



Cisco Nexus 9000 Series NX-OS VXLAN Configuration Guide, Release 9.2(x)

First Published: 2018-07-18

Last Modified: 2021-03-22

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)

© 2018–2021 Cisco Systems, Inc. All rights reserved.



CONTENTS

PREFACE

Preface	xiii
Audience	xiii
Document Conventions	xiii
Related Documentation for Cisco Nexus 9000 Series Switches	xiv
Documentation Feedback	xiv
Communications, Services, and Additional Information	xiv

CHAPTER 1

New and Changed Information	1
New and Changed Information	1

CHAPTER 2

Overview	3
Licensing Requirements	3
Supported Platforms	3
VXLAN Overview	3
Cisco Nexus 9000 as Hardware-Based VXLAN Gateway	4
VXLAN Encapsulation and Packet Format	4
VXLAN Tunnel	5
VXLAN Tunnel Endpoint	5
Underlay Network	5
Overlay Network	5
Distributed Anycast Gateway	5
Control Plane	6

CHAPTER 3

Configuring VXLAN	9
Guidelines and Limitations for VXLAN	9
Considerations for VXLAN Deployment	15

vPC Considerations for VXLAN Deployment	18
Network Considerations for VXLAN Deployments	22
Considerations for the Transport Network	23
Considerations for Tunneling VXLAN	24
Configuring VXLAN	25
Enabling VXLANs	25
Mapping VLAN to VXLAN VNI	25
Creating and Configuring an NVE Interface and Associate VNIs	26
Configuring a VXLAN VTEP in vPC	26
Configuring Static MAC for VXLAN VTEP	29
Disabling VXLANs	29
Configuring BGP EVPN Ingress Replication	30
Configuring Static Ingress Replication	30

CHAPTER 4
Configuring the Underlay 33

IP Fabric Underlay	33
Underlay Considerations	33
Unicast routing and IP addressing options	35
OSPF Underlay IP Network	36
IS-IS Underlay IP Network	41
eBGP Underlay IP Network	47
Multicast Routing in the VXLAN Underlay	51

CHAPTER 5
Configuring VXLAN BGP EVPN 65

About VXLAN BGP EVPN	65
About RD Auto	65
About Route-Target Auto	65
Guidelines and Limitations for VXLAN BGP EVPN	66
Configuring VXLAN BGP EVPN	70
Enabling VXLAN	70
Configuring VLAN and VXLAN VNI	70
Configuring VRF for VXLAN Routing	71
Configuring SVI for Core-facing VXLAN Routing	72
Configuring SVI for Host-Facing VXLAN Routing	73

Configuring the NVE Interface and VNIs Using Multicast	73
Configuring VXLAN EVPN Ingress Replication	74
Configuring BGP on the VTEP	75
Configuring iBGP for EVPN on the Spine	77
Configuring eBGP for EVPN on the Spine	78
Suppressing ARP	79
Disabling VXLANs	80
Duplicate Detection for IP and MAC Addresses	80
Verifying the VXLAN BGP EVPN Configuration	82
Example of VXLAN BGP EVPN (iBGP)	83
Example of VXLAN BGP EVPN (eBGP)	94
Example Show Commands	107

CHAPTER 6

Configuring External VRF Connectivity and Route Leaking 109

Configuring External VRF Connectivity	109
About External Layer-3 Connectivity for VXLAN BGP EVPN Fabrics	109
Guidelines and Limitations for External VRF Connectivity and Route Leaking	109
VXLAN BGP EVPN - VRF-lite brief	109
Configuring VXLAN BGP EVPN with eBGP for VRF-lite	110
VXLAN BGP EVPN - Default-Route, Route Filtering on External Connectivity	114
Configuring VXLAN BGP EVPN with OSPF for VRF-lite	120
Configuring Route Leaking	123
About Centralized VRF Route-Leaking for VXLAN BGP EVPN Fabrics	123
Guidelines and Limitations for Centralized VRF Route-Leaking	123
Centralized VRF Route-Leaking Brief - Specific Prefixes Between Custom VRF	123
Configuring Centralized VRF Route-Leaking - Specific Prefixes between Custom VRF	124
Configuring VRF Context on the Routing-Block VTEP	124
Configuring the BGP VRF instance on the Routing-Block	125
Example - Configuration Centralized VRF Route-Leaking - Specific Prefixes Between Custom VRF	126
Centralized VRF Route-Leaking Brief - Shared Internet with Custom VRF	127
Configuring Centralized VRF Route-Leaking - Shared Internet with Custom VRF	128
Configuring Internet VRF on Border Node	128
Configuring Shared Internet BGP Instance on the Border Node	128

Configuring Custom VRF on Border Node	129
Configuring Custom VRF Context on the Border Node - 1	129
Configuring Custom VRF Instance in BGP on the Border Node	130
Example - Configuration Centralized VRF Route-Leaking - Shared Internet with Custom VRF	131
Centralized VRF Route-Leaking Brief - Shared Internet with VRF Default	133
Configuring Centralized VRF Route-Leaking - Shared Internet with VRF Default	134
Configuring VRF Default on Border Node	134
Configuring BGP Instance for VRF Default on the Border Node	134
Configuring Custom VRF on Border Node	134
Configuring Filter for Permitted Prefixes from VRF Default on the Border Node	135
Configuring Custom VRF Context on the Border Node - 2	135
Configuring Custom VRF Instance in BGP on the Border Node	136
Example - Configuration Centralized VRF Route-Leaking - VRF Default with Custom VRF	137

CHAPTER 7
Configuring VXLAN OAM 139

VXLAN OAM Overview	139
Loopback (Ping) Message	140
Traceroute or Pathtrace Message	141
Guidelines and Limitations for VXLAN NGOAM	142
Configuring VXLAN OAM	142
Configuring NGOAM Profile	145

CHAPTER 8
Configuring vPC Multi-Homing 147

Advertising Primary IP Address	147
BorderPE Switches in a vPC Setup	148
DHCP Configuration in a vPC Setup	148
IP Prefix Advertisement in vPC Setup	148

CHAPTER 9
Configuring Multi-Site 149

About VXLAN EVPN Multi-Site	149
Guidelines and Limitations for VXLAN EVPN Multi-Site	150
Enabling VXLAN EVPN Multi-Site	152
Multi-Site with vPC Support	153
About Multi-Site with vPC Support	153

Guidelines and Limitations for Multi-Site with vPC Support	153
Configuring Multi-Site with vPC Support	154
Configuring Peer Link as Transport in Case of Link Failure	157
Verifying the Multi-Site with vPC Support Configuration	159
Configuring VNI Dual Mode	160
Configuring Fabric/DCI Link Tracking	161
Configuring Fabric External Neighbors	162

CHAPTER 10

Configuring Tenant Routed Multicast 165

About Tenant Routed Multicast	165
About Tenant Routed Multicast Mixed Mode	167
Guidelines and Limitations for Tenant Routed Multicast	167
Guidelines and Limitations for Layer 3 Tenant Routed Multicast	168
Guidelines and Limitations for Layer 2/Layer 3 Tenant Routed Multicast (Mixed Mode)	168
Rendezvous Point for Tenant Routed Multicast	169
Configuring a Rendezvous Point for Tenant Routed Multicast	169
Configuring a Rendezvous Point Inside the VXLAN Fabric	170
Configuring an External Rendezvous Point	171
Configuring Layer 3 Tenant Routed Multicast	173
Configuring TRM on the VXLAN EVPN Spine	177
Configuring Tenant Routed Multicast in Layer 2/Layer 3 Mixed Mode	179
Configuring Layer 2 Tenant Routed Multicast	184
Configuring TRM with vPC Support	184

CHAPTER 11

Configuring Cross Connect 189

About VXLAN Cross Connect	189
Guidelines and Limitations for VXLAN Cross Connect	190
Configuring VXLAN Cross Connect	191
Verifying VXLAN Cross Connect Configuration	193
Configuring NGOAM for VXLAN Cross Connect	194
Verifying NGOAM for VXLAN Cross Connect	194
NGOAM Authentication	195
Guidelines and Limitations for Q-in-VNI	196
Configuring Q-in-VNI	198

Configuring Selective Q-in-VNI	199
Configuring Q-in-VNI with LACP Tunneling	201
Selective Q-in-VNI with Multiple Provider VLANs	204
About Selective Q-in-VNI with Multiple Provider VLANs	204
Guidelines and Limitations for Selective Q-in-VNI with Multiple Provider VLANs	204
Configuring Selective Q-in-VNI with Multiple Provider VLANs	205
Configuring QinQ-QinVNI	207
Overview for QinQ-QinVNI	207
Guidelines and Limitations for QinQ-QinVNI	207
Configuring QinQ-QinVNI	208
Removing a VNI	209

CHAPTER 12

Configuring Port VLAN Mapping	211
About Translating Incoming VLANs	211
Guidelines and Limitations for Port VLAN Mapping	212
Configuring Port VLAN Mapping on a Trunk Port	214
Configuring Inner VLAN and Outer VLAN Mapping on a Trunk Port	216

CHAPTER 13

Configuring IGMP Snooping	219
Configuring IGMP Snooping Over VXLAN	219
Overview of IGMP Snooping Over VXLAN	219
Guidelines and Limitations for IGMP Snooping Over VXLAN	219
Configuring IGMP Snooping Over VXLAN	219

CHAPTER 14

Configuring VLANs	221
About Private VLANs over VXLAN	221
Guidelines and Limitations for Private VLANs over VXLAN	222
Configuration Example for Private VLANs	222

CHAPTER 15

Configuring Policy-Based Redirect	225
Service Redirection in VXLAN EVPN Fabrics	225
Guidelines and Limitations for Policy-Based Redirect	225
Enabling the Policy-Based Redirect Feature	226
Configuring a Route Policy	226

Verifying the Policy-Based Redirect Configuration	228
Configuration Example for Policy-Based Redirect	228

CHAPTER 16

Configuring ACL 231

About Access Control Lists	231
Guidelines and Limitations for VXLAN ACLs	233
VXLAN Tunnel Encapsulation Switch	234
Port ACL on the Access Port on Ingress	234
VLAN ACL on the Server VLAN	235
Routed ACL on an SVI on Ingress	236
Routed ACL on the Uplink on Egress	238
VXLAN Tunnel Decapsulation Switch	238
Routed ACL on the Uplink on Ingress	238
Port ACL on the Access Port on Egress	238
VLAN ACL for the Layer 2 VNI Traffic	238
VLAN ACL for the Layer 3 VNI Traffic	240
Routed ACL on an SVI on Egress	241

CHAPTER 17

Configuring VXLAN QoS 243

Information About VXLAN QoS	243
VXLAN QoS Terminology	243
VXLAN QoS Features	245
Trust Boundaries	245
Classification	245
Marking	245
Policing	245
Queuing and Scheduling	246
Traffic Shaping	246
Network QoS	246
VXLAN Priority Tunneling	247
MQC CLI	247
VXLAN QoS Topology and Roles	247
Ingress VTEP and Encapsulation in the VXLAN Tunnel	247
Transport Through the VXLAN Tunnel	248

Egress VTEP and Decapsulation of the VXLAN Tunnel	248
Classification at the Ingress VTEP, Spine, and Egress VTEP	249
IP to VXLAN	249
Inside the VXLAN Tunnel	249
VXLAN to IP	250
Decapsulated Packet Priority Selection	250
Guidelines and Limitations for VXLAN QoS	251
Default Settings for VXLAN QoS	252
Configuring VXLAN QoS	253
Configuring Type QoS on the Egress VTEP	253
Verifying the VXLAN QoS Configuration	255
VXLAN QoS Configuration Examples	255

CHAPTER 18

Configuring vPC Fabric Peering	257
Information About vPC Fabric Peering	257
Guidelines and Limitations for vPC Fabric Peering	258
Configuring vPC Fabric Peering	259
Migrating from vPC to vPC Fabric Peering	262
Verifying vPC Fabric Peering Configuration	264

APPENDIX A

Configuring Bud Node	267
VXLAN Bud Node Over vPC Overview	267
VXLAN Bud Node Over vPC Topology Example	268

APPENDIX B

DHCP Relay in VXLAN BGP EVPN	273
DHCP Relay in VXLAN BGP EVPN Overview	273
DHCP Relay in VXLAN BGP EVPN Example	274
DHCP Relay on VTEPs	275
Client on Tenant VRF and Server on Layer 3 Default VRF	275
Client on Tenant VRF (SVI X) and Server on the Same Tenant VRF (SVI Y)	278
Client on Tenant VRF (VRF X) and Server on Different Tenant VRF (VRF Y)	282
Client on Tenant VRF and Server on Non-Default Non-VXLAN VRF	285
Configuring vPC Peers Example	287
vPC VTEP DHCP Relay Configuration Example	289

APPENDIX C**Configuring Layer 4 - Layer 7 Network Services Integration 291**

About VXLAN Layer 4 - Layer 7 Services	291
Integrating Layer 3 Firewalls in VXLAN Fabrics	291
Single-Attached Firewall with Static Routing	292
Recursive Static Routes Distributed to the Rest of the Fabric	294
Redistribute Static Routes into BGP and Advertise to the Rest of the Fabric	294
Dual-Attached Firewall with Static Routing	294
Single-Attached Firewall with eBGP Routing	295
Dual-Attached Firewall with eBGP Routing	298
Per-VRF Peering via vPC Peer-Link	301
Single-Attached Firewall with OSPF	301
Redistribute OSPF Routes into BGP and Advertise to the Rest of the Fabric	302
Dual-Attached Firewall with OSPF	303
Redistribute OSPF Routes into BGP and Advertise to the Rest of the Fabric	305
Firewall as Default Gateway	305
Transparent Firewall Insertion	306
Overview of EVPN with Transparent Firewall Insertion	306
EVPN with Transparent Firewall Insertion Example	308
Show Command Examples	311

APPENDIX D**Configuring Multi-Homing 313**

VXLAN EVPN Multi-Homing Overview	313
Introduction to Multi-Homing	313
BGP EVPN Multi-Homing	313
BGP EVPN Multi-Homing Terminology	314
EVPN Multi-Homing Implementation	314
EVPN Multi-Homing Redundancy Group	315
Ethernet Segment Identifier	315
LACP Bundling	316
Guidelines and Limitations for VXLAN EVPN Multi-Homing	316
Configuring VXLAN EVPN Multi-Homing	317
Enabling EVPN Multi-Homing	317
VXLAN EVPN Multi-Homing Configuration Examples	318

Configuring Layer 2 Gateway STP	319
Layer 2 Gateway STP Overview	319
Guidelines for Moving to Layer 2 Gateway STP	320
Enabling Layer 2 Gateway STP on a Switch	321
Configuring VXLAN EVPN Multi-Homing Traffic Flows	324
EVPN Multi-Homing Local Traffic Flows	324
EVPN Multi-Homing Remote Traffic Flows	328
EVPN Multi-Homing BUM Flows	332
Configuring ESI ARP Suppression	335
Overview of ESI ARP Suppression	335
Limitations for ESI ARP Suppression	336
Configuring ESI ARP Suppression	336
Displaying Show Commands for ESI ARP Suppression	336
Configuring VLAN Consistency Checking	338
Overview of VLAN Consistency Checking	338
VLAN Consistency Checking Guidelines and Limitations	339
Configuring VLAN Consistency Checking	339
Displaying Show Command Output for VLAN Consistency Checking	339

APPENDIX E

Configuring Proportional Multipath for VNF	341
About Proportional Multipath for VNF	341
Guidelines and Limitations for Proportional Multipath for VNF	345
Configuring the Route Reflector	346
Configuring the ToR	347
Configuring the Border Leaf	350
Configuring the BGP Legacy Peer	354
Configuring a User-Defined Profile for Maintenance Mode	355
Configuring a User-Defined Profile for Normal Mode	356
Configuring a Default Route Map	356
Applying a Route Map to a Route Reflector	356
Verifying Proportional Multipath for VNF	357



Preface

This preface includes the following sections:

- [Audience, on page xiii](#)
- [Document Conventions, on page xiii](#)
- [Related Documentation for Cisco Nexus 9000 Series Switches, on page xiv](#)
- [Documentation Feedback, on page xiv](#)
- [Communications, Services, and Additional Information, on page xiv](#)

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 contains the following sections:

- [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
	Chapter reorganization	9.2(3)	
	Changed the document title from 9.x to 9.2(x)		Title page
MultiAuth with CoA	Change of authorization.	9.2(3)	Guidelines and Limitations for VXLAN, on page 9
NGOAM	Support added for Cisco Nexus 9504 and 9508 switches with -R line cards.	9.2(3)	Guidelines and Limitations for VXLAN NGOAM, on page 142
PV Routing	Support added for Port VLAN on Cisco Nexus 9300-FX and 9300-FX2 platform switches.	9.2(3)	Configuring Port VLAN Mapping on a Trunk Port, on page 214
Selective Q-in-VNI with Multiple Provider VLANs	Selective Q-in-VNI with multiple provider VLANs is a VXLAN tunneling feature.	9.2(3)	About Selective Q-in-VNI with Multiple Provider VLANs, on page 204
VXLAN QoS	Enables VXLAN encapsulated traffic to use QoS.	9.2(3)	Configuring VXLAN QoS, on page 243

Feature	Description	Changed in Release	Where Documented
vPC Fabric Peering	Added support for Cisco Nexus 9332C, 9364C, and 9300-FX/FXP/FX2 platform switches.	9.2(3)	Configuring vPC Fabric Peering, on page 259
TRM VXLAN BGP EVPN	Added support for the Cisco Nexus 9332C platform switches.	9.2(2)	Guidelines and Limitations for Layer 3 Tenant Routed Multicast, on page 168
VXLAN DHCP v4 Relay	Support added for Cisco Nexus 9504 and 9508 with -R line cards.	9.2(2)	DHCP Relay in VXLAN BGP EVPN, on page 273
CLI Simplification	Support added for the reduction of CLI commands.	9.2(1)	Example of VXLAN BGP EVPN (EBGP), on page 94 Example of VXLAN BGP EVPN (IBGP), on page 83
Multi-Site with vPC Support	Support added for Multi-Site with vPC.	9.2(1)	About Multi-Site with vPC Support, on page 153
PIM BiDir	Support added for PIM BiDir underlay with and without vPC support for Cisco Nexus 9000-EX, 9000-FX, and 9000-FX2 platform switches.	9.2(1)	Guidelines and Limitations for VXLAN, on page 9
Proportional Multipath for VNF	Enables advertising of all the available next hops to a given network destination.	9.2(1)	Configuring Proportional Multipath for VNF, on page 341
PVLANS over VXLAN	Support added for PVLANS to be configured over VXLAN.	9.2(1)	Configuring Port VLAN Mapping, on page 211
TRM Border Leaf with vPC Support	Support added for forwarding multicast between sender and receivers.	9.2(1)	Configuring Layer 3 Tenant Routed Multicast, on page 173
VXLAN Cross Connect	Provides point-to-point tunneling of data and control packets from one VTEP to another.	9.2(1)	About VXLAN Cross Connect, on page 189
VXLAN with vPC	Support added for Cisco Nexus 9504 and 9508 with -R line cards.	9.2(1)	Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide, Release 9.2(x)



CHAPTER 2

Overview

This chapter contains the following sections:

- [Licensing Requirements, on page 3](#)
- [Supported Platforms, on page 3](#)
- [VXLAN Overview, on page 3](#)
- [Cisco Nexus 9000 as Hardware-Based VXLAN Gateway, on page 4](#)
- [VXLAN Encapsulation and Packet Format, on page 4](#)
- [VXLAN Tunnel, on page 5](#)
- [VXLAN Tunnel Endpoint, on page 5](#)
- [Underlay Network, on page 5](#)
- [Overlay Network, on page 5](#)
- [Distributed Anycast Gateway, on page 5](#)
- [Control Plane, on page 6](#)

Licensing Requirements

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

Supported Platforms

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

VXLAN Overview

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

VXLAN has the following benefits:

- Flexible placement of workloads across the data center fabric.

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

- Higher scalability to allow more Layer 2 segments.

