 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 09:31:38.129952 11.11.11.11 -> 192.1.42.3 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
20150824 09:31:40.130990 0.0.0.0 -> 255.255.255.255 DHCP DHCP Request - Transaction ID
0x860cd13
20150824 09:31:40.131457 11.11.11.11 -> 192.1.42.3 DHCP DHCP Request - Transaction ID
0x860cd13
20150824 09:31:40.132009 192.1.42.3 -> 11.11.11.11 DHCP DHCP ACK - Transaction ID
0x860cd13
20150824 09:31:40.132268 172.16.16.1 -> 172.16.16.11 DHCP DHCP ACK - TransactionID
0x860cd13

```



Note 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 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)

```

```

ASR1K-DHCP# sh ip dhcp bin
Bindings from all pools not associated with VRF:
IP address ClientID/Lease expiration Type State Interface
      Hardware address/
      User name

```

```

Bindings from VRF pool vxlan-900001:
IP address ClientID/Lease expiration Type State Interface
      Hardware address/
      User name

172.16.16.10 0100.0010.0175.75 Aug 25 2015 10:02 AM Automatic Active GigabitEthernet2/1/0
172.16.16.11 0100.0010.0151.51 Aug 25 2015 09:50 AM Automatic Active GigabitEthernet2/1/0

9372-1# sh ip route vrf vxlan-900001
IP Route Table for VRF "vxlan-900001"
'*' denotes best ucast nexthop
'***' denotes best mcast nexthop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

11.11.11.11/32, ubest/mbest: 2/0, attached
    *via 11.11.11.11, Lo1, [0/0], 19:13:56, local
    *via 11.11.11.11, Lo1, [0/0], 19:13:56, direct
22.22.22.22/32, ubest/mbest: 1/0
    *via 2.2.2.2%default, [200/0], 19:13:56, bgp65535,internal, tag 65535 (evpn)segid:
900001 tunnelid: 0x2020202
encap: VXLAN
172.16.16.0/20, ubest/mbest: 1/0, attached
    *via 172.16.16.1, Vlan1001, [0/0], 19:13:56, direct
172.16.16.1/32, ubest/mbest: 1/0, attached
    *via 172.16.16.1, Vlan1001, [0/0], 19:13:56, local
172.16.16.10/32, ubest/mbest: 1/0
    *via 2.2.2.2%default, [200/0], 00:01:27, bgp65535,
internal, tag 65535 (evpn)segid: 900001 tunnelid: 0x2020202
encap: VXLAN
172.16.16.11/32, ubest/mbest: 1/0, attached
    *via 172.16.16.11, Vlan1001, [190/0], 00:13:56, hmm
192.1.42.0/24, ubest/mbest: 1/0, attached
    *via 192.1.42.1, Vlan10, [0/0], 00:36:08, direct
192.1.42.1/32, ubest/mbest: 1/0, attached
    *via 192.1.42.1, Vlan10, [0/0], 00:36:08, local
9372-1# ping 172.16.16.10 vrf vxlan-900001 cou 1
PING 172.16.16.10 (172.16.16.10): 56 data bytes
64 bytes from 172.16.16.10: icmp_seq=0 ttl=62 time=0.808 ms
- 172.16.16.10 ping statistics -
1 packets transmitted, 1 packets received, 0.00% packet loss
round-trip min/avg/max = 0.808/0.808/0.808 ms

9372-1# ping 172.16.16.11 vrf vxlan-900001 cou 1
PING 172.16.16.11 (172.16.16.11): 56 data bytes
64 bytes from 172.16.16.11: icmp_seq=0 ttl=63 time=0.872 ms
- 172.16.16.11 ping statistics -
1 packets transmitted, 1 packets received, 0.00% packet loss
round-trip min/avg/max = 0.872/0.871/0.872 ms

```

Client on Tenant VRF (VRF X) and Server on Different Tenant VRF (VRF Y)

The DHCP server is placed into another tenant VRF vxlan-900002 so that DHCP response packets can access the original relay agent. We use loopback2 to avoid any anycast ip address that is used as the source address for the DHCP relay packets.

```

9372-1# sh run int vl 10
!Command: show runningconfig interface Vlan10
!Time: Tue Aug 25 08:48:22 2015

```

```

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 runningconfig interface loopback2
!Time: Tue Aug 25 08:48:57 2015
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 runningconfig interface loopback2
!Time: Tue Aug 25 08:48:44 2015
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 cou 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) Ii (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

```

!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 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 0x3eebcca
20150825 08:59:35.758878 33.33.33.33 -> 192.1.42.3 DHCP DHCP Discover - Transaction ID
0x3eebcca
20150825 08:59:37.759560 192.1.42.3 -> 33.33.33.33 DHCP DHCP Offer - Transaction ID
0x3eebcca
20150825 08:59:37.759905 172.16.16.1 -> 172.16.16.11 DHCP DHCP Offer - Transaction ID
0x3eebcca
20150825 08:59:37.760313 0.0.0.0 -> 255.255.255.255 DHCP DHCP Request - Transaction ID
0x3eebcca
20150825 08:59:37.760733 33.33.33.33 -> 192.1.42.3 DHCP DHCP Request - Transaction ID
0x3eebcca
20150825 08:59:37.761297 192.1.42.3 -> 33.33.33.33 DHCP DHCP ACK - Transaction ID
0x3eebcca
20150825 08:59:37.761554 172.16.16.1 -> 172.16.16.11 DHCP DHCP ACK - Transaction ID
0x3eebcca

```

- 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: 0x3eebcca
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-iughes_01:51:51 (00:00:10:01:51:51)

```

```

Client hardware address padding: 00000000000000000000
Server host name not given
Boot file name not given
Magic cookie: DHCP
Option: (53) DHCP Message Type
  Length: 1
  DHCP: Discover (1)
Option: (55) Parameter Request List
Option: (61) client identifier
Option: (82) Agent Information option
  Length: 47
Option 82 Suboption: (1) Agent circuit W
Option 82 suboption: (2) Agent Remote 10
Option 82 suboption: (151) VRF name/VPN ID
Option 82 Suboption: (11) Server ID Override
  Length: 4
  Server ID Override: 172.16.16.1 (172.16.16.1)
Option 82 Suboption: (5) Link selection
  Length: 4
  Link selection: 172.16.16.0 (172.16.16.0)

```

Client on Tenant VRF and Server on Non-Default Non-VXLAN VRF

The DHCP server is placed into the management VRF and is reachable through the 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
- 10.122.164.147 ping statistics -
1 packets transmitted, 1 packets received, 0.00% packet loss
round-trip min/avg/max = 1.024/1.024/1.024 ms

9372-2# sh run int m0
!Command: show running-config interface mgmt0
!Time: Tue Aug 25 09:17:47 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
- 10.122.164.147 ping statistics -
1 packets transmitted, 1 packets received, 0.00% packet loss
round-trip min/avg/max = 1.03/1.03/1.03 ms

```

DHCP Relay Configuration

- 9372-1


```

9372-1# sh run dhcp 9372-2# sh run dhcp

!Command: show running-config dhcp
!Time: Mon Aug 24 08:26:00 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 09:30:56.216137 10.122.164.147 -> 10.122.165.134 DHCP DHCP Offer - Transaction
ID 0x28a8606d

```