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

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

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

Cisco Nexus 9000 as Hardware-Based VXLAN Gateway

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

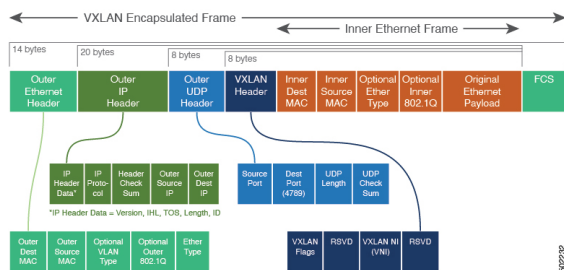
VXLAN Encapsulation and Packet Format

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

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

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

Figure 1:



VXLAN Tunnel

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

VXLAN Tunnel Endpoint

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

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

Underlay Network

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

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

Overlay Network

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

Distributed Anycast Gateway

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

Control Plane

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

Flood and Learn Multicast-Based Learning Control Plane

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

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

VXLAN MPBGP EVPN Control Plane

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

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

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

- IP reachability information for the tunnel endpoints associated with a segment and the hosts behind a specific tunnel endpoint.
 - Distribution of host MAC reachability to reduce/eliminate unknown unicast flooding.
 - Distribution of host IP/MAC bindings to provide local ARP suppression.
 - Host mobility.
 - A single address family (MPBGPEVPN) to distribute both L2 and L3 route reachability information.
- Segmentation of Layer 2 and Layer 3 traffic
 - Traffic segmentation is achieved with using VXLAN encapsulation, where VNI acts as segment identifier.



CHAPTER 3

Configuring VXLAN

This chapter contains the following sections:

- [Guidelines and Limitations for VXLAN, on page 9](#)
- [Considerations for VXLAN Deployment, on page 15](#)
- [vPC Considerations for VXLAN Deployment, on page 18](#)
- [Network Considerations for VXLAN Deployments, on page 22](#)
- [Considerations for the Transport Network, on page 23](#)
- [Considerations for Tunneling VXLAN, on page 24](#)
- [Configuring VXLAN, on page 25](#)

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)I6(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.
- For scale environments, the VLAN IDs related to the VRF and Layer-3 VNI (L3VNI) must be reserved with the **system vlan nve-overlay id** command.
- NLB in the unicast, multicast, and IGMP multicast modes is not supported on Cisco Nexus 9000 Series based VXLAN VTEPs. The work around is to move the NLB cluster behind intermediary device (which supports NLB in the respective mode) and inject the cluster IP address as external prefix into VXLAN fabric.
- Beginning with Cisco NX-OS Release 9.2(3), support added for MultiAuth Change of Authorization (CoA). For more information, see the [Cisco Nexus 9000 Series NX-OS Security Configuration Guide, Release 9.2\(x\)](#).
- The **lACP vpc-convergence** command can be configured in VXLAN and non-VXLAN environments that have vPC port channels to hosts that support LACP.
- Beginning with Cisco NX-OS Release 9.2(1), PIM BiDir for VXLAN underlay with and without vPC is supported.

The following is a list of what is not supported when the PIM BiDir for VXLAN underlay feature is configured:

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

For redundant RPs, use Phantom RP.

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

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

Wait for all tables to clean up.

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

- When entering the **no feature pim** command, NVE ownership on the route is not removed so the route stays and traffic continues to flow. Aging is done by PIM. PIM does not age out entries having a VXLAN encaps flag.
- 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.

- VXLAN is supported on the Cisco Nexus 9348GC-FXP switch.
 - When SVI is enabled on a VTEP (flood and learn, or EVPN) regardless of ARP suppression, make sure that ARP-ETHER TCAM is carved using the **hardware access-list tcam region arp-ether 256 double-wide** command. This is not applicable to the Cisco Nexus 9200 and 9300-EX platform switches and Cisco Nexus 9500 platform switches with 9700-EX line cards.
 - For information regarding 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 9.2\(x\)](#).
 - For the Cisco Nexus 9504 and 9508 switches with -R line cards, VXLAN Layer 2 Gateway is supported on the 9636C-RX line card. VXLAN and MPLS cannot be enabled on the Cisco Nexus 9508 switch at the same time.
 - For the Cisco Nexus 9504 and 9508 switches with -R line cards, if VXLAN is enabled, the Layer 2 Gateway cannot be enabled when there is any line card other than the 9636C-RX.
 - For the Cisco Nexus 9504 and 9508 switches with -R line cards, PIM/ASM is supported in the underlay ports. PIM/Bidir is not supported. For more information, see the [Cisco Nexus 9000 Series NX-OS Multicast Routing Configuration Guide, Release 9.2\(x\)](#).
 - For the Cisco Nexus 9504 and 9508 switches with -R line cards, IPv6 hosts routing in the overlay is supported.
 - For the Cisco Nexus 9504 and 9508 switches with -R line cards, ARP suppression is supported.
- load-share**
- 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 9.2\(x\)](#).

- 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 bpdupfilter enable
  lacp vpc-convergence
  vpc 10
```

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

- Port-VLAN with VXLAN is supported on Cisco Nexus 9300-EX and 9500 Series switches with 9700-EX line cards with the following exceptions:
 - Only Layer 2 (no routing) is supported with port-VLAN with VXLAN on these switches.
 - No inner VLAN mapping is supported.
- The **system nve ipmc** CLI command is not applicable to the Cisco 9200 and 9300-EX platform switches and Cisco 9500 platform switches with 9700-EX line cards.
- Bind NVE to a loopback address that is separate from other loopback addresses that are required by Layer 3 protocols. A best practice is to use a dedicated loopback address for VXLAN. This best practice should be applied not only for the vPC VXLAN deployment, but for all VXLAN deployments.
- To remove configurations from an NVE interface, we recommend manually removing each configuration rather than using the **default interface nve** command.
- 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.
- VXLAN is supported for the Cisco Nexus 93108TC-EX and 93180YC-EX switches and for Cisco Nexus 9500 Series switches with the X9732C-EX line card.
- DHCP snooping (Dynamic Host Configuration Protocol snooping) is not supported on VXLAN VLANs.
- RACLs are not supported on Layer 3 uplinks for VXLAN traffic. Egress VACLs support is not available for de-capsulated packets in the network to access direction on the inner payload.
As a best practice, use PACLS/VACLs for the access to the network direction.
- The QoS buffer-boost feature is not applicable for VXLAN traffic.
- 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.
- 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.
- VXLAN is supported on Cisco Nexus 9500 platform switches with the following line cards:
 - 9500-R
 - 9564PX
 - 9564TX
 - 9536PQ
 - 9700-EX
 - 9700-FX
- Cisco Nexus 9300 Series switches with 100G uplinks only support VXLAN switching/bridging. 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.

- MDP is not supported for VXLAN configurations.
- Consistency checkers are not supported for VXLAN tables.
- ARP suppression is supported for a VNI only if the VTEP hosts the First-Hop Gateway (Distributed Anycast Gateway) for this VNI. The VTEP and SVI for this VLAN must be properly configured for the Distributed Anycast Gateway operation (for example, global anycast gateway MAC address configured and anycast gateway with the virtual IP address on the SVI).
- ARP suppression is a per-L2VNI fabric-wide setting in the VXLAN fabric. Enable or disable this feature consistently across all VTEPs in the fabric. Inconsistent ARP suppression configuration across VTEPs is not supported.
- The VXLAN network identifier (VNID) 16777215 is reserved and should not be configured explicitly.
- VXLAN supports In Service Software Upgrade (ISSU).

- VXLAN does not support coexistence with the GRE tunnel feature or the MPLS (static or segment-routing) feature.
- VTEP connected to FEX host interface ports is not supported.
- If multiple VTEPs use the same multicast group address for underlay multicast but have different VNIs, the VTEPs should have at least one VNI in common. Doing so ensures that NVE peer discovery occurs and underlay multicast traffic is forwarded correctly. For example, leafs L1 and L4 could have VNI 10 and leafs L2 and L3 could have VNI 20, and both VNIs could share the same group address. When leaf L1 sends traffic to leaf L4, the traffic could pass through leaf L2 or L3. Because NVE peer L1 is not learned on leaf L2 or L3, the traffic is dropped. Therefore, VTEPs that share a group address need to have at least one VNI in common so that peer learning occurs and traffic is not dropped. This requirement applies to VXLAN bud-node topologies.
- VXLAN does not support co-existence with MVR and MPLS for Cisco Nexus 9504 and 9508 with -R line cards.
- Resilient hashing (port-channel load-balancing resiliency) and VXLAN configurations are not compatible with VTEPs using ALE uplink ports.



Note Resilient hashing is disabled by default.

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

Considerations for VXLAN Deployment

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

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

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

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

```
system vlan nve-overlay id 2000

    vlan 2000
        vn-segment 50000

    interface Vlan2000
```

```

vrf member MYVRF_50000
ip forward
ipv6 forward

vrf context MYVRF_50000
vni 50000

```



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

- When configuring VXLAN BGP EVPN, the "System Routing Mode: Default" is applicable for the following hardware platforms:
 - Cisco Nexus 9200 platform switches
 - Cisco Nexus 9300 platform switches
 - Cisco Nexus 9300-EX platform switches
 - Cisco Nexus 9300-FX/FX2 platform switches
 - Cisco Nexus 9500 platform switches with X9500 line cards
 - Cisco Nexus 9500 platform switches with X9700-EX/FX line cards
- The "System Routing Mode: template-vxlan-scale" is not applicable.
- When using VXLAN BGP EVPN in combination with Cisco NX-OS Release 7.0(3)I4(x) or NX-OS Release 7.0(3)I5(1), the "System Routing Mode: template-vxlan-scale" is required on the following hardware platforms:
 - Cisco Nexus 9300-EX Switches
 - Cisco Nexus 9500 Switches with X9700-EX line cards
- Changing the "System Routing Mode" requires a reload of the switch.
- A loopback address is required when using the **source-interface config** command. The loopback address represents the local VTEP IP.
- During boot-up of a switch, you can use the **source-interface hold-down-time** *hold-down-time* command to suppress advertisement of the NVE loopback address until the overlay has converged. The range for the *hold-down-time* is 0 - 2147483647 seconds. The default is 300 seconds.



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

- To establish IP multicast routing in the core, IP multicast configuration, PIM configuration, and RP configuration is required.
- VTEP to VTEP unicast reachability can be configured through any IGP protocol.

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

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

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



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

For example:

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



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

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



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

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

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

- When changing the IP address of a VTEP device, you must shut the NVE interface before changing the IP address.
- As a best practice, when migrating any sets of VTEP to a multisite BGW, NVE interface must be shut on all the VTEPs where this migration is being performed. NVE interface should be brought back up once the migration is complete and all necessary configurations for multisite are applied to the VTEPs.
- As a best practice, the RP for the multicast group should be configured only on the spine layer. Use the anycast RP for RP load balancing and redundancy.

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

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



Note

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

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

vPC Considerations for VXLAN Deployment

- As a best practice when feature vPC is 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, we recommend to increase the **delay restore interface-vlan** timer to 45 seconds.
- If a ping is initiated to the attached hosts on VXLAN VLAN from a vPC VTEP node, the source IP address used by default is the anycast IP that is configured on the SVI. This ping can fail to get a response from the host in case the response is hashed to the vPC peer node. This issue can happen when a ping is initiated from a VXLAN vPC node to the attached hosts without using a unique source IP address. As a workaround for this situation, use VXLAN OAM or create a unique loopback on each vPC VTEP and route the unique address via a backdoor path.
- The loopback address used by NVE needs to be configured to have a primary IP address and a secondary IP address.

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

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

- Consistent VNI to group mapping.
- For multicast, the vPC node that receives the (S, G) join from the RP (rendezvous point) becomes the DF (designated forwarder). On the DF node, encap routes are installed for multicast.
Decap routes are installed based on the election of a decapper from between the vPC primary node and the vPC secondary node. The winner of the decap election is the node with the least cost to the RP. However, if the cost to the RP is the same for both nodes, the vPC primary node is elected.
The winner of the decap election has the decap mroute installed. The other node does not have a decap route installed.
- On a vPC device, BUM traffic (broadcast, unknown-unicast, and multicast traffic) from hosts is replicated on the peer-link. A copy is made of every native packet and each native packet is sent across the peer-link to service orphan-ports connected to the peer vPC switch.
To prevent traffic loops in VXLAN networks, native packets ingressing the peer-link cannot be sent to an uplink. However, if the peer switch is the encapper, the copied packet traverses the peer-link and is sent to the uplink.



Note Each copied packet is sent on a special internal VLAN (VLAN 4041 or VLAN 4046).

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



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

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

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

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

- The vPC peer-gateway feature must be enabled on both peers to facilitate NVE RMAC/VMAC programming on both peers. For peer-gateway functionality, at least one backup routing SVI is required to be enabled across peer-link and also configured with PIM. This provides a backup routing path in the case when VTEP loses complete connectivity to the spine. Remote peer reachability is re-routed over

peer-link in his case. In BUD node topologies, the backup SVI needs to be added as a static OIF for each underlay multicast group.

```
switch# sh ru int vlan 2

interface Vlan2
  description 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.

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

- When the NVE or loopback is shut in vPC configurations:
 - If the NVE or loopback is shut only on the primary vPC switch, the global VXLAN vPC consistency checker fails. Then the NVE, loopback, and vPCs are taken down on the secondary vPC switch.
 - If the NVE or loopback is shut only on the secondary vPC switch, the global VXLAN vPC consistency checker fails. Then, the NVE, loopback, and secondary vPC are brought down on the secondary. Traffic continues to flow through the primary vPC switch.
 - As a best practice, you should keep both the NVE and loopback up on both the primary and secondary vPC switches.
- Redundant anycast RPs configured in the network for multicast load-balancing and RP redundancy are supported on vPC VTEP topologies.
- As a best practice, when changing the secondary IP address of an anycast vPC VTEP, the NVE interfaces on both the vPC primary and the vPC secondary must be shut before the IP changes are made.
- When SVI is enabled on a VTEP (flood and learn, or EVPN) regardless of ARP suppression, make sure that ARP-ETHER TCAM is carved using the **hardware access-list tcam region arp-ether 256 double-wide** command. This requirement does not apply to Cisco Nexus 9200, 9300-EX, and 9300-FX/FX2 platform switches and Cisco Nexus 9500 platform switches with 9700-EX line cards.
- The **show** commands with the **internal** keyword are not supported.
- DHCP snooping (Dynamic Host Configuration Protocol snooping) is not supported on VXLAN VLANs.
- RACLs are not supported on Layer 3 uplinks for VXLAN traffic. Egress VACLs support is not available for de-capsulated packets in the network to access direction on the inner payload.

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

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

- QoS classification is not supported for VXLAN traffic in the network to access direction on the Layer 3 uplink interface.
See the [Cisco Nexus 9000 Series NX-OS Quality of Service Configuration Guide, Release 9.3\(x\)](#) for other guidelines and limitations for the VXLAN QoS feature.
- The QoS buffer-boost feature is not applicable for VXLAN traffic.
- VTEPs do not support VXLAN-encapsulated traffic over parent interfaces if subinterfaces are configured, regardless of VRF participation.
- VTEPs do not support VXLAN encapsulated traffic over subinterfaces. This is regardless of VRF participation or IEEE802.1Q encapsulation.
- Mixing subinterfaces for VXLAN and non-VXLAN VLANs is not supported.
- Point-to-multipoint Layer 3 and SVI uplinks are not supported.
- Using the **ip forward** command enables the VTEP to forward the VXLAN de-capsulated packet destined to its router IP to the SUP/CPU.
- Before configuring it as an SVI, the backup VLAN needs to be configured on Cisco Nexus 9200, 9300-EX, and 9300-FX/FX2 platform switches as an infra-VLAN with the **system nve infra-vlans** command.
- VXLAN is supported on Cisco Nexus 9500 platform switches with the following line cards:
 - 9700-EX
 - 9700-FX
- When Cisco Nexus 9500 platform switches are used as VTEPs, 100G line cards are not supported on Cisco Nexus 9500 platform switches. This limitation does not apply to a Cisco Nexus 9500 switch with 9700-EX or -FX line cards.
- Cisco Nexus 9300 platform switches with 100G uplinks only support VXLAN switching/bridging. Cisco Nexus 9200 and Cisco Nexus 9300-EX/FX platform switches do not have this restriction.



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

- The VXLAN UDP port number is used for VXLAN encapsulation. For Cisco Nexus NX-OS, the UDP port number is 4789. It complies with IETF standards and is not configurable.
- For Cisco Nexus 9200 platform switches that have the Application Spine Engine (ASE2). There exists a Layer 3 VXLAN (SVI) throughput issue. There is a data loss for packets of sizes 99 - 122.
- The VXLAN network identifier (VNID) 16777215 is reserved and should not be configured explicitly.
- VXLAN supports In Service Software Upgrade (ISSU).
- VXLAN does not support coexistence with the GRE tunnel feature or the MPLS (static or segment routing) feature.
- VTEP connected to FEX host interface ports is not supported.
- Resilient hashing (port-channel load-balancing resiliency) and VXLAN configurations are not compatible with VTEPs using ALE uplink ports.



Note Resilient hashing is disabled by default.

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



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

Network Considerations for VXLAN Deployments

- MTU Size in the Transport Network

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

- ECMP and LACP Hashing Algorithms in the Transport Network

As described in a previous section, Cisco Nexus 9000 Series Switches introduce a level of entropy in the source UDP port for ECMP and LACP hashing in the transport network. As a way to augment this implementation, the transport network uses an ECMP or LACP hashing algorithm that takes the UDP source port as an input for hashing, which achieves the best load-sharing results for VXLAN encapsulated traffic.

- Multicast Group Scaling

The VXLAN implementation on Cisco Nexus 9000 Series Switches uses multicast tunnels for broadcast, unknown unicast, and multicast traffic forwarding. Ideally, one VXLAN segment mapping to one IP multicast group is the way to provide the optimal multicast forwarding. It is possible, however, to have multiple VXLAN segments share a single IP multicast group in the core network. VXLAN can support up to 16 million logical Layer 2 segments, using the 24-bit VNID field in the header. With one-to-one mapping between VXLAN segments and IP multicast groups, an increase in the number of VXLAN segments causes a parallel increase in the required multicast address space and the amount of forwarding states on the core network devices. At some point, multicast scalability in the transport network can become a concern. In this case, mapping multiple VXLAN segments to a single multicast group can help conserve multicast control plane resources on the core devices and achieve the desired VXLAN scalability. However, this mapping comes at the cost of suboptimal multicast forwarding. Packets forwarded to the multicast group for one tenant are now sent to the VTEPs of other tenants that are sharing the same multicast group. This causes inefficient utilization of multicast data plane resources. Therefore, this solution is a trade-off between control plane scalability and data plane efficiency.

Despite the suboptimal multicast replication and forwarding, having multiple-tenant VXLAN networks to share a multicast group does not bring any implications to the Layer 2 isolation between the tenant networks. After receiving an encapsulated packet from the multicast group, a VTEP checks and validates

the VNID in the VXLAN header of the packet. The VTEP discards the packet if the VNID is unknown to it. Only when the VNID matches one of the VTEP's local VXLAN VNIDs, does it forward the packet to that VXLAN segment. Other tenant networks will not receive the packet. Thus, the segregation between VXLAN segments is not compromised.