```
20150825 09:30:56.217444 10.122.165.134 -> 10.122.164.147 DHCP DHCP Request - Transaction
ID 0x28a8606d
20150825 09:30:56.218207 10.122.164.147 -> 10.122.165.134 DHCP DHCP ACK - Transaction
ID 0x28a8606d
```

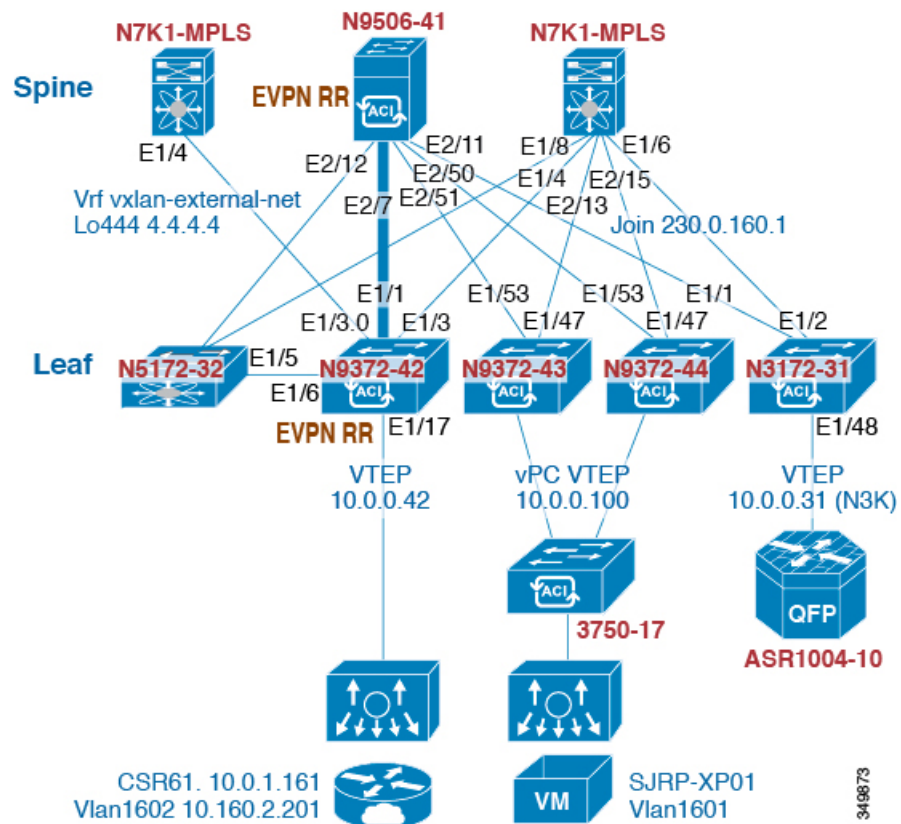
- DHCP Discover packet 9372-1 sent to DHCP server.

giaddr is set to 10.122.165.134 (mgmt0) and suboptions 5/11/151 are set accordingly.

```
Bootstrap Protocol
Message type: Boot Request (1)
Hardware type: Ethernet (0x01)
Hardware address length: 6
Hops: 1
Transaction ID: 0x28a8606d
Seconds elapsed: 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.

```
interface Vlan1601
  vrf member evpn-tenant-kk1
  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**.

```
interface Vlan1600
  vrf member evpn-tenant-kk1
  ip forward
```

- Create the routing VLAN/SVI for the VPC VRF.



Note Only required for VPC VTEP.

```
Vlan 1605
interface Vlan1605
  vrf member evpn-tenant-kk1
  ip address 10.160.5.43/24
  ip router ospf 1 area 0.0.0.41
```

- Create the VRF routing.



Note Only required for VPC VTEP.

```
router ospf 1
  vrf evpn-tenant-kk1
  router-id 10.160.5.43
```

vPC VTEP DHCP Relay Configuration Example

To address a need to configure a VLAN that is allowed across the 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
```




APPENDIX **C**

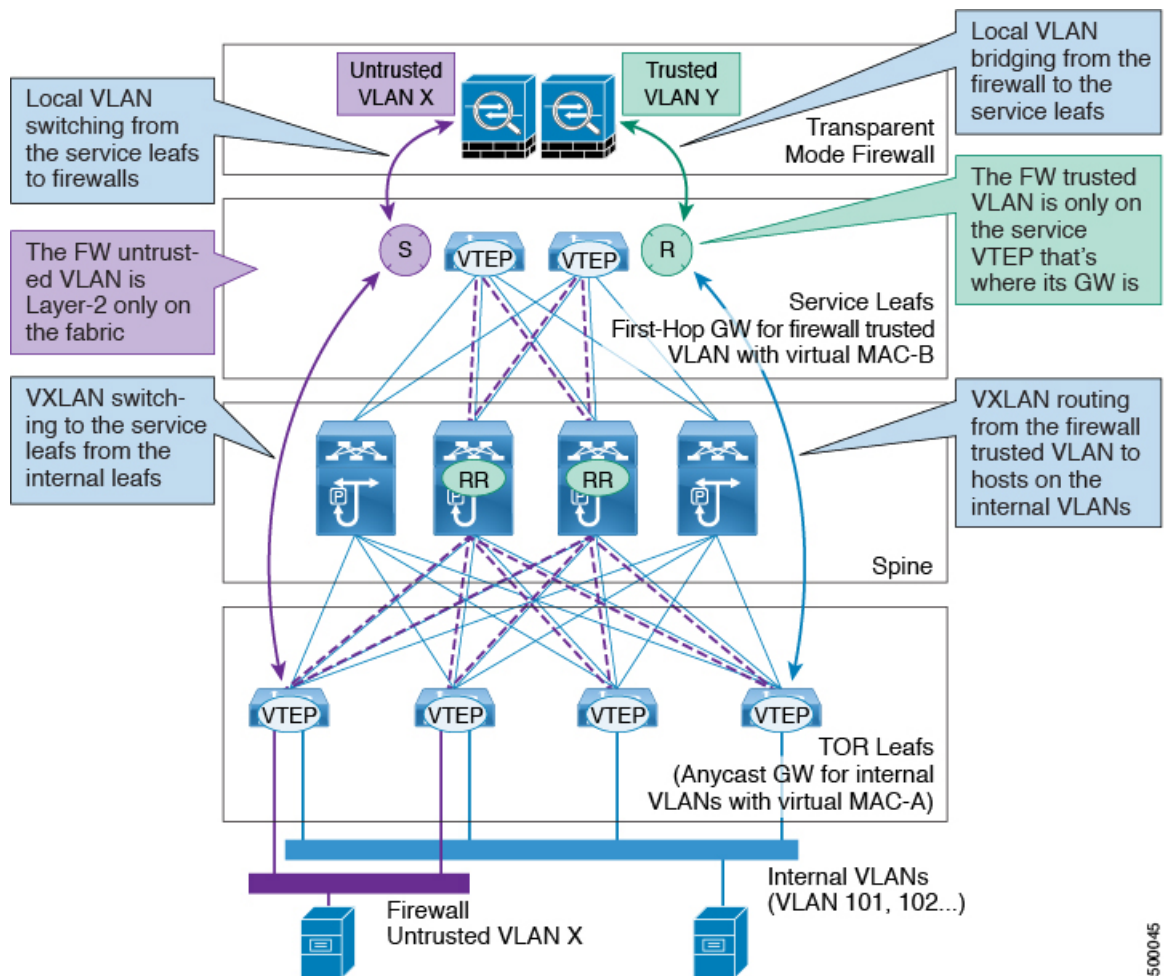
EVPN with Transparent Firewall Insertion

- [Overview of EVPN with Transparent Firewall Insertion, on page 245](#)
- [EVPN with Transparent Firewall Insertion Example, on page 247](#)
- [Show Command Examples, on page 250](#)

Overview of 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.