Considerations for the Transport Network

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

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

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

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

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

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

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

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

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

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



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



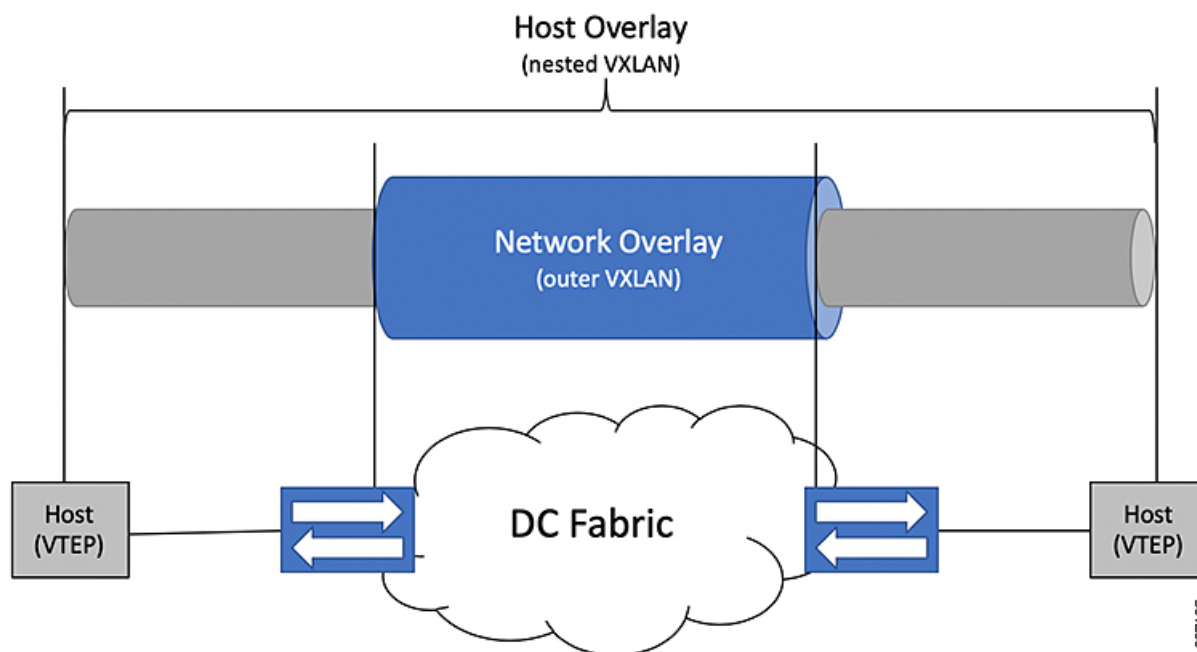
Note * Not required for static ingress replication or BGP EVPN ingress replication.

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, 9300-FX2, 9500-EX, 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.

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 a VXLAN VTEP in vPC

You can configure a VXLAN VTEP in a vPC.

Procedure

-
- | | |
|---------------|---|
| Step 1 | Enter global configuration mode.

switch# configure terminal |
| Step 2 | Enable the vPC feature on the device.

switch(config)# feature vpc |
| Step 3 | Enable the interface VLAN feature on the device.

switch(config)# feature interface-vlan |
| Step 4 | Enable the LACP feature on the device. |

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

Step 5 Enable the PIM feature on the device.

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

Step 6 Enables the OSPF feature on the device.

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

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

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

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

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

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

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

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

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

Step 11 Create primary and secondary IP addresses.

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

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

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

Step 13 Create a vPC domain.

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

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

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

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

The management ports and VRF are the defaults.

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

Step 15 Enable Peer-Gateway on the vPC domain.

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

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

Step 16 Enable Peer-switch on the vPC domain.

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

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

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

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

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

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

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

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

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

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

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

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

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

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


Configuring Static MAC for VXLAN VTEP

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



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

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.
Step 5	(Optional) show mac address-table static interface nve <i>x</i>	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

	Command or Action	Purpose
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 <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	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.

	Command or Action	Purpose
		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.



CHAPTER 4

Configuring the Underlay

This chapter contains the following sections:

- [IP Fabric Underlay, on page 33](#)

IP Fabric Underlay

Underlay Considerations

Unicast Underlay:

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

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

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

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

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



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

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

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

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

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

Figure 3: Overlay Overview

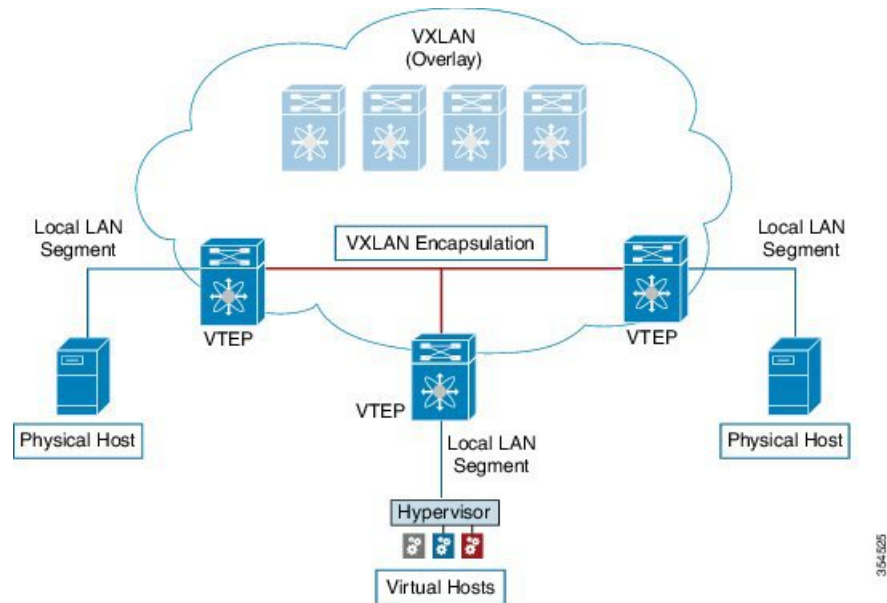
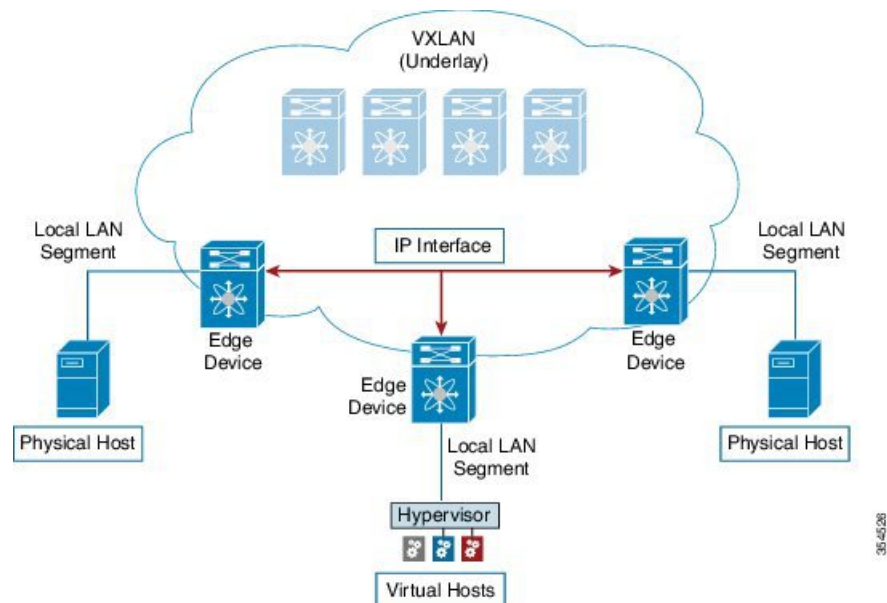


Figure 4: Underlay Overview



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

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

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



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

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

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



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

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

Unicast routing and IP addressing options

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

**Important**

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

OSPF Underlay IP Network

Some considerations are given below:

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

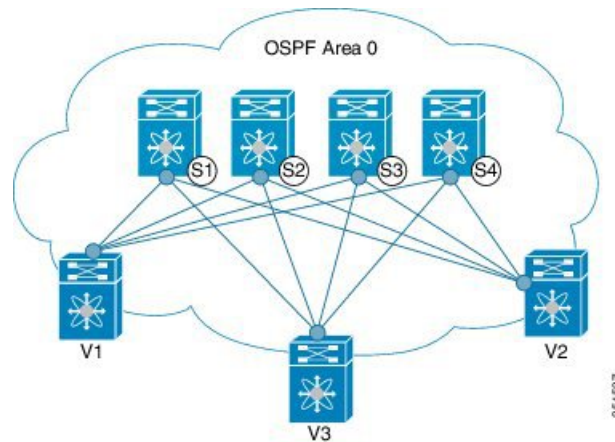
**Note**

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

**Note**

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

OSPF configuration sample – P2P and IP unnumbered network scenarios

Figure 5: OSPF as the underlay routing protocol**OSPF – P2P link scenario with /31 mask**

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

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

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

OSPF global configuration on leaf switch V1

(config) #

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

OSPF leaf switch V1 P2P interface configuration

(config) #

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

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

The OSPF instance is tagged as UNDERLAY for better recall.

OSPF loopback interface configuration (leaf switch V1)

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

(config) #

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

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

OSPF global configuration on spine switch S1

(config) #

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

(Corresponding) OSPF spine switch S1 P2P interface configuration

(config) #

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



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

OSPF loopback Interface Configuration (spine switch S1)

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

(config) #

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

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

.

.

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

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

OSPF - IP unnumbered scenario

A sample OSPF IP unnumbered configuration is given below:

OSPF leaf switch V1 configuration

OSPF global configuration on leaf switch V1

(config) #

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

The OSPF instance is tagged as UNDERLAY for better recall.

OSPF leaf switch V1 P2P interface configuration

(config) #

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

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

OSPF loopback interface configuration

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

(config) #

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

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

OSPF spine switch S1 configuration:

OSPF global configuration on spine switch S1

(config) #

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

(Corresponding) OSPF spine switch S1 P2P interface configuration

(config) #

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

OSPF loopback interface configuration (spine switch S1)

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

(config) #

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

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

.

.

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

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

OSPF Verification

Use the following commands for verifying OSPF configuration:

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

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

Leaf-Switch-V1# show ip ospf interface

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

```

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

```

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

```

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

```

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

IS-IS Underlay IP Network

Some considerations are given below:

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



Important

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

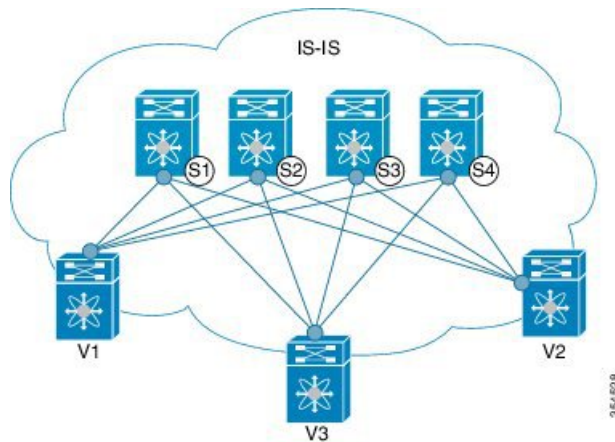


Note

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

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

Figure 6: IS-IS as the underlay routing protocol



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

IS-IS – P2P link scenario with /31 mask

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

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

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

IS-IS configuration on leaf switch V1

IS-IS global configuration

(config) #

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

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

IS-IS P2P interface configuration (leaf switch V1)

(config) #

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

```
ip router isis UNDERLAY
```

IS-IS loopback interface configuration (leaf switch V1)

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

(config) #

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

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

(Corresponding) IS-IS spine switch S1 configuration

IS-IS global configuration

(config) #

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

IS-IS P2P interface configuration (spine switch S1)

(config) #

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

IS-IS loopback interface configuration (spine switch S1)

(config) #

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

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

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

IS-IS - IP unnumbered scenario

IS-IS configuration on leaf switch V1

IS-IS global configuration

```
(config)#
```

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

IS-IS interface configuration (leaf switch V1)

```
(config) #
```

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

IS-IS loopback interface configuration (leaf switch V1)

```
(config)
```

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

IS-IS configuration on the spine switch S1

IS-IS global configuration

```
(config)#
```

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

IS-IS interface configuration (spine switch S1)

```
(config)#
```

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

IS-IS loopback interface configuration (spine switch S1)

```
(config)#
```

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


IS-IS Verification

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

```
Leaf-Switch-V1# show isis
```

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

```
Leaf-Switch-V1# show isis interface
```

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

```

Level Metric 1 1
2 1
Topologies enabled:
  L MT Metric MetricCfg Fwdng IPv4-MT IPv4Cfg IPv6-MT IPv6Cfg
  1 0 1 no UP UP yes DN no
  2 0 1 no DN DN no DN no

loopback1, Interface status: protocol-up/link-up/admin-up
IP address: 10.1.2.74, IP subnet: 10.1.2.74/32
IPv6 routing is disabled
Level1
  No auth type and keychain
  Auth check set
Level2
  No auth type and keychain
  Auth check set
Index: 0x0002, Local Circuit ID: 0x01, Circuit Type: L1
BFD IPv4 is locally disabled for Interface loopback1
BFD IPv6 is locally disabled for Interface loopback1
MTR is disabled
Passive level: level-2
Level Metric
1 1
2 1
Topologies enabled:
  L MT Metric MetricCfg Fwdng IPv4-MT IPv4Cfg IPv6-MT IPv6Cfg
  1 0 1 no UP UP yes DN no
  2 0 1 no DN DN no DN no

Ethernet1/41, Interface status: protocol-up/link-up/admin-up
IP unnumbered interface (loopback0)
IPv6 routing is disabled
  No auth type and keychain
  Auth check set
Index: 0x0002, Local Circuit ID: 0x01, Circuit Type: L1
BFD IPv4 is locally disabled for Interface Ethernet1/41
BFD IPv6 is locally disabled for Interface Ethernet1/41
MTR is disabled
Extended Local Circuit ID: 0x1A028000, P2P Circuit ID: 0000.0000.0000.00
Retx interval: 5, Retx throttle interval: 66 ms
LSP interval: 33 ms, MTU: 9192
P2P Adjs: 1, AdjsUp: 1, Priority 64
Hello Interval: 10, Multi: 3, Next IIH: 00:00:01
MT Adjs AdjsUp Metric CSNP Next CSNP Last LSP ID
1 1 1 4 60 00:00:35 ffff.ffff.ffff.ff-ff
2 0 0 4 60 Inactive ffff.ffff.ffff.ff-ff
Topologies enabled:
  L MT Metric MetricCfg Fwdng IPv4-MT IPv4Cfg IPv6-MT IPv6Cfg
  1 0 4 no UP UP yes DN no
  2 0 4 no UP DN no DN no

Leaf-Switch-V1# show isis adjacency

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

```

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

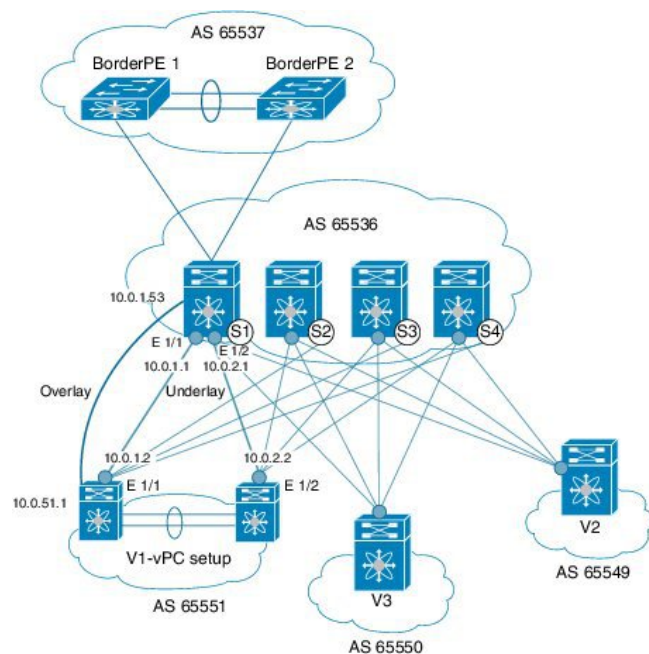
eBGP Underlay IP Network

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

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

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

Figure 7: eBGP as underlay



eBGP configuration sample

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

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

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

(config) #

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

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

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

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

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

(config) #

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

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

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

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

(config) #

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

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



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

The overlay sessions are configured on loopback addresses.

(config) #

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

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

(config) #

```
vrf context coke
  vni 50000
  rd auto
  address-family ipv4 unicast
    route-target both auto
    route-target both auto evpn
  address-family ipv6 unicast
    route-target both auto
    route-target both auto evpn
```

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

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

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

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

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

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

(config) #

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

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

Underlay session on leaf switch VTEP V1 towards spine switch S1

(config) #

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

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

Overlay session towards spine switch S1

(config) #

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

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

vPC backup session

(config) #

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

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



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

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

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

BGP Verification

Use the following commands for verifying BGP configuration:

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

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

Multicast Routing in the VXLAN Underlay

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

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

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

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

PIM ASM and PIM Bidir Underlay IP Network

Some multicast topology design pointers are given below:

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

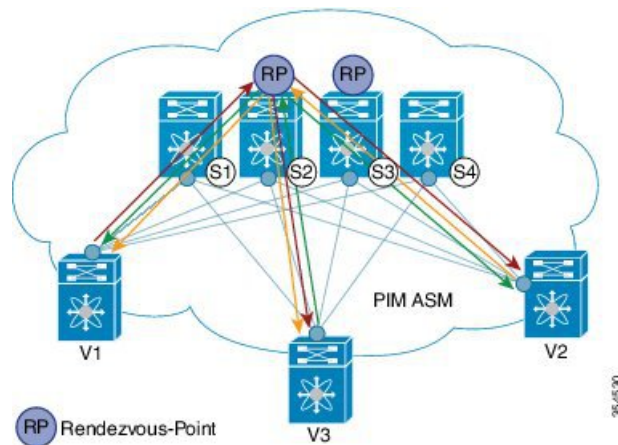
- Reserve a range of multicast groups (destination groups/DGroups) to service the overlay and optimize for diverse VNIs.
- In a spine-leaf topology with a lean spine,
 - Use multiple Rendezvous-Points across multiple spine switches.
 - Use redundant Rendezvous-Points.
 - Map different VNIs to different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

**Important**

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

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

Figure 8: PIM ASM as the IP multicast routing protocol



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

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

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

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

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

Programmable Fabric specific pointers are:

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

PIM ASM Configuration



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

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

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

PIM Anycast Rendezvous-Point association on leaf switch V1

(config) #

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

198.51.100.220 is the Anycast Rendezvous-Point IP address.

Loopback interface PIM configuration on leaf switch V1

(config) #

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

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

(config) #

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

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

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

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

(config) #

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

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

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

Rendezvous Point Configuration on the spine switches**PIM configuration on spine switch S2**

(config) #

```
feature pim
```

Loopback Interface Configuration (RP)

(config) #

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

Loopback interface configuration (Anycast RP)

(config) #

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

Anycast-RP configuration on spine switch S2

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

(config) #

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



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

Non-RP Spine Switch Configuration

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

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

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

(config) #

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

Loopback interface configuration (non RP)

(config) #

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

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

(config) #

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

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

PIM ASM Verification

Use the following commands for verifying PIM ASM configuration:

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

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

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

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

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

Leaf-Switch-V1# **show ip pim rp**

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

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

Leaf-Switch-V1# **show ip pim interface**

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

```

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

```

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

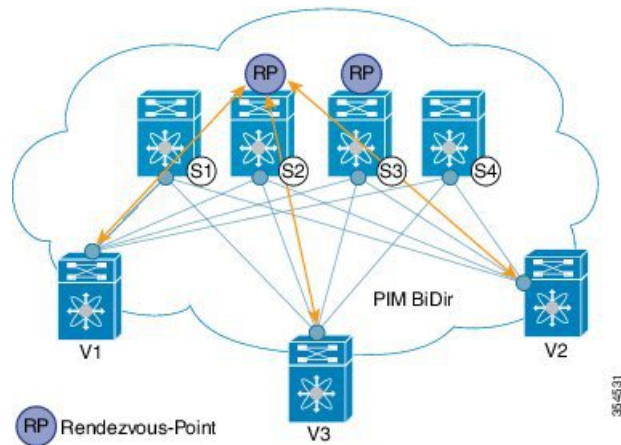
PIM Neighbor Status for VRF "default"

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

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

PIM Bidirectional (BiDir)

Figure 9: PIM BiDir as the IP multicast routing protocol



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

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



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

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

Programmable Fabric specific pointers are:

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

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

PIM BiDir Configuration

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



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

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

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

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

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

Leaf switch V1 configuration

Phantom Rendezvous-Point association on leaf switch V1

(config) #

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

Loopback interface PIM configuration on leaf switch V1

(config) #

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

IP unnumbered P2P interface configuration on leaf switch V1

(config) #

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

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

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

Using phantom RP on spine switch S2

(config) #

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

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

(config) #

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

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

(config) #

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

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

(config) #

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

(config) #

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

Using phantom RP on spine switch S3

(config) #


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

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

(config) #

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

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

(config) #

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

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

(config) #

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

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

PIM BiDir Verification

Use the following commands for verifying PIM BiDir configuration:

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

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

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

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

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

```

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

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

```

Leaf-Switch-V1# **show ip pim rp**

```

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

```

```

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

```

Leaf-Switch-V1# **show ip pim interface**

```

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

```

```

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

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

```

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

PIM Neighbor Status for VRF "default"

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

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

Underlay deployment without multicast (Ingress replication)

Ingress replication is supported on Cisco Nexus 9000 Series switches.