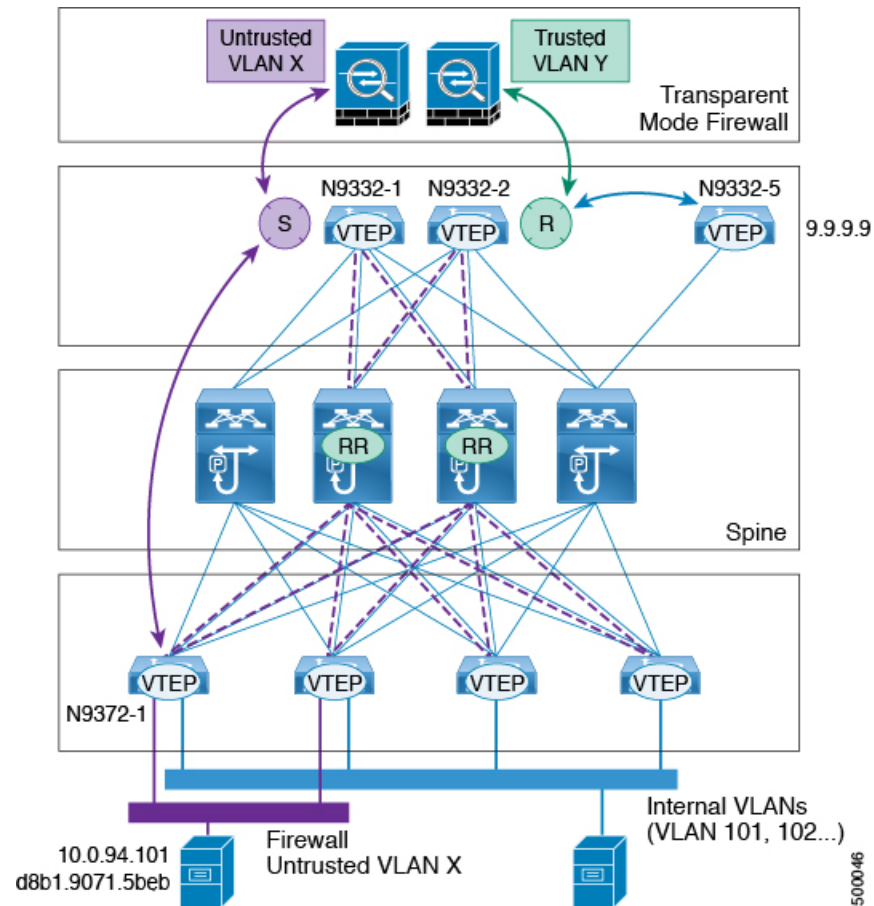
The service leafs that are connected to the firewall have Layer 2 VNI VLAN X, non-VXLAN VLAN Y, and SVI Y with a HSRP gateway.



Note 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



- Note**
- 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 l2
rd auto
route-target import auto
route-target export auto

```

- Service leaf 1 configuration

```

vlan 94
description untrusted_vlan
vn-segment 100094

vlan 95
description trusted_vlan

vpc domain 10
peer-switch
peer-keepalive destination 10.1.59.160
peer-gateway
auto-recovery
ip arp synchronize

interface Vlan2
description vpc_backup_svi_for_overlay
no shutdown
no ip redirects
ip address 10.10.60.17/30
no ipv6 redirects
ip router ospf 100 area 0.0.0.0
ip ospf bfd
ip pim sparsemode

interface Vlan95
description SVI_for_trusted_vlan
no shutdown
mtu 9216
vrf member Ten-1
no ip redirects
ip address 10.0.94.2/24
hsrp 0
preempt
priority 255
ip 10.0.94.1

interface nve1
member vni 100094

```

```
mcast-group 239.1.1.1

router bgp 64500
  routerid 1.1.2.1
  neighbor 1.1.1.1 remote-as 64500
  address-family l2vpn evpn
    send-community extended
  neighbor 1.1.1.2 remote-as 64500
  address-family l2vpn evpn
    send-community extended
  vrf Ten-1
    address-family ipv4 unicast
      network 10.0.94.0/24 /*advertise /24 for SVI 95 subnet; it is not VXLAN anymore*/
      advertise l2vpn evpn

evpn
  vni 100094 l2
  rd auto
  route-target import auto
  route-target export auto
```

- Service leaf 2 configuration

```
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_vla
  no shutdown
  mtu 9216
  vrf member Ten-1
  no ip redirects
  ip address 10.0.94.3/24
  hsrp 0
    preempt
    priority 255
  ip 10.0.94.1

interface nve1
  member vni 100094
  mcastgroup 239.1.1.1

router bgp 64500
  router-id 1.1.2.1
```

```

neighbor 1.1.1.1 remote-as 64500
address-family l2vpn evpn
send-community extended
neighbor 1.1.1.2 remote-as 64500
address-family l2vpn evpn
send-community extended
vrf Ten-1
address-family ipv4 unicast
network 10.0.94.0/24 /*advertise /24 for SVI 95 subnet; it is not VXLAN anymore*/
advertise l2vpn evpn