CHAPTER 5

Configuring VXLAN BGP EVPN

This chapter contains the following sections:

- [About VXLAN BGP EVPN, on page 65](#)
- [Guidelines and Limitations for VXLAN BGP EVPN, on page 66](#)
- [Configuring VXLAN BGP EVPN, on page 70](#)

About VXLAN BGP EVPN

About RD Auto

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

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

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

Example auto-derived Route Distinguisher (RD)

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

About Route-Target Auto

The auto-derived Route-Target (route-target import/export/both auto) is based on the Type 0 encoding format as described in IETF RFC 4364 section 4.2 (<https://tools.ietf.org/html/rfc4364#section-4.2>). IETF RFC 4364 section 4.2 describes the Route Distinguisher format and IETF RFC 4364 section 4.3.1 refers that it is desirable

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

2-byte ASN

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

4-byte ASN

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

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

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



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

Guidelines and Limitations for VXLAN BGP EVPN

VXLAN BGP EVPN has the following guidelines and limitations:

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

For more information, see the [Cisco Nexus 9000 Series NX-OS System Management Configuration Guide, Release 9.3\(x\)](#).

- When SVI is enabled on a VTEP (flood and learn, or EVPN) regardless of ARP suppression, make sure that ARP-ETHER TCAM is carved using the **hardware access-list tcam region arp-ether 256 double-wide** command. This requirement does not apply to Cisco Nexus 9200, 9300-EX, and 9300-FX/FX2 platform switches and Cisco Nexus 9500 platform switches with 9700-EX/FX line cards.
- For the Cisco Nexus 9504 and 9508 with R-series line cards, VXLAN EVPN (Layer 2 and Layer 3) is only supported with the 9636C-RX and 96136YC-R line cards.
- You can configure EVPN over segment routing or MPLS. See the [Cisco Nexus 9000 Series NX-OS Label Switching Configuration Guide, Release 9.3\(x\)](#) for more information.
- You can use MPLS tunnel encapsulation using the new CLI encapsulation `mpls` command. You can configure the label allocation mode for the EVPN address family. See the [Cisco Nexus 9000 Series NX-OS Label Switching Configuration Guide, Release 9.3\(x\)](#) for more information.
- In a VXLAN EVPN setup that has 2K VNI scale configuration, the control plane down time may take more than 200 seconds. To avoid potential BGP flap, extend the graceful restart time to 300 seconds.
- Starting from Cisco NX-OS Release 9.3(5), new VXLAN uplink capabilities are introduced:
 - A physical interface in default VRF is supported as VXLAN uplink.
 - A parent interface in default VRF, carrying subinterfaces with VRF and dot1q tags, is supported as VXLAN uplink.
 - A subinterface in any VRF and/or with dot1q tag remains not supported as VXLAN uplink.
 - An SVI in any VRF remains not supported as VXLAN uplink.
 - In vPC with physical peer-link, a SVI can be leveraged as backup underlay, default VRF only between the vPC members (infra-VLAN, system nve infra-vlans).
 - On a vPC pair, shutting down NVE or NVE loopback on one of the vPC nodes is not a supported configuration. This means that traffic failover on one-side NVE shut or one-side loopback shut is not supported.
 - FEX host interfaces remain not supported as VXLAN uplink and cannot have VTEPs connected (BUD node).
- In a VXLAN EVPN setup, border nodes must be configured with unique route distinguishers, preferably using the **auto rd** command. Not using unique route distinguishers across all border nodes is not supported. The use of unique route distinguishers is strongly recommended for all VTEPs of a fabric.
- ARP suppression is only supported for a VNI if the VTEP hosts the First-Hop Gateway (Distributed Anycast Gateway) for this VNI. The VTEP and the SVI for this VLAN have to be properly configured for the distributed Anycast Gateway operation, for example, global Anycast Gateway MAC address configured and Anycast Gateway feature with the virtual IP address on the SVI.
- The ARP suppression setting must match across the entire fabric. For a specific VNID, all VTEPs must be either configured or not configured.
- Mobility Sequence number of a locally originated type-2 route (MAC/MAC-IP) can be mismatched between vPC peers, with one vTEP having a sequence number K while other vTEP in the same complex

can have the same route with sequence number 0. This does not cause any functional impact and the traffic is not impacted even after the host moves.

- DHCP snooping (Dynamic Host Configuration Protocol snooping) is not supported on VXLAN VLANs.
- RACLs are not supported on VXLAN uplink interfaces. VACLs are not supported on VXLAN de-capsulated traffic in egress direction; this applies for the inner traffic coming from network (VXLAN) towards the access (Ethernet).

As a best practice, always use PACLS/VACLs for the access (Ethernet) to the network (VXLAN) direction. See the [Cisco Nexus 9000 Series NX-OS Security Configuration Guide, Release 9.3\(x\)](#) for other guidelines and limitations for the VXLAN ACL feature.

- The Cisco Nexus 9000 QoS buffer-boost feature is not applicable for VXLAN traffic.
- 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 VXLAN BGP EVPN fabrics with EBGp, the following recommendations are applicable:
 - It is recommended to use loopbacks for the EBGp EVPN peering sessions (overlay control-plane).
 - It is a best practice to use the physical interfaces for EBGp IPv4/IPv6 peering sessions (underlay).
- Bind the NVE source-interface to a dedicated loopback interface and do not share this loopback with any function or peerings of Layer-3 protocols. A best practice is to use a dedicated loopback address for the VXLAN VTEP function.
- You must bind NVE to a loopback address that is separate from other loopback addresses that are required by Layer 3 protocols. NVE and other Layer 3 protocols using the same loopback is not supported.
- The NVE source-interface loopback is required to be present in the default VRF.
- Only EBGp peering between a VTEP and external nodes (Edge Router, Core Router or VNF) is supported.
 - EBGp peering from the VTEP to the external node using a physical interface or subinterfaces is recommended and it is a best practice (external connectivity).
 - The EBGp peering from the VTEP to the external node can be in the default VRF or in a tenant VRF (external connectivity).
 - The EBGp peering from the VTEP to an external node over VXLAN must be in a tenant VRF and must use the update-source of a loopback interface (peering over VXLAN).
 - Using an SVI for EBGp peering on a from the VTEP to the External Node requires the VLAN to be local (not VXLAN extended).
- When configuring VXLAN BGP EVPN, only the "System Routing Mode: Default" is applicable for the following hardware platforms:
 - Cisco Nexus 9300 platform switches
 - Cisco Nexus 9300-EX platform switches
 - Cisco Nexus 9300-FX/FX2 platform switches
 - Cisco Nexus 9500 platform switches with X9500 line cards

- Cisco Nexus 9500 platform switches with X9700-EX and X9700-FX line cards
- Changing the “System Routing Mode” requires a reload of the switch.
- Cisco Nexus 9516 platform is not supported for VXLAN EVPN.
- VXLAN is supported on Cisco Nexus 9500 platform switches with the following line cards:
 - 9500-R
 - 9564PX
 - 9564TX
 - 9536PQ
 - 9700-EX
 - 9700-FX
- Cisco Nexus 9500 platform switches with 9700-EX or -FX line cards support 1G, 10G, 25G, 40G, 100G and 400G for VXLAN uplinks.
- Cisco Nexus 9200 and 9300-EX/FX/FX2/FX3 and -GX support 1G, 10G, 25G, 40G, 100G and 400G for VXLAN uplinks.
- The Cisco Nexus 9000 platform switches use standards conforming UDP port number 4789 for VXLAN encapsulation. This value is not configurable.
- The Cisco Nexus 9200 platform switches with Application Spine Engine (ASE2) have throughput constraints for packet sizes of 99-122 bytes; packet drops might be experienced.
- The VXLAN network identifier (VNID) 16777215 is reserved and should explicitly not be configured.
- Non-Disruptive In Service Software Upgrade (ND-ISSU) is supported on Nexus 9300 with VXLAN enabled. Exception is ND-ISSU support for Cisco Nexus 9300-FX3 and 9300-GX platform switch.
- Gateway functionality for VXLAN to MPLS (LDP), VXLAN to MPLS-SR (Segment Routing) and VXLAN to SRv6 can be operated on the same Cisco Nexus 9000 Series platform.
 - VXLAN to MPLS (LDP) Gateway is supported on the Cisco Nexus 3600-R and the Cisco Nexus 9500 with R-Series line cards.
 - VXLAN to MPLS-SR Gateway is supported on the Cisco Nexus 9300-FX2/FX3/GX and Cisco Nexus 9500 with R-Series line cards.
 - VXLAN to SRv6 is supported on the Cisco Nexus 9300-GX platform.
 - Multiple Tunnel Encapsulations (VXLAN, GRE and/or MPLS, static label or segment routing) can not co-exist on the same Cisco Nexus 9000 Series switch with Network Forwarding Engine (NFE).
- Resilient hashing is supported on the following switch platform with a VXLAN VTEP configured:
 - Cisco Nexus 9300-EX/FX/FX2/FX3/GX support ECMP resilient hashing.
 - Cisco Nexus 9300 with ALE uplink ports does not support resilient hashing.



Note Resilient hashing is disabled by default.

- It is recommended to use the **vpc orphan-ports suspend** command for single attached and/or routed devices on a Cisco Nexus 9000 platform switch acting as vPC VTEP.



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

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	feature vn-segment-vlan-based	Enable VN-Segment for VLANs.
Step 4	feature interface-vlan	Enable Switch Virtual Interface (SVI).
Step 5	nv overlay evpn	Enable the EVPN control plane for VXLAN.

Configuring VLAN and VXLAN VNI



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

Procedure

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

	Command or Action	Purpose
Step 3	evpn	Enter EVI (EVPN Virtual Instance) configuration mode.
Step 4	vni number 12	Specify the Service Instance (VNI) for the EVI.
Step 5	rd auto	Specify the MAC-VRF's route distinguisher (RD).
Step 6	route-target both {auto rt}	<p>Configure the route target (RT) for import and export of MAC prefixes. The RT is used for a per-MAC-VRF prefix import/export policy. If you enter an RT, the following formats are supported: ASN2:NN, ASN4:NN, or IPV4:NN.</p> <p>Note Specifying the auto option is applicable only for IBGP.</p> <p>Manually configured route targets are required for EBGp and for asymmetric VNIs.</p>

Configuring VRF for VXLAN Routing

Configure the tenant VRF.



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

Procedure

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

	Command or Action	Purpose
		Note Specifying the auto option is applicable only for IBGP. Manually configured route targets are required for EBGP.
Step 6	route-target both {auto rt} evpn	Configure the route target (RT) for import and export of IPv4 or IPv6 prefixes. The RT is used for a per-VRF prefix import/export policy. If you enter an RT, the following formats are supported: ASN2:NN, ASN4:NN, or IPV4:NN. Note Specifying the auto option is applicable only for IBGP. Manually configured route targets are required for EBGP.

Configuring SVI for Core-facing VXLAN Routing

Configure the core-facing SVI VRF.

Procedure

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

Configuring SVI for Host-Facing VXLAN Routing

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

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 should have the same virtual MAC address.
Step 2	vlan <i>number</i>	Specify VLAN.
Step 3	vn-segment <i>number</i>	Specify vn-segment.
Step 4	interface <i>vlan-number</i>	Specify VLAN interface.
Step 5	vrf member <i>vrf-name</i>	Assign to VRF.
Step 6	ip address <i>address</i>	Specify IP address.
Step 7	fabric forwarding mode anycast-gateway	Associate SVI with anycast gateway under VLAN configuration mode.

Configuring the NVE Interface and VNIs Using Multicast

Procedure

	Command or Action	Purpose
Step 1	interface <i>nve-interface</i>	Configure the NVE interface.
Step 2	source-interface loopback1	Binds the NVE source-interface to a dedicated loopback interface.
Step 3	host-reachability protocol bgp	This defines BGP as the mechanism for host reachability advertisement
Step 4	global mcast-group <i>ip-address</i> {L2 L3}	Configures the mcast group globally (for all VNI) on a per-NVE interface basis. This applies and gets inherited s to all Layer 2 or Layer 3 VNIs. Note Layer3 macst group is only used for Tenant Routed Multicast (TRM).

	Command or Action	Purpose
Step 5	member vni <i>vni</i>	Add Layer 2 VNIs to the tunnel interface.
Step 6	mcast-group <i>ip address</i>	Configure the mcast group on a per-VNI basis. Add Layer 2 VNI specific mcast group and override the global set configuration. Note Instead of a mcast group, ingress replication can be configured.
Step 7	member vni <i>vni</i> associate-vrf	Add Layer-3 VNIs, one per tenant VRF, to the overlay. Note Required for VXLAN routing only.
Step 8	mcast-group <i>address</i>	Configure the mcast group on a per-VNI basis. Add Layer 3 VNI specific mcast group and override the global set configuration.

Configuring VXLAN EVPN Ingress Replication

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



Note VXLAN EVPN ingress replication is supported on:

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

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

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

Procedure

	Command or Action	Purpose
Step 1	interface <i>nve-interface</i>	Configure the NVE interface.

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

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.

	Command or Action	Purpose
Step 3	neighbor <i>address</i> remote-as <i>number</i>	Define MPBGP neighbors. Under each neighbor define L2VPN EVPN.
Step 4	address-family <i>l2vpn evpn</i>	Configure address family Layer 2 VPN EVPN under the BGP neighbor. Note Address-family IPv4 EVPN for VXLAN host-based routing
Step 5	(Optional) Allowas-in	Only for EBGp deployment cases: Allows duplicate autonomous system (AS) numbers in the AS path. Configure this parameter on the leaf for eBGP when all leafs are using the same AS, but the spines have a different AS than leafs.
Step 6	send-community <i>extended</i>	Configures community for BGP neighbors.
Step 7	vrf <i>vrf-name</i>	Specify VRF.
Step 8	address-family <i>ipv4 unicast</i>	Configure the address family for IPv4.
Step 9	advertise <i>l2vpn evpn</i>	Enable advertising EVPN routes. Note Beginning with Cisco NX-OS Release 9.2(1), the advertise l2vpn evpn command no longer takes effect. To disable advertisement for a VRF toward the EVPN, disable the VNI in NVE by entering the no member vni vni associate-vrf command in interface nve1. The <i>vni</i> is the VNI associated with that particular VRF.
Step 10	maximum-paths <i>path {ibgp}</i>	Enable ECMP for EVPN transported IP Prefixes within the IPv6 address-family of the respective VRF.
Step 11	address-family <i>ipv6 unicast</i>	Configure the address family for IPv6.
Step 12	advertise <i>l2vpn evpn</i>	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.

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

Configuring iBGP for EVPN on the Spine

Procedure

	Command or Action	Purpose
Step 1	router bgp <i>autonomous system number</i>	Specify BGP.
Step 2	neighbor <i>address</i> remote-as <i>number</i>	Define neighbor.
Step 3	address-family l2vpn evpn	Configure address family Layer 2 VPN EVPN under the BGP neighbor.
Step 4	send-community extended	Configures community for BGP neighbors.
Step 5	route-reflector-client	Enable Spine as Route Reflector.
Step 6	retain route-target all	Configure retain route-target all under address-family Layer 2 VPN EVPN [global]. Note Required for eBGP. Allows the spine to retain and advertise all EVPN routes when there are no local VNI configured with matching import route targets.
Step 7	address-family l2vpn evpn	Configure address family Layer 2 VPN EVPN under the BGP neighbor.
Step 8	disable-peer-as-check	Disables checking the peer AS number during route advertisement. Configure this parameter on the spine for eBGP when all leafs are using the same AS but the spines have a different AS than leafs. Note Required for eBGP.
Step 9	route-map permitall out	Applies route-map to keep the next-hop unchanged. Note Required for eBGP.

Configuring eBGP for EVPN on the Spine

Procedure

	Command or Action	Purpose
Step 1	route-map NEXT-HOP-UNCH permit 10	Configure route-map to keep the next-hop unchanged for EVPN routes.
Step 2	set ip next-hop unchanged	<p>Set next-hop address.</p> <p>Note When two next hops are enabled, next hop ordering is not maintained.</p> <p>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.</p> <p>Directly/locally connected next hops are always given priority over remotely connected next hops.</p>
Step 3	router bgp <i>autonomous system number</i>	Specify BGP.
Step 4	address-family l2vpn evpn	Configure address family Layer 2 VPN EVPN under the BGP neighbor.
Step 5	retain route-target all	<p>Configure retain route-target all under address-family Layer 2 VPN EVPN [global].</p> <p>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.</p>
Step 6	neighbor <i>address</i> remote-as <i>number</i>	Define neighbor.
Step 7	address-family l2vpn evpn	Configure address family Layer 2 VPN EVPN under the BGP neighbor.
Step 8	disable-peer-as-check	Disables checking the peer AS number during route advertisement. Configure this parameter on the spine for eBGP when all leafs are using the same AS but the spines have a different AS than leafs.
Step 9	send-community extended	Configures community for BGP neighbors.

	Command or Action	Purpose
Step 10	route-map NEXT-HOP-UNCH out	Applies route-map to keep the next-hop unchanged.

Suppressing ARP

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



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

Procedure

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

Disabling VXLANs

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

For IP addresses:

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

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

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

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

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

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

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

Command	Description
<pre>switch(config)# fabric forwarding ? anycast-gateway-mac dup-host-ip-addr-detection</pre>	<p>Available sub-commands:</p> <ul style="list-style-type: none"> • Anycast gateway MAC of the switch. • To detect duplicate host addresses in n seconds.

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

For MAC addresses:

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

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

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

Wherever a MAC address is permanently frozen, a syslog message will be 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
```

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

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

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

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

Command	Description
<pre>switch(config)# l2rib dup-host-mac-detection ? <1-1000> default</pre>	<p>Available sub-commands for L2RIB:</p> <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).
<pre>switch(config)# l2rib dup-host-mac-detection 100 ? <2-36000></pre>	The duplicate detection timeout in seconds for the number of host moves. The range is 2 to 36000 seconds; default is 180 seconds.
<pre>switch(config)# l2rib dup-host-mac-detection 100 10</pre>	Detects duplicate host addresses (limited to 100 moves) in a period of 10 seconds.

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.
show vxlan interface count	<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>
show l2route evpn mac [all evi <i>evi</i> [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.

Command	Purpose
<code>show l2route evpn mac-ip all</code> <code>show l2route evpn mac-ip all detail</code>	Displays all MAC IP routes.
<code>show l2route topology</code>	Displays Layer 2 route topology.

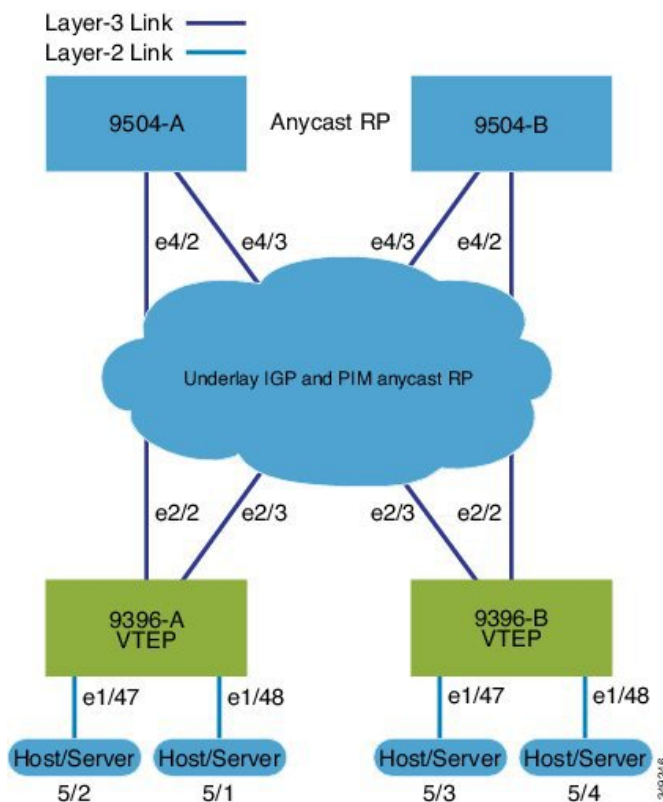


Note Although the `show ip bgp` command is available for verifying a BGP configuration, as a best practice, it is preferable to use the `show bgp` command instead.

Example of VXLAN BGP EVPN (IBGP)

An example of a VXLAN BGP EVPN (IBGP):

Figure 10: VXLAN BGP EVPN Topology (IBGP)



IBGP between Spine and Leaf

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

- Enable the relevant protocols

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

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

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

- Configure Loopback for 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 anycast-rp 100.1.1.1 10.1.1.1
ip pim anycast-rp 100.1.1.1 20.1.1.1
```

- Enable OSPF for underlay routing

```
router ospf 1
```

- Configure interfaces for Spine-leaf interconnect

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

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

- Configure BGP

```
router bgp 65535
router-id 10.1.1.1
 neighbor 30.1.1.1 remote-as 65535
 update-source loopback0
```



```

address-family l2vpn evpn
 send-community both
 route-reflector-client
neighbor 40.1.1.1 remote-as 65535
update-source loopback0
address-family l2vpn evpn
 send-community both
 route-reflector-client

```

- Spine (9504-B)

- Enable the EVPN control plane

```
nv overlay evpn
```

- Enable the relevant Protocols

```

feature ospf
feature bgp
feature pim

```

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

```

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

```

- Configure Loopback for 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 AnycastRP

```

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

```

- Configure Anycast RP

```

ip pim rp-address 100.1.1.1 group-list 224.0.0.0/4
ip pim anycast-rp 100.1.1.1 10.1.1.1
ip pim anycast-rp 100.1.1.1 20.1.1.1

```

- Enable OSPF for underlayrouting

```
router ospf 1
```

- Configure interfaces for Spine-leaf interconnect

```

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

```

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

- Configure BGP

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

- Leaf (9396-A)

- Enable the EVPN control plane

```
nv overlay evpn
```

- Enable the relevant protocols

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

- Enable VXLAN with distributed anycast-gateway using BGP EVPN

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

- Enabling OSPF for underlay routing

```
router ospf 1
```

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

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

- Configure Loopback for local VTEP IP, 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
 shutdown
```

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

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

- Configure PIM RP

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

- Create VLANs

```
vlan 1001-1002
```

- Create overlay VRF VLAN and configure vn-segment

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

- Create overlay VRF VLAN and configure vn-segment

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

- Configure Core-facing SVI for VXLAN routing

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

- Create VLAN and provide mapping to VXLAN

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

- Create VRF and configure VNI

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



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

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

- Create server facing SVI and enable distributed anycast-gateway.

```
interface vlan1001
no shutdown
vrf member vxlan-900001
ip address 4.1.1.1/24 tag 54321
ipv6 address 4:1:0:1::1/64 tag 54321
fabric forwarding mode anycast-gateway

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

- Configure ACL TCAM region for ARP suppression



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

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



**Note** You can choose either of the following two options for creating the NVE interface. Use Option 1 for a small number of VNIs. Use Option 2 to leverage the simplified configuration mode.

Create the network virtualization endpoint (NVE) interface

Option 1

```
interface nve1
no shutdown
```

```

source-interface loopback1
host-reachability protocol bgp
member vni 900001 associate-vrf
member vni 2001001
 mcast-group 239.0.0.1
member vni 2001002
 mcast-group 239.0.0.1

```

## Option 2

```

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

```

- Configure interfaces for hosts/servers

```

interface Ethernet1/47
switchport
switchport access vlan 1002

interface Ethernet1/48
switchport
switchport access vlan 1001

```

- Configure BGP

```

router bgp 65535
router-id 30.1.1.1
neighbor 10.1.1.1 remote-as 65535
 update-source loopback0
 address-family l2vpn evpn
 send-community both
neighbor 20.1.1.1 remote-as 65535
 update-source loopback0
 address-family l2vpn evpn
 send-community both
vrf vxlan-900001
 address-family ipv4 unicast
 redistribute direct route-map HOST-SVI
 address-family ipv6 unicast
 redistribute direct route-map HOST-SVI

```




---

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

---

```

evpn
vni 2001001 l2
vni 2001002 l2

```



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

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



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



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

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

- Leaf (9396-B)

- Enable the EVPN control plane

```
nv overlay evpn
```

- Enable the relevant protocols

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

- Enable VxLAN with distributed anycast-gateway using BGP EVPN

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

- Enabling OSPF for underlayrouting

```
router ospf 1
```

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

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

- Configure Loopback for local VTEP IP, and BGP

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

- Configure interfaces for Spine-leaf interconnect

```
interface Ethernet2/2
 no switchport
 ip address 192.168.3.22/24
 ip router ospf 1 area 0.0.0.0
 ip pim sparse-mode
 no shutdown
```

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

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

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

- Configure PIM RP

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

- Create VLANs

```
vlan 1001-1002
```

- Create overlay VRF VLAN and configure vn-segment

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

- Configure Core-facing SVI for VXLAN routing

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

- Create VLAN and provide mapping to VXLAN

```
vlan 1001
 vn-segment 2001001
```

```
vlan 1002
 vn-segment 2001002
```

- Create VRF and configure VNI

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



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

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

- Create server facing SVI and enable distributed anycast-gateway

```
interface vlan1001
 no shutdown
 vrf member vxlan-900001
 ip address 4.1.1.1/24
 ipv6 address 4:1:0:1::1/64
 fabric forwarding mode anycast-gateway

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

- Configure ACL TCAM region for ARP suppression



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

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



**Note** You can choose either of the following two command procedures for creating the NVE interfaces. Use Option 1 for a small number of VNIs. Use Option 2 to leverage the simplified configuration mode.

Create the network virtualization endpoint (NVE) interface

Option 1



```
interface nve1
 no shutdown
 source-interface loopback1
 host-reachability protocol bgp
 member vni 900001 associate-vrf
 member vni 2001001
 mcast-group 239.0.0.1
 member vni 2001002
 mcast-group 239.0.0.1
```

## Option 2

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

- Configure interfaces for hosts/servers

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

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

- Configure BGP

```
router bgp 65535
 router-id 40.1.1.1
 neighbor 10.1.1.1 remote-as 65535
 update-source loopback0
 address-family l2vpn evpn
 send-community both
 neighbor 20.1.1.1 remote-as 65535
 update-source loopback0
 address-family l2vpn evpn
 send-community both
 vrf vxlan-900001
 vrf vxlan-900001
 address-family ipv4 unicast
 redistribute direct route-map HOST-SVI
 address-family ipv6 unicast
 redistribute direct route-map HOST-SVI
```



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

```
evpn
vni 2001001 12
vni 2001002 12
```



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

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



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

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



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

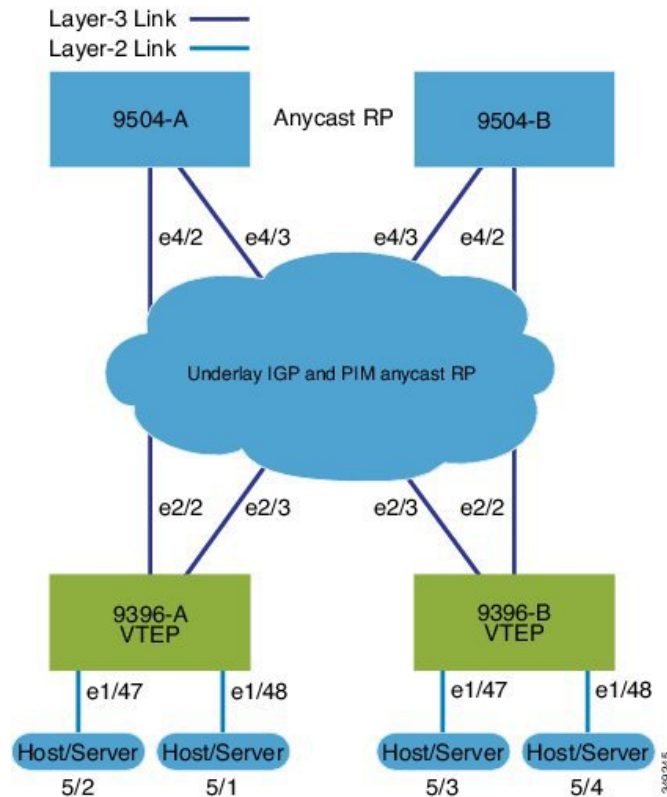
```
ip prefix-list vip_ip seq 5 permit 192.0.2.1/32
ip prefix-list vip_route_nh seq 5 permit 198.51.100.1/32

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

## Example of VXLAN BGP EVPN (EBGP)

An example of a VXLAN BGP EVPN (EBGP):

Figure 11: VXLAN BGP EVPN Topology (EBGP)



## EBGP between Spine and Leaf

## • Spine (9504-A)

- Enable the EVPN control plane

```
nv overlay evpn
```

- Enable the relevant protocols

```
feature bgp
feature pim
```

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

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

- Configure Loopback for Anycast RP

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

- Configure Anycast RP

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

```
ip pim anycast-rp 100.1.1.1 10.1.1.1
ip pim anycast-rp 100.1.1.1 20.1.1.1
```

- Configure route-map used by EBGp for Spine

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

- Configure route-map to Redistribute Loopback

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

- Configure interfaces for Spine-leaf interconnect

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

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

- Configure the BGP overlay for the EVPN address family.

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

- Configure BGP underlay for the IPv4 unicast address family.

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

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

- Spine (9504-B)

- Enable the EVPN control plane

```
nv overlay evpn
```

- Enable the relevant protocols

```
feature bgp
feature pim
```

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

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

- Configure Loopback for AnycastRP

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

- Configure Anycast RP

```
ip pim rp-address 100.1.1.1 group-list 224.0.0.0/4
ip pim anycast-rp 100.1.1.1 10.1.1.1
ip pim anycast-rp 100.1.1.1 20.1.1.1
```

- Configure route-map used by EBGp for Spine

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

- Configure route-map to Redistribute Loopback

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

- Configure interfaces for Spine-leaf interconnect

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

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

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

- Configure BGP overlay for the EVPN address family

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

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

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

- Leaf (9396-A)

- Enable the EVPN control plane.

```
nv overlay evpn
```

- Enable the relevant protocols.

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

- Enable VXLAN with distributed anycast-gateway using BGP EVPN.

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

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

- Enabling OSPF for underlay routing.

```
router ospf 1
```

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

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

- Configure Loopback for VTEP.

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

- Configure interfaces for Spine-leaf interconnect.

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

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

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

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

- Enable PIM RP.

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

- Create VLANs.

```
vlan 1001-1002
```

- Create overlay VRF VLAN and configure vn-segment.

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

- Configure core-facing SVI for VXLAN routing.

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

- Create VLAN and provide mapping to VXLAN.

```

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

```

- Create VRF and configure VNI

```

vrf context vxlan-900001
 vni 900001
 rd auto

```



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

```

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

```

- Create server facing SVI and enable distributed anycast-gateway

```

interface vlan1001
 no shutdown
 vrf member vxlan-900001
 ip address 4.1.1.1/24 tag 54321
 ipv6 address 4:1:0:1::1/64 tag 54321
 fabric forwarding mode anycast-gateway

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

```

- Configure ACL TCAM region for ARP suppression



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

```

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

```



**Note**

You can choose either of the following two options for creating the NVE interface. Use Option 1 for a small number of VNIs. Use Option 2 to leverage the simplified configuration mode.

Create the network virtualization endpoint (NVE) interface

**Option 1**

```
interface nve1
 no shutdown
 source-interface loopback1
 host-reachability protocol bgp
 member vni 900001 associate-vrf
 member vni 2001001
 mcast-group 239.0.0.1
 member vni 2001002
 mcast-group 239.0.0.1
```

**Option 2**

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

- Configure interfaces for hosts/servers.

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

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

- Configure BGP underlay for the IPv4 unicast address family.

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

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

- Configure BGP overlay for the EVPN address family.

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



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

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



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

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



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

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

- Leaf (9396-B)

- Enable the EVPN control plane.

```
nv overlay evpn
```

- Enable the relevant protocols.

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

- Enable VXLAN with distributed anycast-gateway using BGP EVPN.

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

- Enabling OSPF for underlay routing.

```
router ospf 1
```

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

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

- Configure Loopback for VTEP.

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

- Configure interfaces for Spine-leaf interconnect.

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

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

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

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

- Enable PIM RP

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

- Create VLANs

```
vlan 1001-1002
```

- Create overlay VRF VLAN and configure vn-segment.

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

- Configure core-facing SVI for VXLAN routing.

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

- Create VLAN and provide mapping to VXLAN.

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

- Create VRF and configure VNI

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


**Note**

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

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

- Create server facing SVI and enable distributed anycast-gateway.

```
interface vlan1001
 no shutdown
 vrf member vxlan-900001
 ip address 4.1.1.1/24 tag 54321
 ipv6 address 4:1:0:1::1/64 tag 54321
 fabric forwarding mode anycast-gateway

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

- Configure ACL TCAM region for ARP suppression



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

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



**Note** You can choose either of the following two procedures for creating the NVE interface. Use Option 1 for a small number of VNIs. Use Option 2 to leverage the simplified configuration mode.

Create the network virtualization endpoint (NVE) interface.

#### Option 1

```
interface nve1
 no shutdown
 source-interface loopback1
 host-reachability protocol bgp
 member vni 900001 associate-vrf
 member vni 2001001
 mcast-group 239.0.0.1
 member vni 2001002
 mcast-group 239.0.0.1
```

#### Option 2

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

- Configure interfaces for hosts/servers

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

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

- Configure BGP underlay for the IPv4 unicast address family.

```

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

```

- Configure BGP overlay for the EVPN address family.

```

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

```



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

```

evpn
 vni 2001001 12
 vni 2001002 12

```



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

```

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

```



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

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

## Example Show Commands

### • show nve peers

```
9396-B# show nve peers
Interface Peer-IP State LearnType Uptime Router-Mac

nve1 30.1.1.1 Up CP 00:00:38 6412.2574.9f27
```

### • show nve vni

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

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

### • show ip arp suppression-cache detail

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

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

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



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

- **show vxlan interface**

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

- **show bgp l2vpn evpn summary**

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

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

- **show bgp l2vpn evpn**

```
leaf3# show bgp l2vpn evpn
BGP routing table information for VRF default, address family L2VPN EVPN
BGP table version is 60, local router ID is 40.0.0.4
Status: s-suppressed, x-deleted, S-stale, d-dampened, h-history, *-valid,
>-best
Path type: i-internal, e-external, c-confed, l-local, a-aggregate, r-redist,
I-injected
Origin codes: i - IGP, e - EGP, ? - incomplete, | - multipath, & - backup

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

- **show l2route evpn mac all**

```
leaf3# show l2route evpn mac all
Topology Mac Address Prod Next Hop (s)

101 0000.8816.b645 BGP 40.0.0.2
101 0001.0000.0033 Local Ifindex 4362086
101 0001.0000.0035 Local Ifindex 4362086
101 0011.0000.0034 BGP 40.0.0.2
```

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

```
leaf3# show l2route evpn mac-ip all
Topology ID Mac Address Prod Host IP Next Hop (s)

101 0011.0000.0034 BGP 5.1.3.2 40.0.0.2
102 0011.0000.0034 BGP 5.1.3.2 40.0.0.2
```





## CHAPTER 6

# Configuring External VRF Connectivity and Route Leaking

---

This chapter contains the following sections:

- [Configuring External VRF Connectivity, on page 109](#)
- [Configuring Route Leaking, on page 123](#)

## Configuring External VRF Connectivity

### About External Layer-3 Connectivity for VXLAN BGP EVPN Fabrics

A VXLAN BGP EVPN fabric can be extended by using per-VRF IP routing to achieve external connectivity. The approach that is used for the Layer-3 extensions is commonly referred to as VRF Lite, while the functionality itself is more accurately defined as Inter-AS Option A or back-to-back VRF connectivity.

### Guidelines and Limitations for External VRF Connectivity and Route Leaking

The following guidelines and limitations apply to external Layer 3 connectivity for VXLAN BGP EVPN fabrics:

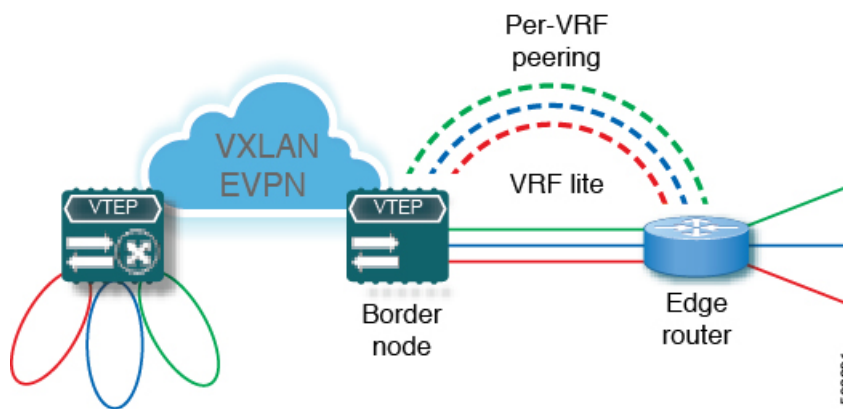
- A physical Layer 3 interface (parent interface) can be used for external Layer 3 connectivity (that is, VRF default).
- The parent interface to multiple subinterfaces cannot be used for external Layer 3 connectivity (that is, Ethernet1/1 for a VRF default). You can use a subinterface instead.
- VTEPs do not support VXLAN-encapsulated traffic over parent interfaces if subinterfaces are configured, regardless of VRF participation.
- VTEPs do not support VXLAN-encapsulated traffic over subinterfaces, regardless of VRF participation or IEEE 802.1Q encapsulation.
- Mixing subinterfaces for VXLAN and non-VXLAN VLANs is not supported.

### VXLAN BGP EVPN - VRF-lite brief

Some pointers are given below:

- The VXLAN BGP EVPN fabrics is depicted on the left in the following figure.
- Routes within the fabric are exchanged between all Edge-Devices (VTEPs) as well as Route-Reflectors; the control-plane used is MP-BGP with EVPN address-family.
- The Edge-Devices (VTEPs) acting as border nodes are configured to pass on prefixes to the external router (ER). This is achieved by exporting prefixes from MP-BGP EVPN to IPv4/IPv6 per-VRF peerings.
- Various routing protocols can be used for the per-VRF peering. While eBGP is the protocol of choice, IGP's like OSPF, IS-IS or EIGRP can be leveraged but require redistribution

Figure 12: External Layer-3 Connectivity - VRF-lite



## Configuring VXLAN BGP EVPN with eBGP for VRF-lite

### Configuring VRF for VXLAN Routing and External Connectivity using BGP

Configure the VRF on the border node.