evpn
vni 100094 l2
rd auto
route-target import auto
route-target export auto

```

Show Command Examples

- Display information about 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 ucast nexthop
'***' denotes best mcast nexthop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>
9.9.9.9/32, ubest/mbest: 1/0
*via 1.1.2.7%default, [200/0], 02:57:27, bgp64500,internal, tag 65000 (evpn) segid:
10011
tunnelid: 0x1
010207 encap: 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 ucast nexthop
'***' denotes best mcast nexthop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>
10.0.94.0/24, ubest/mbest: 1/0
*via 10.100.5.0, [20/0], 03:14:27, bgp65000,external, tag 6450
```




APPENDIX **D**

IPv6 Across a VXLAN EVPN Fabric

- [Overview of IPv6 Across a VXLAN EVPN Fabric, on page 253](#)
- [Configuring IPv6 Across a VXLAN EVPN Fabric Example, on page 253](#)
- [Show Command Examples, on page 256](#)

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.

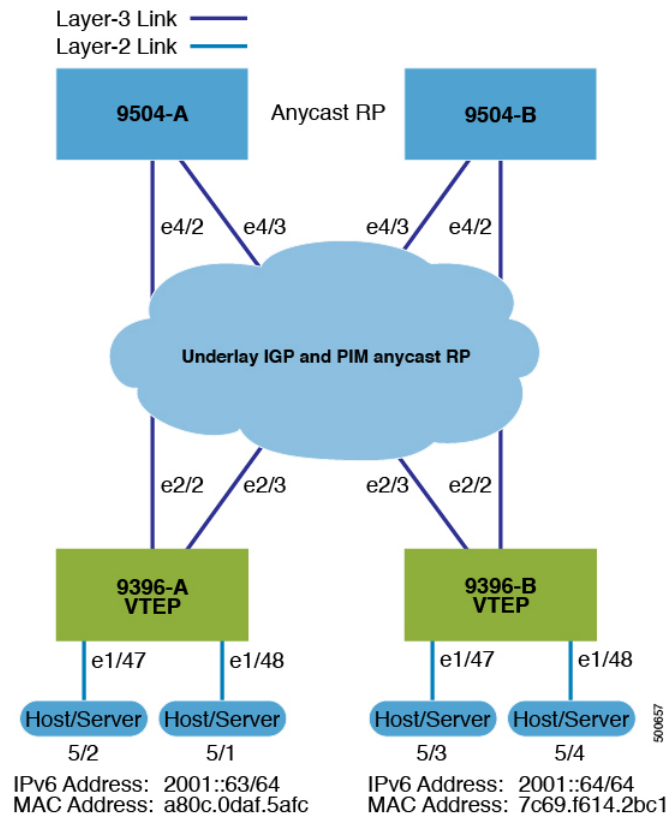
To enable IPv6 across a VXLAN EVPN fabric, the IPv6 address family is included in VRF, BGP, and EVPN. IPv6 routes are initiated in the tenant VRF IPv6 unicast address-family on a VTEP and are advertised in the VXLAN fabric through the L2VPN EVPN address family as EVPN route-type 2 or 5.



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

Configuring IPv6 Across a VXLAN EVPN Fabric Example

Topology for the example:



Note In the example:

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

- Configure the Layer 2 VLAN.

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

- Configure the VLAN for L3 VNI .

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

- Define the anycast gateway MAC.

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


- 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 vni20000
  suppress-arp
  mcast-group 225.1.1.1
  member vni 20001
  suppress-arp
  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					

```
00:01:13 7c69.f614.2bc1 50 icmpv6 Vlan10
```