#### Procedure

|               | Command or Action                     | Purpose                                                                                                                                                                                   |
|---------------|---------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 1</b> | <b>configure terminal</b>             | Enter global configuration mode.                                                                                                                                                          |
| <b>Step 2</b> | <b>vrf context</b> <i>vrf-name</i>    | Configure the VRF.                                                                                                                                                                        |
| <b>Step 3</b> | <b>vni</b> <i>number</i>              | Specify the VNI. The VNI associated with the VRF is often referred to as a Layer 3 VNI, L3VNI, or L3VPN. The L3VNI is configured as the common identifier across the participating VTEPs. |
| <b>Step 4</b> | <b>rd</b> { <i>auto</i>   <i>rd</i> } | Specify the VRF's route distinguisher (RD). The RD uniquely identifies a VTEP within an L3VNI. If you enter an RD, the following formats are supported: ASN2:NN, ASN4:NN, or IPV4:NN.     |

|               | Command or Action                             | Purpose                                                                                                                                                                                                                  |
|---------------|-----------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 5</b> | <b>address-family {ipv4   ipv6} unicast</b>   | Configure the IPv4 or IPv6 unicast address family.                                                                                                                                                                       |
| <b>Step 6</b> | <b>route-target both {auto   rt}</b>          | Configure the route target (RT) for import and export of IPv4 prefixes. The RT is used for a per-VRF prefix import/export policy. If you enter an RT, the following formats are supported: ASN2:NN, ASN4:NN, or IPV4:NN. |
| <b>Step 7</b> | <b>route-target both {auto   rt} evpn</b>     | Configure the route target (RT) for import and export of IPv4 prefixes. The RT is used for a per-VRF prefix import/export policy. If you enter an RT, the following formats are supported: ASN2:NN, ASN4:NN, or IPV4:NN. |
| <b>Step 8</b> | Repeat Step 1 through Step 7 for every L3VNI. |                                                                                                                                                                                                                          |

### Configuring the L3VNI's Fabric Facing VLAN and SVI on the Border Node

#### Procedure

|                | Command or Action                              | Purpose                                                            |
|----------------|------------------------------------------------|--------------------------------------------------------------------|
| <b>Step 1</b>  | <b>configure terminal</b>                      | Enter configuration mode.                                          |
| <b>Step 2</b>  | <b>vlan <i>number</i></b>                      | Specify the VLAN id that is used for the L3VNI.                    |
| <b>Step 3</b>  | <b>vn-segment <i>number</i></b>                | Map the L3VNI to the VLAN for VXLAN EVPN routing.                  |
| <b>Step 4</b>  | <b>interface <i>vlan-number</i></b>            | Specify the SVI (Switch Virtual Interface) for VXLAN EVPN routing. |
| <b>Step 5</b>  | <b>mtu <i>value</i></b>                        | Specify the MTU for the L3VNI.                                     |
| <b>Step 6</b>  | <b>vrf member <i>vrf-name</i></b>              | Map the SVI to the matching VRF context.                           |
| <b>Step 7</b>  | <b>ip forward</b>                              | Enable IPv4 forwarding for the L3VNI.                              |
| <b>Step 8</b>  | <b>no ip redirects</b>                         | Disable ICMP redirects                                             |
| <b>Step 9</b>  | <b>ipv6 <i>ip-address</i></b>                  | Enable IPv6 forwarding for the L3VNI.                              |
| <b>Step 10</b> | <b>no ipv6 redirects</b>                       | Disable ICMPv6 redirects.                                          |
| <b>Step 11</b> | Repeat Step 2 through Step 10 for every L3VNI. |                                                                    |

## Configuring the VTEP on the Border Node

## Procedure

|               | Command or Action                          | Purpose                                               |
|---------------|--------------------------------------------|-------------------------------------------------------|
| <b>Step 1</b> | <b>configure terminal</b>                  | Enter global configuration mode.                      |
| <b>Step 2</b> | <b>interface nve1</b>                      | Configure the NVE interface.                          |
| <b>Step 3</b> | <b>member vni <i>vni</i> associate-vrf</b> | Add Layer-3 VNIs, one per tenant VRF, to the overlay. |
| <b>Step 4</b> |                                            | Repeat Step 3 for every L3VNI.                        |

## Configuring the BGP VRF Instance on the Border Node for IPv4 per-VRF Peering

## Procedure

|                | Command or Action                                                                           | Purpose                                                                                                              |
|----------------|---------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------|
| <b>Step 1</b>  | <b>configure terminal</b>                                                                   | Enter global configuration mode.                                                                                     |
| <b>Step 2</b>  | <b>router bgp <i>autonomous-system-number</i></b>                                           | Configure BGP. The range of the <i>autonomous-system-number</i> is from 1 to 4294967295.                             |
| <b>Step 3</b>  | <b>vrf <i>vrf-name</i></b>                                                                  | Specify the VRF.                                                                                                     |
| <b>Step 4</b>  | <b>address-family ipv4 unicast</b>                                                          | Configure address family for IPv4.                                                                                   |
| <b>Step 5</b>  | <b>advertise l2vpn evpn</b>                                                                 | Enable the advertisement of EVPN routes within IPv4 address-family.                                                  |
| <b>Step 6</b>  | <b>maximum-paths ibgp <i>number</i></b>                                                     | Enabling equal cost multipathing (ECMP) for iBGP prefixes. The range for <i>number</i> is 1 to 64. The default is 1. |
| <b>Step 7</b>  | <b>maximum-paths <i>number</i></b>                                                          | Enabling equal cost multipathing (ECMP) for eBGP prefixes.                                                           |
| <b>Step 8</b>  | <b>neighbor <i>address</i> remote-as <i>number</i></b>                                      | Define eBGP neighbor IPv4 address and remote Autonomous-System (AS) number.                                          |
| <b>Step 9</b>  | <b>update-source <i>type/id</i></b>                                                         | Define interface for eBGP peering.                                                                                   |
| <b>Step 10</b> | <b>address-family ipv4 unicast</b>                                                          | Activate the IPv4 address family for IPv4 prefix exchange.                                                           |
| <b>Step 11</b> | Repeat Step 3 through Step 10 for every L3VNI that requires external connectivity for IPv4. |                                                                                                                      |

## Configuring the BGP VRF Instance on the Border Node for IPv6 per-VRF Peering

## Procedure

|         | Command or Action                                                                           | Purpose                                                                     |
|---------|---------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------|
| Step 1  | <b>configure terminal</b>                                                                   | Enter global configuration mode.                                            |
| Step 2  | <b>router bgp</b> <i>autonomous-system-number</i>                                           | Configure BGP.                                                              |
| Step 3  | <b>vrf</b> <i>vrf-name</i>                                                                  | Specify the VRF.                                                            |
| Step 4  | <b>address-family ipv6 unicast</b>                                                          | Configure address family for IPv4.                                          |
| Step 5  | <b>advertise l2vpn evpn</b>                                                                 | Enable the advertisement of EVPN routes within IPv6 address-family.         |
| Step 6  | <b>maximum-paths ibgp</b> <i>number</i>                                                     | Enabling equal cost multipathing (ECMP) for iBGP prefixes.                  |
| Step 7  | <b>maximum-paths</b> <i>number</i>                                                          | Enabling equal cost multipathing (ECMP) for eBGP prefixes.                  |
| Step 8  | <b>neighbor</b> <i>address</i> <b>remote-as</b> <i>number</i>                               | Define eBGP neighbor IPv6 address and remote Autonomous-System (AS) number. |
| Step 9  | <b>update-source</b> <i>type/id</i>                                                         | Define interface for eBGP peering.                                          |
| Step 10 | <b>address-family ipv6 unicast</b>                                                          | Configure address family for IPv6.                                          |
| Step 11 | Repeat Step 3 Through Step 10 for every L3VNI that requires external connectivity for IPv6. |                                                                             |

## Configuring the Sub-Interface Instance on the Border Node for Per-VRF Peering - Version 1

## Procedure

|        | Command or Action                        | Purpose                                                                                                  |
|--------|------------------------------------------|----------------------------------------------------------------------------------------------------------|
| Step 1 | <b>configure terminal</b>                | Enters global configuration mode.                                                                        |
| Step 2 | <b>interface</b> <i>type/id</i>          | Configure parent interface.                                                                              |
| Step 3 | <b>no switchport</b>                     | Disable Layer-2 switching mode on interface.                                                             |
| Step 4 | <b>no shutdown</b>                       | Bring up parent interface.                                                                               |
| Step 5 | <b>exit</b>                              | Exit interface configuration mode.                                                                       |
| Step 6 | <b>interface</b> <i>type/id</i>          | Define the Sub-Interface instance.                                                                       |
| Step 7 | <b>encapsulation dot1q</b> <i>number</i> | Configure the VLAN ID for the sub-interface. The <i>number</i> argument can have a value from 1 to 3967. |

|                | Command or Action                                       | Purpose                                            |
|----------------|---------------------------------------------------------|----------------------------------------------------|
| <b>Step 8</b>  | <b>vrf member</b> <i>vrf-name</i>                       | Map the Sub-Interface to the matching VRF context. |
| <b>Step 9</b>  | <b>ip address</b> <i>address</i>                        | Configure the Sub-Interfaces IP address.           |
| <b>Step 10</b> | <b>no shutdown</b>                                      | Bring up Sub-Interface.                            |
| <b>Step 11</b> | Repeat Step 5 through Step 9 for every per-VRF peering. |                                                    |

## VXLAN BGP EVPN - Default-Route, Route Filtering on External Connectivity

### About Configuring Default Routing for External Connectivity

For default-route advertisement into a VXLAN BGP EVPN fabric, we have to ensure that the default-route advertised into the fabric is at the same time not advertised outside of the fabric. For this case, it is necessary to have route filtering in place that prevents this eventuality.

### Configuring the Default Route in the Border Nodes VRF

#### Procedure

|               | Command or Action                         | Purpose                           |
|---------------|-------------------------------------------|-----------------------------------|
| <b>Step 1</b> | <b>configure terminal</b>                 | Enters global configuration mode. |
| <b>Step 2</b> | <b>vrf context</b> <i>vrf-name</i>        | Configure the VRF.                |
| <b>Step 3</b> | <b>ip route 0.0.0.0/0</b> <i>next-hop</i> | Configure the IPv4 default-route. |
| <b>Step 4</b> | <b>ipv6 route 0::/0</b> <i>next-hop</i>   | Configure the IPv6 default-route. |

### Configuring the BGP VRF Instance on the Border Node for IPv4/IPv6 Default-Route Advertisement

#### Procedure

|               | Command or Action                                 | Purpose                                                                                     |
|---------------|---------------------------------------------------|---------------------------------------------------------------------------------------------|
| <b>Step 1</b> | <b>configure terminal</b>                         | Enters global configuration mode.                                                           |
| <b>Step 2</b> | <b>router bgp</b> <i>autonomous-system-number</i> | Configure BGP.                                                                              |
| <b>Step 3</b> | <b>vrf</b> <i>vrf-name</i>                        | Specify the VRF.                                                                            |
| <b>Step 4</b> | <b>address-family ipv4 unicast</b>                | Configure the IPv4 Unicast address-family. Required for IPv6 over VXLAN with IPv4 underlay. |
| <b>Step 5</b> | <b>network 0.0.0.0/0</b>                          | Creating IPv4 default-route network statement.                                              |
| <b>Step 6</b> | <b>address-family ipv6 unicast</b>                | Configure the IPv6 unicast address-family.                                                  |

|                | Command or Action                                                                                               | Purpose                                                                     |
|----------------|-----------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------|
| <b>Step 7</b>  | <b>network 0::/0</b>                                                                                            | Creating IPv6 default-route network statement.                              |
| <b>Step 8</b>  | <b>neighbor <i>address</i> remote-as <i>number</i></b>                                                          | Define eBGP neighbor IPv4 address and remote Autonomous-System (AS) number. |
| <b>Step 9</b>  | <b>update-source <i>type/id</i></b>                                                                             | Define interface for eBGP peering                                           |
| <b>Step 10</b> | <b>address-family {ipv4   ipv6} unicast</b>                                                                     | Activate the IPv4 or IPv6 address family for IPv4/IPv6 prefix exchange.     |
| <b>Step 11</b> | <b>route-map <i>name</i> out</b>                                                                                | Attach route-map for egress route filtering.                                |
| <b>Step 12</b> | Repeat Step 3 through Step 11 for every L3VNI that requires external connectivity with default-route filtering. |                                                                             |

### Configuring Route Filtering for IPv4 Default-Route Advertisement

You can configure route filtering for IPv4 default-route advertisement.

#### Procedure

|               | Command or Action                                        | Purpose                                                                                                                  |
|---------------|----------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------|
| <b>Step 1</b> | <b>configure terminal</b>                                | Enters global configuration mode.                                                                                        |
| <b>Step 2</b> | <b>ip prefix-list <i>name</i> seq 5 permit 0.0.0.0/0</b> | Configure IPv4 prefix-list for default-route filtering.                                                                  |
| <b>Step 3</b> | <b>route-map <i>name</i> deny 10</b>                     | Create route-map with leading deny statement to prevent the default-route of being advertised via External Connectivity. |
| <b>Step 4</b> | <b>match ip address prefix-list <i>name</i></b>          | Match against the IPv4 prefix-list that contains the default-route.                                                      |
| <b>Step 5</b> | <b>route-map <i>name</i> permit 1000</b>                 | Create route-map with trailing allow statement to advertise non-matching routes via External Connectivity.               |

### Configuring Route Filtering for IPv6 Default-Route Advertisement

#### Procedure

|               | Command or Action                                      | Purpose                                                 |
|---------------|--------------------------------------------------------|---------------------------------------------------------|
| <b>Step 1</b> | <b>configure terminal</b>                              | Enters global configuration mode.                       |
| <b>Step 2</b> | <b>ipv6 prefix-list <i>name</i> seq 5 permit 0::/0</b> | Configure IPv6 prefix-list for default-route filtering. |

|               | Command or Action                                 | Purpose                                                                                                                  |
|---------------|---------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------|
| <b>Step 3</b> | <b>route-map</b> <i>name</i> <b>deny 10</b>       | Create route-map with leading deny statement to prevent the default-route of being advertised via External Connectivity. |
| <b>Step 4</b> | <b>match ipv6 address prefix-list</b> <i>name</i> | Match against the IPv6 prefix-list that contains the default-route.                                                      |
| <b>Step 5</b> | <b>route-map</b> <i>name</i> <b>permit 1000</b>   | Create route-map with trailing allow statement to advertise non-matching routes via External Connectivity.               |

### About Configuring Default-Route Distribution and Host-Route Filter

Per-default, a VXLAN BGP EVPN fabric always advertises all known routes via the External Connectivity. As not in all circumstances it is beneficial to advertise IPv4 /32 or IPv6 /128 Host-Routes, a respective route filtering approach can become necessary.

### Configuring the BGP VRF Instance on the Border Node for IPv4/IPv6 Host-Route Filtering

#### Procedure

|               | Command or Action                                                                                           | Purpose                                                                          |
|---------------|-------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------|
| <b>Step 1</b> | <b>configure terminal</b>                                                                                   | Enters global configuration mode.                                                |
| <b>Step 2</b> | <b>router bgp</b> <i>autonomous-system-number</i>                                                           | Configure BGP.                                                                   |
| <b>Step 3</b> | <b>vrf</b> <i>vrf-name</i>                                                                                  | Specify the VRF.                                                                 |
| <b>Step 4</b> | <b>neighbor</b> <i>address</i> <b>remote-as</b> <i>number</i>                                               | Define eBGP neighbor IPv4/IPv6 address and remote Autonomous-System (AS) number. |
| <b>Step 5</b> | <b>update-source</b> <i>type/id</i>                                                                         | Define interface for eBGP peering.                                               |
| <b>Step 6</b> | <b>address-family</b> { <i>ipv4</i>   <i>ipv6</i> } <b>unicast</b>                                          | Activate the IPv4 or IPv6 address family for IPv4/IPv6 prefix exchange.          |
| <b>Step 7</b> | <b>route-map</b> <i>name</i> <b>out</b>                                                                     | Attach route-map for egress route filtering.                                     |
| <b>Step 8</b> | Repeat Step 3 through Step 7 for every L3VNI that requires external connectivity with host-route filtering. |                                                                                  |

### Configuring Route Filtering for IPv4 Host-Route Advertisement

#### Procedure

|               | Command or Action         | Purpose                           |
|---------------|---------------------------|-----------------------------------|
| <b>Step 1</b> | <b>configure terminal</b> | Enters global configuration mode. |



|               | Command or Action                                              | Purpose                                                                                                                  |
|---------------|----------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------|
| <b>Step 2</b> | <b>ip prefix-list <i>name</i> seq 5 permit 0.0.0.0/0 eq 32</b> | Configure IPv4 prefix-list for host-route filtering.                                                                     |
| <b>Step 3</b> | <b>route-map <i>name</i> deny 10</b>                           | Create route-map with leading deny statement to prevent the default-route of being advertised via External Connectivity. |
| <b>Step 4</b> | <b>match ip address prefix-list <i>name</i></b>                | Match against the IPv4 prefix-list that contains the host-route.                                                         |
| <b>Step 5</b> | <b>route-map <i>name</i> permit 1000</b>                       | Create route-map with trailing allow statement to advertise non-matching routes via external connectivity.               |

### Configuring Route Filtering for IPv6 Host-Route Advertisement

#### Procedure

|               | Command or Action                                             | Purpose                                                                                                                  |
|---------------|---------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------|
| <b>Step 1</b> | <b>configure terminal</b>                                     | Enters global configuration mode.                                                                                        |
| <b>Step 2</b> | <b>ipv6 prefix-list <i>name</i> seq 5 permit 0::/0 eq 128</b> | Configure IPv4 prefix-list for host-route filtering.                                                                     |
| <b>Step 3</b> | <b>route-map <i>name</i> deny 10</b>                          | Create route-map with leading deny statement to prevent the default-route of being advertised via External Connectivity. |
| <b>Step 4</b> | <b>match ipv6 address prefix-list <i>name</i></b>             | Match against the IPv4 prefix-list that contains the host-route.                                                         |
| <b>Step 5</b> | <b>route-map <i>name</i> permit 1000</b>                      | Create route-map with trailing allow statement to advertise non-matching routes via External Connectivity.               |

### Example - Configuring VXLAN BGP EVPN with eBGP for VRF-lite

An example of external connectivity from VXLAN BGP EVPN to an external router using VRF-lite.

#### Configuring VXLAN BGP EVPN Border Node

The VXLAN BGP EVPN Border Node acts as neighbor device to the External Router. The VRF Name is purely localized and can be different to the VRF Name on the External Router, only significance is the L3VNI must be consistent across the VXLAN BGP EVPN fabric. For the ease of reading, the VRF and interface enumeration will be consistently used.

The configuration examples represents a IPv4 and IPv6 dual-stack approach; IPv4 or IPv6 can be substituted of each other.

```
vrf context myvrf_50001
 vni 50001
 rd auto
```

## Configuring Default-Route, Route Filtering on External Connectivity

```

 address-family ipv4 unicast
 route-target both auto
 route-target both auto evpn
 address-family ipv6 unicast
 route-target both auto
 route-target both auto evpn
!
vlan 2000
 vn-segment 50001
!
interface Vlan2000
 no shutdown
 mtu 9216
 vrf member myvrf_50001
 no ip redirects
 ip forward
 ipv6 address use-link-local-only
 no ipv6 redirects
!
interface nve1
 no shutdown
 host-reachability protocol bgp
 source-interface loopback1
 member vni 50001 associate-vrf
!
router bgp 65002
 vrf myvrf_50001
 router-id 10.2.0.6
 address-family ipv4 unicast
 advertise l2vpn evpn
 maximum-paths ibgp 2
 maximum-paths 2
 address-family ipv6 unicast
 advertise l2vpn evpn
 maximum-paths ibgp 2
 maximum-paths 2
 neighbor 10.31.95.95
 remote-as 65099
 address-family ipv4 unicast
 neighbor 2001::95/64
 remote-as 65099
 address-family ipv4 unicast
!
interface Ethernet1/3
 no switchport
 no shutdown
interface Ethernet1/3.2
 encapsulation dot1q 2
 vrf member myvrf_50001
 ip address 10.31.95.31/24
 ipv6 address 2001::31/64
 no shutdown

```

## Configuring Default-Route, Route Filtering on External Connectivity

The VXLAN BGP EVPN Border Node has the ability to advertise IPv4 and IPv6 default-route within the fabric. In cases where it is not beneficial to advertise the Host Routes from the VXLAN BGP EVPN fabric to the External Router, these IPv4 /32 and IPv6 /128 can be filtered at the External Connectivity peering configuration.

```

ip prefix-list default-route seq 5 permit 0.0.0.0/0 le 1
ipv6 prefix-list default-route-v6 seq 5 permit 0::/0
!
ip prefix-list host-route seq 5 permit 0.0.0.0/0 eq 32

```

```

ipv6 prefix-list host-route-v6 seq 5 permit 0::/0 eq 128
!
route-map extcon-rmap-filter deny 10
 match ip address prefix-list default-route
route-map extcon-rmap-filter deny 20
 match ip address prefix-list host-route
route-map extcon-rmap-filter permit 1000
!
route-map extcon-rmap-filter-v6 deny 10
 match ipv6 address prefix-list default-route-v6
route-map extcon-rmap-filter-v6 deny 20
 match ip address prefix-list host-route-v6
route-map extcon-rmap-filter-v6 permit 1000
!
vrf context myvrf_50001
 ip route 0.0.0.0/0 10.31.95.95
 ipv6 route 0::/0 2001::95/64
!
router bgp 65002
 vrf myvrf_50001
 address-family ipv4 unicast
 network 0.0.0.0/0
 address-family ipv6 unicast
 network 0::/0

 neighbor 10.31.95.95
 remote-as 65099
 address-family ipv4 unicast
 route-map extcon-rmap-filter out
 neighbor 2001::95/64
 remote-as 65099
 address-family ipv4 unicast
 route-map extcon-rmap-filter-v6 out

```

## Configuring External Router

The External Router performs as a neighbor device to the VXLAN BGP EVPN border node. The VRF Name is purely localized and can be different to the VRF Name on the VXLAN BGP EVPN Fabric. For the ease of reading, the VRF and interface enumeration will be consistently used.

The configuration examples represents a IPv4 and IPv6 dual-stack approach; IPv4 or IPv6 can be substituted of each other.

```

vrf context myvrf_50001
!
router bgp 65099
 vrf myvrf_50001
 address-family ipv4 unicast
 maximum-paths 2
 address-family ipv6 unicast
 maximum-paths 2
 neighbor 10.31.95.31
 remote-as 65002
 address-family ipv4 unicast
 neighbor 2001::31/64
 remote-as 65002
 address-family ipv4 unicast
!
interface Ethernet1/3
 no switchport
 no shutdown
interface Ethernet1/3.2
 encapsulation dot1q 2

```

```

vrf member myvrf_50001
ip address 10.31.95.95/24
Ipv6 address 2001::95/64
no shutdown

```

## Configuring VXLAN BGP EVPN with OSPF for VRF-lite

### Configuring VRF for VXLAN Routing and External Connectivity using OSPF

Configure the BGP VRF instance on the border node for OSPF per-VRF peering.

#### Procedure

|               | Command or Action                                                 | Purpose                                                            |
|---------------|-------------------------------------------------------------------|--------------------------------------------------------------------|
| <b>Step 1</b> | <b>configure terminal</b>                                         | Enter global configuration mode.                                   |
| <b>Step 2</b> | <b>router bgp</b> <i>autonomous-system-number</i>                 | Configure BGP.                                                     |
| <b>Step 3</b> | <b>vrf</b> <i>vrf-name</i>                                        | Specify the VRF.                                                   |
| <b>Step 4</b> | <b>address-family ipv4 unicast</b>                                | Configure the IPv4 address family.                                 |
| <b>Step 5</b> | <b>advertise l2vpn evpn</b>                                       | Enable the advertisement of EVPN routes within the address family. |
| <b>Step 6</b> | <b>maximum-paths ibgp</b> <i>number</i>                           | Enabling equal-cost multipathing (ECMP) for iBGP prefixes.         |
| <b>Step 7</b> | <b>redistribute ospf</b> <i>name</i> <b>route-map</b> <i>name</i> | Define redistribution from OSPF into BGP.                          |
| <b>Step 8</b> | Repeat Step 3 through Step 7 for every per-VRF peering.           |                                                                    |

### Configuring the Route-Map for BGP to OSPF Redistribution

#### Procedure

|               | Command or Action                             | Purpose                                                                                                                    |
|---------------|-----------------------------------------------|----------------------------------------------------------------------------------------------------------------------------|
| <b>Step 1</b> | <b>configure terminal</b>                     | Enter global configuration mode.                                                                                           |
| <b>Step 2</b> | <b>route-map</b> <i>name</i> <b>permit 10</b> | Create route-map for BGP to OSPF redistribution                                                                            |
| <b>Step 3</b> | <b>match route-type internal</b>              | Redistribution route-map must allow the matching of BGP internal route-types if iBGP is used in the VXLAN BGP EVPN fabric. |

## Configuring the OSPF on the Border Node for Per-VRF Peering

### Procedure

|               | Command or Action                                                                       | Purpose                                 |
|---------------|-----------------------------------------------------------------------------------------|-----------------------------------------|
| <b>Step 1</b> | <b>configure terminal</b>                                                               | Enter global configuration mode.        |
| <b>Step 2</b> | <b>router ospf</b> <i>instance</i>                                                      | Configure OSPF.                         |
| <b>Step 3</b> | <b>vrf</b> <i>vrf-name</i>                                                              | Specify the VRF.                        |
| <b>Step 4</b> | <b>redistribute bgp</b> <i>autonomous-system-number</i><br><b>route-map</b> <i>name</i> | Define redistribution from BGP to OSPF. |
| <b>Step 5</b> | Repeat Step 3 through Step 4 for every per-VRF peering.                                 |                                         |

## Configuring the Sub-Interface Instance on the Border Node for Per-VRF Peering - Version 2

### Procedure

|                | Command or Action                                        | Purpose                                                                   |
|----------------|----------------------------------------------------------|---------------------------------------------------------------------------|
| <b>Step 1</b>  | <b>configure terminal</b>                                | Enters global configuration mode.                                         |
| <b>Step 2</b>  | <b>interface</b> <i>type/id</i>                          | Configure parent interface.                                               |
| <b>Step 3</b>  | <b>no switchport</b>                                     | Disable Layer-2 switching mode on interface.                              |
| <b>Step 4</b>  | <b>no shutdown</b>                                       | Bring up parent interface.                                                |
| <b>Step 5</b>  | <b>exit</b>                                              | Exit interface configuration mode.                                        |
| <b>Step 6</b>  | <b>interface</b> <i>type/id</i>                          | Define the Sub-Interface instance.                                        |
| <b>Step 7</b>  | <b>encapsulation dot1q</b> <i>number</i>                 | Configure the VLAN ID for the sub-interface. The range is from 2 to 4093. |
| <b>Step 8</b>  | <b>vrf member</b> <i>vrf-name</i>                        | Map the Sub-Interface to the matching VRF context.                        |
| <b>Step 9</b>  | <b>ip address</b> <i>address</i>                         | Configure the Sub-Interfaces IP address.                                  |
| <b>Step 10</b> | <b>ip ospf network point-to-point</b>                    | Define OSPF network-type for sub-interface.                               |
| <b>Step 11</b> | <b>ip router ospf</b> <i>name area area-id</i>           | Configure the OSPF instance.                                              |
| <b>Step 12</b> | <b>no shutdown</b>                                       | Bring up Sub-Interface.                                                   |
| <b>Step 13</b> | Repeat Step 5 through Step 12 for every per-VRF peering. |                                                                           |

### Example - Configuration VXLAN BGP EVPN with OSPF for VRF-lite

An example of external connectivity from VXLAN BGP EVPN to an External Router using VRF-lite.

### Configuring VXLAN BGP EVPN Border Node with OSPF

The VXLAN BGP EVPN Border Node acts as neighbor device to the External Router. The VRF Name is purely localized and can be different to the VRF Name on the External Router, only significance is the L3VNI must be consistent across the VXLAN BGP EVPN fabric. For the ease of reading, the VRF and interface enumeration will be consistently used.

The configuration examples represents a IPv4 approach with OSPFv2.

```
route-map extcon-rmap-BGP-to-OSPF permit 10
 match route-type internal
route-map extcon-rmap-OSPF-to-BGP permit 10
!
vrf context myvrf_50001
 vni 50001
 rd auto
 address-family ipv4 unicast
 route-target both auto
 route-target both auto evpn
!
vlan 2000
 vn-segment 50001
!
interface Vlan2000
 no shutdown
 mtu 9216
 vrf member myvrf_50001
 no ip redirects
 ip forward
!
interface nve1
 no shutdown
 host-reachability protocol bgp
 source-interface loopback1
 member vni 50001 associate-vrf
!
router bgp 65002
 vrf myvrf_50001
 router-id 10.2.0.6
 address-family ipv4 unicast
 advertise l2vpn evpn
 maximum-paths ibgp 2
 maximum-paths 2
 redistribute ospf EXT route-map extcon-rmap-OSPF-to-BGP
!
router ospf EXT
 vrf myvrf_50001
 redistribute bgp 65002 route-map extcon-rmap-BGP-to-OSPF
!
interface Ethernet1/3
 no switchport
 no shutdown
interface Ethernet1/3.2
 encapsulation dot1q 2
 vrf member myvrf_50001
 ip address 10.31.95.31/24
 ip ospf network point-to-point
 ip router ospf EXT area 0.0.0.0
 no shutdown
```

# Configuring Route Leaking

## About Centralized VRF Route-Leaking for VXLAN BGP EVPN Fabrics

VXLAN BGP EVPN uses MP-BGP and its route-policy concept to import and export prefixes. The ability of this very extensive route-policy model allows to leak routes from one VRF to another VRF and vice-versa; any combination of custom VRF or VRF default can be used. VRF route-leaking is a switch-local function at specific to a location in the network, the location where the cross-VRF route-target import/export configuration takes place (leaking point). The forwarding between the different VRFs follows the control-plane, the location of where the configuration for the route-leaking is performed - hence Centralized VRF route-leaking. With the addition of VXLAN BGP EVPN, the leaking point requires to advertise the cross-VRF imported/exported route and advertise them towards the remote VTEPs or External Routers.

The advantage of Centralized VRF route-leaking is that only the VTEP acting as leaking point requires the special capabilities needed, while all other VTEPs in the network are neutral to this function.

## Guidelines and Limitations for Centralized VRF Route-Leaking

The following are the guidelines and limitations for Centralized VRF Route-Leaking:

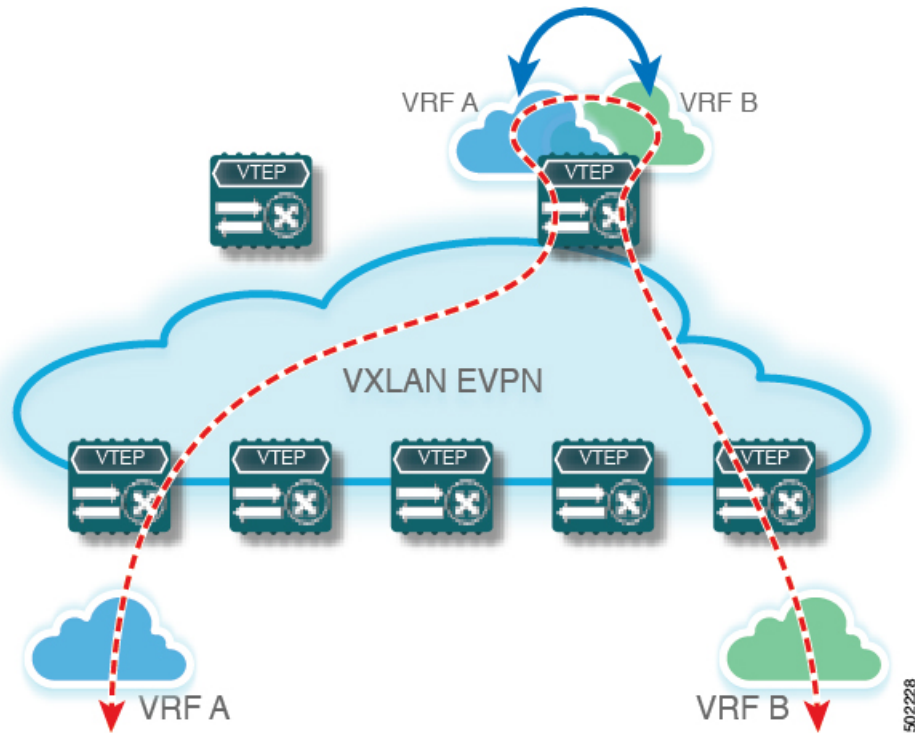
- Each prefix must be imported into each VRF for full cross-VRF reachability.
- The **feature bgp** command is required for the **export vrf default** command.
- If a VTEP has a less specific local prefix in its VRF, the VTEP might not be able to reach a more specific prefix in a different VRF.
- VXLAN routing in hardware and packet reencapsulation at VTEP is required for Centralized VRF Route-Leaking with BGP EVPN.

## Centralized VRF Route-Leaking Brief - Specific Prefixes Between Custom VRF

Some pointers are given below:

- The Centralized VRF route-leaking for VXLAN BGP EVPN fabrics is depicted within Figure 2.
- BGP EVPN prefixes are cross-VRF leaked by exporting them from VRF Blue with an import into VRF Red and vice-versa. The Centralized VRF route-leaking is performed on the centralized Routing-Block (RBL) and could be any or multiple VTEPs.
- Configured less specific prefixes (aggregates) are advertised from the Routing-Block to the remaining VTEPs in the respective destination VRF.
- BGP EVPN does not export prefixes that were previously imported to prevent the occurrence of routing loops.

Figure 13: Centralized VRF Route-Leaking - Specific Prefixes with Custom VRF



## Configuring Centralized VRF Route-Leaking - Specific Prefixes between Custom VRF

### Configuring VRF Context on the Routing-Block VTEP

This procedure applies equally to IPv6.

#### Procedure

|               | Command or Action                  | Purpose                                                                                                                                                                                      |
|---------------|------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 1</b> | <b>configure terminal</b>          | Enter global configuration mode.                                                                                                                                                             |
| <b>Step 2</b> | <b>vrf context</b> <i>vrf-name</i> | Configure the VRF.                                                                                                                                                                           |
| <b>Step 3</b> | <b>vni</b> <i>number</i>           | Specify the VNI.<br><br>The VNI associated with the VRF is often referred to as Layer 3 VNI, L3VNI, or L3VPN. The L3VNI is configured as a common identifier across the participating VTEPs. |
| <b>Step 4</b> | <b>rd auto</b>                     | Specify the VRF's route distinguisher (RD). The RD uniquely identifies a VTEP within an L3VNI.                                                                                               |



|               | Command or Action                                     | Purpose                                                                                                                                                                                                                  |
|---------------|-------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 5</b> | <b>address-family ipv4 unicast</b>                    | Configure the IPv4 unicast address family.                                                                                                                                                                               |
| <b>Step 6</b> | <b>route-target both {auto   rt}</b>                  | Configure the route target (RT) for import and export of IPv4 prefixes. The RT is used for a per-VRF prefix import/export policy. If you enter an RT, the following formats are supported: ASN2:NN, ASN4:NN, or IPV4:NN. |
| <b>Step 7</b> | <b>route-target both {auto   rt} evpn</b>             | Configure the route target (RT) for import and export of IPv4 prefixes. The RT is used for a per-VRF prefix import/export policy. If you enter an RT, the following formats are supported: ASN2:NN, ASN4:NN, or IPV4:NN. |
| <b>Step 8</b> | <b>route-target import rt-from-different-vrf</b>      | Configure the RT for importing IPv4 prefixes from the leaked-from VRF. The following formats are supported: ASN2:NN, ASN4:NN, or IPV4:NN.                                                                                |
| <b>Step 9</b> | <b>route-target import rt-from-different-vrf evpn</b> | Configure the RT for importing IPv4 prefixes from the leaked-from VRF. The following formats are supported: ASN2:NN, ASN4:NN, or IPV4:NN.                                                                                |

## Configuring the BGP VRF instance on the Routing-Block

This procedure applies equally to IPv6.

### Procedure

|               | Command or Action                                 | Purpose                                                             |
|---------------|---------------------------------------------------|---------------------------------------------------------------------|
| <b>Step 1</b> | <b>configure terminal</b>                         | Enters global configuration mode.                                   |
| <b>Step 2</b> | <b>router bgp <i>autonomous-system number</i></b> | Configure BGP.                                                      |
| <b>Step 3</b> | <b>vrf <i>vrf-name</i></b>                        | Specify the VRF.                                                    |
| <b>Step 4</b> | <b>address-family ipv4 unicast</b>                | Configure address family for IPv4                                   |
| <b>Step 5</b> | <b>advertise l2vpn evpn</b>                       | Enable the advertisement of EVPN routes within IPv4 address-family. |
| <b>Step 6</b> | <b>aggregate-address <i>prefix/mask</i></b>       | Create less specific prefix aggregate into the destination VRF.     |
| <b>Step 7</b> | <b>maximum-paths ibgp <i>number</i></b>           | Enabling equal cost multipathing (ECMP) for iBGP prefixes.          |
| <b>Step 8</b> | <b>maximum-paths <i>number</i></b>                | Enabling equal cost multipathing (ECMP) for eBGP prefixes           |

## Example - Configuration Centralized VRF Route-Leaking - Specific Prefixes Between Custom VRF

### Configuring VXLAN BGP EVPN Routing-Block

The VXLAN BGP EVPN Routing-Block acts as centralized route-leaking point. The leaking configuration is localized such that control-plane leaking and data-path forwarding follow the same path. Most significantly is the VRF configuration of the Routing-Block and the advertisement of the less specific prefixes (aggregates) into the respective destination VRFs.

```
vrf context Blue
 vni 51010
 rd auto
 address-family ipv4 unicast
 route-target both auto
 route-target both auto evpn
 route-target import 65002:51020
 route-target import 65002:51020 evpn
!
vlan 2110
 vn-segment 51010
!
interface Vlan2110
 no shutdown
 mtu 9216
 vrf member Blue
 no ip redirects
 ip forward
!
vrf context Red
 vni 51020
 rd auto
 address-family ipv4 unicast
 route-target both auto
 route-target both auto evpn
 route-target import 65002:51010
 route-target import 65002:51010 evpn
!
vlan 2120
 vn-segment 51020
!
interface Vlan2120
 no shutdown
 mtu 9216
 vrf member Blue
 no ip redirects
 ip forward
!
interface nve1
 no shutdown
 host-reachability protocol bgp
 source-interface loopback1
 member vni 51010 associate-vrf
 member vni 51020 associate-vrf
!
router bgp 65002
 vrf Blue
 address-family ipv4 unicast
 advertise l2vpn evpn
 aggregate-address 10.20.0.0/16
 maximum-paths ibgp 2
 Maximum-paths 2
 vrf Red
 address-family ipv4 unicast
```

```

advertise l2vpn evpn
aggregate-address 10.10.0.0/16
maximum-paths ibgp 2
Maximum-paths 2

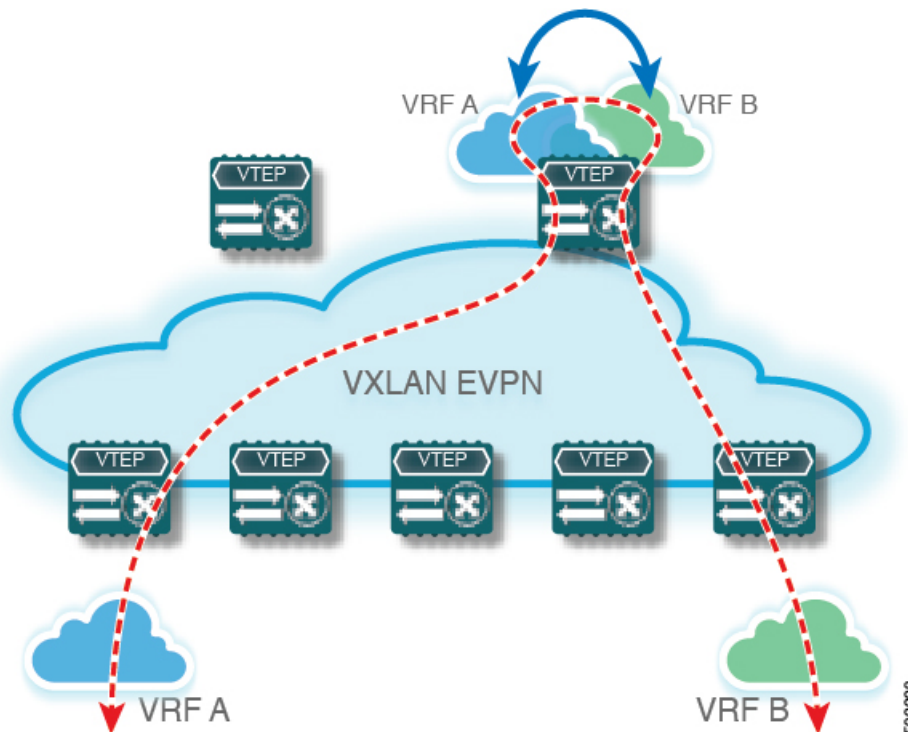
```

## Centralized VRF Route-Leaking Brief - Shared Internet with Custom VRF

Some pointers follow:

- The Shared Internet with VRF route-leaking for VXLAN BGP EVPN fabrics is depicted in the following figure.
- The default-route is made exported from the Shared Internet VRF and re-advertisement within VRF Blue and VRF Red on the Border Node.
- Ensure the default-route in VRF Blue and VRF Red is not leaked to the Shared Internet VRF.
- The less specific prefixes for VRF Blue and VRF Red are exported for the Shared Internet VRF and re-advertised as necessary.
- Configured less specific prefixes (aggregates) that are advertised from the Border Node to the remaining VTEPs to the destination VRF (Blue or Red).
- BGP EVPN does not export prefixes that were previously imported to prevent the occurrence of routing loops.