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

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



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

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

```
9396-B_VTEP# show bgp l2vpn evpn 2001::64
BGP routing table information for VRF default, address family L2VPN EVPN
Route Distinguisher: 198.19.0.15:34180 (L2VNI 10010)
BGP routing table entry for [2]:[0]:[0]:[48]:[7c69.f614.2bc1]:[128]:[2001::64]/368,
version 678
Paths: (1 available, best #1)
Flags: (0x00010a) on xmit-list, is not in l2rib/evpn

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

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

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

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

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

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

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



INDEX

A

address-family ipv4 unicast [86, 90](#)
 address-family ipv6 unicast [86, 90](#)
 address-family l2vpn evpn [90, 94](#)
 advertise [90](#)
 associate- vrf [83](#)

C

clear nve peer-ip [63–64](#)
 clear nve peers [63–64](#)
 clear nve vni [64](#)

E

evpn [90](#)

F

fabric forwarding [83](#)
 fabric forwarding anycast-gateway-mac [89](#)
 fabric forwarding mode anycast-gateway [89](#)
 feature nv overlay [38, 85](#)
 feature vn-segment [85](#)
 feature vn-segment-vlan-based [38](#)
 force [32–33](#)

H

hardware access-list team region arp-ether double-wide [23, 77, 94](#)
 host-reachability protocol bgp [83, 89, 92](#)
 how interface [34](#)

I

ingress-replication protocol bgp [39, 92–93](#)
 ingress-replication protocol static [10, 39](#)
 interface [89](#)
 interface nve [37–39](#)
 interface nve 1 [95](#)
 ip address [88](#)

M

mac address-table static [37](#)
 mcast-group [37, 89](#)
 member vni [37, 39, 83, 89, 92, 95](#)

N

neighbor [89, 94](#)
 no feature nv overlay [95](#)
 no feature vn-segment-vlan-based [95](#)
 no nv overlay evpn [95](#)
 nv overlay evpn [83, 85](#)

O

overlay-encapsulation vxlan-with-tag [42](#)

P

peer-ip [39](#)

R

rd auto [86, 90](#)
 retain route-target all [94](#)
 route-map permitall out [94](#)
 route-map permitall permit 10 [93](#)
 route-target both auto [86](#)
 route-target both auto evpn [86–87](#)
 route-target export auto [91](#)
 route-target import auto [90](#)
 router bgp [83, 89, 93](#)
 router-id [89](#)

S

send-community extended [90, 94](#)
 set ip next-hop unchanged [93](#)
 show bgp l2vpn evpn [83, 98, 118](#)
 show bgp l2vpn evpn summary [83, 118](#)
 show interface [36, 64](#)
 show ip arp suppression-cache [98](#)
 show ip arp suppression-cache detail [117](#)

show l2route evpn fl all [99](#)
 show l2route evpn imet all [99](#)
 show l2route evpn mac [99](#)
 show l2route evpn mac all [119](#)
 show l2route evpn mac-ip all [99, 119](#)
 show l2route evpn mac-ip all detail [99](#)
 show l2route topology [99](#)
 show logging level nve [63–64](#)
 show mac address-table static interface nve [38](#)
 show mac address-table static interface nve 1 [64](#)
 show nve interface [63–64](#)
 show nve peers [63–64, 117](#)
 show nve vni [63–64, 83, 117](#)
 show nve vni ingress-replication [64](#)
 show nve vni summary [83](#)
 show nve vrf [98](#)
 show nve vxlan-params [64](#)
 show run interface nve [63–64](#)
 show run track [65](#)
 show tech-support nve [63–64](#)
 show tech-support vxlan [63–64](#)
 show track [65](#)
 show vxlan interface [65, 98, 118](#)

show vxlan interface | count [65, 99](#)
 source-interface [37, 39](#)
 source-interface config [22, 76](#)
 source-interface hold-down-time [22, 76](#)
 spanning-tree bpduguard enable [42](#)
 suppress-arp [83, 95](#)
 suppress-mac-route [85](#)
 switchport access vlan [41](#)
 switchport mode dot1q-tunnel [41](#)
 switchport mode trunk [35](#)
 switchport vlan mapping [33, 36](#)
 switchport vlan mapping all [34](#)
 switchport vlan mapping enable [33, 36](#)
 system vlan nve-overlay id [84](#)

V

vlan [30, 86, 88](#)
 vn-segment [30, 86, 88](#)
 vni [86, 88, 90](#)
 vrf [90](#)
 vrf context [83, 86, 88](#)
 vrf member [88](#)