**Figure 14: Centralized VRF Route-Leaking - Shared Internet with Custom VRF**



# Configuring Centralized VRF Route-Leaking - Shared Internet with Custom VRF

## Configuring Internet VRF on Border Node

This procedure applies equally to IPv6.

### Procedure

|               | Command or Action                                    | Purpose                                                                                                                                                                                      |
|---------------|------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 1</b> | <b>configure terminal</b>                            | Enter global configuration mode.                                                                                                                                                             |
| <b>Step 2</b> | <b>vrf context</b> <i>vrf-name</i>                   | Configure the VRF.                                                                                                                                                                           |
| <b>Step 3</b> | <b>vni</b> <i>number</i>                             | Specify the VNI.<br><br>The VNI associated with the VRF is often referred to as Layer 3 VNI, L3VNI, or L3VPN. The L3VNI is configured as a common identifier across the participating VTEPs. |
| <b>Step 4</b> | <b>ip route 0.0.0.0/0</b> <i>next-hop</i>            | Configure the default route in the shared internet VRF to the external router.                                                                                                               |
| <b>Step 5</b> | <b>rd auto</b>                                       | Specify the VRF's route distinguisher (RD). The RD uniquely identifies a VTEP within an L3VNI.                                                                                               |
| <b>Step 6</b> | <b>address-family ipv4 unicast</b>                   | Configure the IPv4 unicast address family. This configuration is required for IPv4 over VXLAN with IPv4 underlay.                                                                            |
| <b>Step 7</b> | <b>route-target both</b> { <i>auto</i>   <i>rt</i> } | Configure the route target (RT) for the import and export of EVPN and IPv4 prefixes. If you enter an RT, the following formats are supported: ASN2:NN, ASN4:NN, or IPV4:NN.                  |
| <b>Step 8</b> | <b>route-target both</b> <i>shared-vrf-rt evpn</i>   | Configure a special route target (RT) for the import and export of the shared IPv4 prefixes. An additional import/export map for further qualification is supported.                         |

## Configuring Shared Internet BGP Instance on the Border Node

This procedure applies equally to IPv6.

### Procedure

|               | Command or Action                                 | Purpose                           |
|---------------|---------------------------------------------------|-----------------------------------|
| <b>Step 1</b> | <b>configure terminal</b>                         | Enters global configuration mode. |
| <b>Step 2</b> | <b>router bgp</b> <i>autonomous-system number</i> | Configure BGP.                    |

|               | Command or Action                           | Purpose                                                             |
|---------------|---------------------------------------------|---------------------------------------------------------------------|
| <b>Step 3</b> | <b>vrf</b> <i>vrf-name</i>                  | Specify the VRF.                                                    |
| <b>Step 4</b> | <b>address-family ipv4 unicast</b>          | Configure address family for IPv4                                   |
| <b>Step 5</b> | <b>advertise l2vpn evpn</b>                 | Enable the advertisement of EVPN routes within IPv4 address-family. |
| <b>Step 6</b> | <b>aggregate-address</b> <i>prefix/mask</i> | Create less specific prefix aggregate into the destination VRF.     |
| <b>Step 7</b> | <b>maximum-paths ibgp</b> <i>number</i>     | Enabling equal cost multipathing (ECMP) for iBGP prefixes.          |
| <b>Step 8</b> | <b>maximum-paths</b> <i>number</i>          | Enabling equal cost multipathing (ECMP) for eBGP prefixes.          |

## Configuring Custom VRF on Border Node

This procedure applies equally to IPv6

### Procedure

|               | Command or Action                                               | Purpose                                                                                            |
|---------------|-----------------------------------------------------------------|----------------------------------------------------------------------------------------------------|
| <b>Step 1</b> | <b>configure terminal</b>                                       | Enters global configuration mode.                                                                  |
| <b>Step 2</b> | <b>ip prefix-list</b> <i>name</i> <b>seq 5 permit 0.0.0.0/0</b> | Configure IPv4 prefix-list for default-route filtering.                                            |
| <b>Step 3</b> | <b>route-map</b> <i>name</i> <b>deny 10</b>                     | Create route-map with leading deny statement to prevent the default-route of being leaked.         |
| <b>Step 4</b> | <b>match ip address prefix-list</b> <i>name</i>                 | Match against the IPv4 prefix-list that contains the default-route.                                |
| <b>Step 5</b> | <b>route-map</b> <i>name</i> <b>permit 20</b>                   | Create route-map with trailing allow statement to advertise non-matching routes via route-leaking. |

## Configuring Custom VRF Context on the Border Node - 1

This procedure applies equally to IPv6.

### Procedure

|               | Command or Action                  | Purpose                           |
|---------------|------------------------------------|-----------------------------------|
| <b>Step 1</b> | <b>configure terminal</b>          | Enters global configuration mode. |
| <b>Step 2</b> | <b>vrf context</b> <i>vrf-name</i> | Configure the VRF.                |

|               | Command or Action                         | Purpose                                                                                                                                                                                                                                                     |
|---------------|-------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 3</b> | <b>vni <i>number</i></b>                  | Specify the VNI. The VNI associated with the VRF is often referred to as Layer 3 VNI, L3VNI, or L3VPN. The L3VNI is configured as the common identifier across the participating VTEPs.                                                                     |
| <b>Step 4</b> | <b>rd auto</b>                            | Specify the VRF's route distinguisher (RD). The RD uniquely identifies a VTEP within an L3VNI.                                                                                                                                                              |
| <b>Step 5</b> | <b>ip route 0.0.0.0/0 Null0</b>           | Configure default-route in common VRF to attract traffic towards Border Node with Shared Internet VRF.                                                                                                                                                      |
| <b>Step 6</b> | <b>address-family ipv4 unicast</b>        | Configure the IPv4 address family. This configuration is required for IPv4 over VXLAN with IPv4 underlay.                                                                                                                                                   |
| <b>Step 7</b> | <b>route-target both {auto   rt}</b>      | Configure the route target (RT) for the import and export of IPv4 prefixes within the IPv4 address family. The RT is used for a per-VRF prefix import/export policy. If you enter an RT, the following formats are supported: ASN2:NN, ASN4:NN, or IPV4:NN. |
| <b>Step 8</b> | <b>route-target both {auto   rt} evpn</b> | Configure the route target (RT) for the import and export of IPv4 prefixes within the IPv4 address family. The RT is used for a per-VRF prefix import/export policy. If you enter an RT, the following formats are supported: ASN2:NN, ASN4:NN, or IPV4:NN. |
| <b>Step 9</b> | <b>import map <i>name</i></b>             | Apply a route-map on routes being imported into this routing table.                                                                                                                                                                                         |

## Configuring Custom VRF Instance in BGP on the Border Node

This procedure applies equally to IPv6.

### Procedure

|               | Command or Action                                 | Purpose                            |
|---------------|---------------------------------------------------|------------------------------------|
| <b>Step 1</b> | <b>configure terminal</b>                         | Enters global configuration mode.  |
| <b>Step 2</b> | <b>router bgp <i>autonomous-system-number</i></b> | Configure BGP.                     |
| <b>Step 3</b> | <b>vrf <i>vrf-name</i></b>                        | Specify the VRF.                   |
| <b>Step 4</b> | <b>address-family ipv4 unicast</b>                | Configure address family for IPv4. |

|               | Command or Action                       | Purpose                                                             |
|---------------|-----------------------------------------|---------------------------------------------------------------------|
| <b>Step 5</b> | <b>advertise l2vpn evpn</b>             | Enable the advertisement of EVPN routes within IPv4 address-family. |
| <b>Step 6</b> | <b>network 0.0.0.0/0</b>                | Creating IPv4 default-route network statement.                      |
| <b>Step 7</b> | <b>maximum-paths ibgp <i>number</i></b> | Enabling equal cost multipathing (ECMP) for iBGP prefixes.          |
| <b>Step 8</b> | <b>maximum-paths <i>number</i></b>      | Enabling equal cost multipathing (ECMP) for eBGP prefixes.          |

## Example - Configuration Centralized VRF Route-Leaking - Shared Internet with Custom VRF

An example of Centralized VRF route-leaking with Shared Internet VRF

### Configuring VXLAN BGP EVPN Border Node for Shared Internet VRF

The VXLAN BGP EVPN Border Node provides a centralized Shared Internet VRF. The leaking configuration is localized such that control-plane leaking and data-path forwarding following the same path. Most significantly is the VRF configuration of the Border Node and the advertisement of the default-route and less specific prefixes (aggregates) into the respective destination VRFs.

```
vrf context Shared
 vni 51099
 ip route 0.0.0.0/0 10.9.9.1
 rd auto
 address-family ipv4 unicast
 route-target both auto
 route-target both auto evpn
 route-target both 99:99
 route-target both 99:99 evpn
 !
vlan 2199
 vn-segment 51099
 !
interface Vlan2199
 no shutdown
 mtu 9216
 vrf member Shared
 no ip redirects
 ip forward
 !
ip prefix-list PL_DENY_EXPORT seq 5 permit 0.0.0.0/0
 !
route-map RM_DENY_IMPORT deny 10
 match ip address prefix-list PL_DENY_EXPORT
route-map RM_DENY_IMPORT permit 20
 !
vrf context Blue
 vni 51010
 ip route 0.0.0.0/0 Null0
 rd auto
 address-family ipv4 unicast
 route-target both auto
 route-target both auto evpn
 route-target both 99:99
 route-target both 99:99 evpn
 import map RM_DENY_IMPORT
```

## Example - Configuration Centralized VRF Route-Leaking - Shared Internet with Custom VRF

```

!
vlan 2110
 vn-segment 51010
!
interface Vlan2110
 no shutdown
 mtu 9216
 vrf member Blue
 no ip redirects
 ip forward
!
vrf context Red
 vni 51020
 ip route 0.0.0.0/0 Null0
 rd auto
 address-family ipv4 unicast
 route-target both auto
 route-target both auto evpn
 route-target both 99:99
 route-target both 99:99 evpn
 import map RM_DENY_IMPORT
!
vlan 2120
 vn-segment 51020
!
interface Vlan2120
 no shutdown
 mtu 9216
 vrf member Blue
 no ip redirects
 ip forward
!
interface nve1
 no shutdown
 host-reachability protocol bgp
 source-interface loopback1
 member vni 51099 associate-vrf
 member vni 51010 associate-vrf
 member vni 51020 associate-vrf
!
router bgp 65002
 vrf Shared
 address-family ipv4 unicast
 advertise l2vpn evpn
 aggregate-address 10.10.0.0/16
 aggregate-address 10.20.0.0/16
 maximum-paths ibgp 2
 maximum-paths 2
 vrf Blue
 address-family ipv4 unicast
 advertise l2vpn evpn
 network 0.0.0.0/0
 maximum-paths ibgp 2
 maximum-paths 2
 vrf Red
 address-family ipv4 unicast
 advertise l2vpn evpn
 network 0.0.0.0/0
 maximum-paths ibgp 2
 maximum-paths 2

```

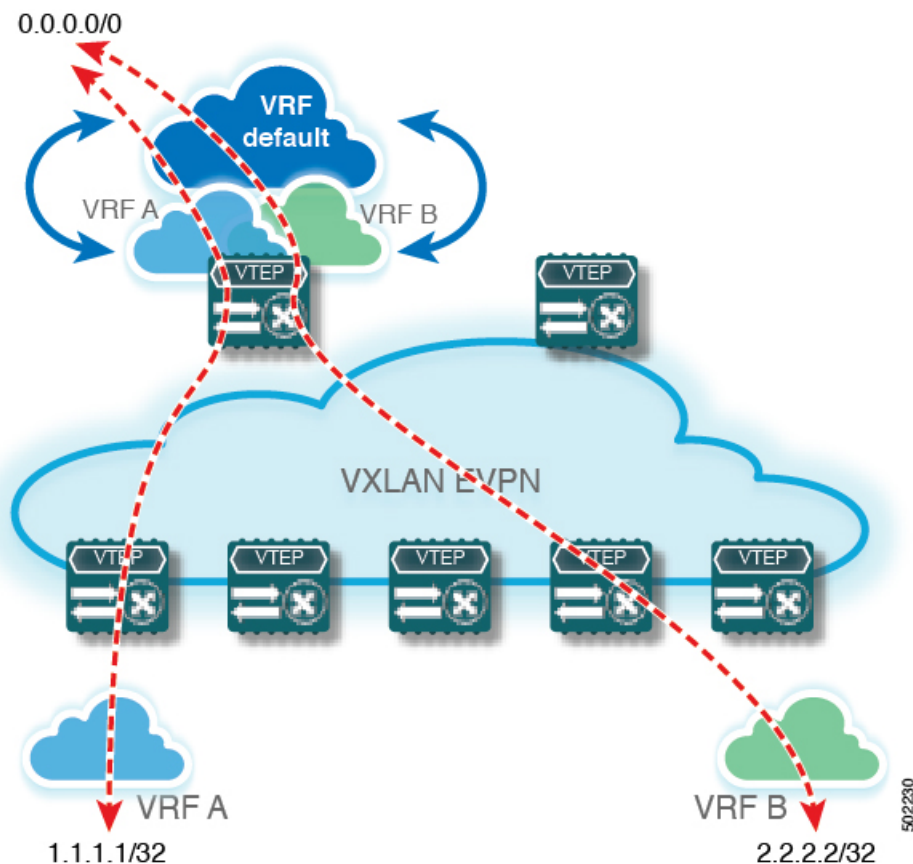


## Centralized VRF Route-Leaking Brief - Shared Internet with VRF Default

Some pointers are given below:

- The Shared Internet with VRF route-leaking for VXLAN BGP EVPN fabrics is depicted within Figure 4.
- The default-route is made exported from VRF default and re-advertisement within VRF Blue and VRF Red on the Border Node.
- Ensure the default-route in VRF Blue and VRF Red is not leaked to the Shared Internet VRF
- The less specific prefixes for VRF Blue and VRF Red are exported to VRF default and re-advertised as necessary.
- Configured less specific prefixes (aggregates) that are advertised from the Border Node to the remaining VTEPs to the destination VRF (Blue or Red).
- BGP EVPN does not export prefixes that were previously imported to prevent the occurrence of routing loops.

**Figure 15: Centralized VRF Route-Leaking - Shared Internet with VRF Default**



## Configuring Centralized VRF Route-Leaking - Shared Internet with VRF Default

### Configuring VRF Default on Border Node

This procedure applies equally to IPv6.

#### Procedure

|               | Command or Action                         | Purpose                                                             |
|---------------|-------------------------------------------|---------------------------------------------------------------------|
| <b>Step 1</b> | <b>configure terminal</b>                 | Enters global configuration mode.                                   |
| <b>Step 2</b> | <b>ip route 0.0.0.0/0 <i>next-hop</i></b> | Configure default-route in VRF default to external router (example) |

### Configuring BGP Instance for VRF Default on the Border Node

This procedure applies equally to IPv6.

#### Procedure

|               | Command or Action                                 | Purpose                                                    |
|---------------|---------------------------------------------------|------------------------------------------------------------|
| <b>Step 1</b> | <b>configure terminal</b>                         | Enters global configuration mode.                          |
| <b>Step 2</b> | <b>router bgp <i>autonomous-system number</i></b> | Configure BGP.                                             |
| <b>Step 3</b> | <b>address-family ipv4 unicast</b>                | Configure address family for IPv4.                         |
| <b>Step 4</b> | <b>aggregate-address <i>prefix/mask</i></b>       | Create less specific prefix aggregate in VRF default.      |
| <b>Step 5</b> | <b>maximum-paths <i>number</i></b>                | Enabling equal cost multipathing (ECMP) for eBGP prefixes. |

### Configuring Custom VRF on Border Node

This procedure applies equally to IPv6

#### Procedure

|               | Command or Action                                        | Purpose                                                                                    |
|---------------|----------------------------------------------------------|--------------------------------------------------------------------------------------------|
| <b>Step 1</b> | <b>configure terminal</b>                                | Enters global configuration mode.                                                          |
| <b>Step 2</b> | <b>ip prefix-list <i>name</i> seq 5 permit 0.0.0.0/0</b> | Configure IPv4 prefix-list for default-route filtering.                                    |
| <b>Step 3</b> | <b>route-map <i>name</i> deny 10</b>                     | Create route-map with leading deny statement to prevent the default-route of being leaked. |
| <b>Step 4</b> | <b>match ip address prefix-list <i>name</i></b>          | Match against the IPv4 prefix-list that contains the default-route.                        |

|               | Command or Action                      | Purpose                                                                                            |
|---------------|----------------------------------------|----------------------------------------------------------------------------------------------------|
| <b>Step 5</b> | <b>route-map <i>name</i> permit 20</b> | Create route-map with trailing allow statement to advertise non-matching routes via route-leaking. |

## Configuring Filter for Permitted Prefixes from VRF Default on the Border Node

This procedure applies equally to IPv6.

### Procedure

|               | Command or Action                      | Purpose                                                                                                                        |
|---------------|----------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 1</b> | <b>configure terminal</b>              | Enters global configuration mode.                                                                                              |
| <b>Step 2</b> | <b>route-map <i>name</i> permit 10</b> | Create route-map with allow statement to advertise routes via route-leaking to the customer VRF and subsequently remote VTEPs. |

## Configuring Custom VRF Context on the Border Node - 2

This procedure applies equally to IPv6.

### Procedure

|               | Command or Action                                  | Purpose                                                                                                                                                                                 |
|---------------|----------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 1</b> | <b>configure terminal</b>                          | Enter global configuration mode.                                                                                                                                                        |
| <b>Step 2</b> | <b>vrf context <i>vrf-name</i></b>                 | Configure the VRF.                                                                                                                                                                      |
| <b>Step 3</b> | <b>vni <i>number</i></b>                           | Specify the VNI. The VNI associated with the VRF is often referred to as Layer 3 VNI, L3VNI, or L3VPN. The L3VNI is configured as the common identifier across the participating VTEPs. |
| <b>Step 4</b> | <b>rd auto</b>                                     | Specify the VRF's route distinguisher (RD). The RD uniquely identifies a VTEP within an L3VNI.                                                                                          |
| <b>Step 5</b> | <b>ip route 0.0.0.0/0 Null0</b>                    | Configure default-route in common VRF to attract traffic towards Border Node with Shared Internet VRF.                                                                                  |
| <b>Step 6</b> | <b>address-family ipv4 unicast</b>                 | Configure the IPv4 address family. This configuration is required for IPv4 over VXLAN with IPv4 underlay.                                                                               |
| <b>Step 7</b> | <b>route-target both {<i>auto</i>   <i>rt</i>}</b> | Configure the route target (RT) for the import and export of EVPN and IPv4 prefixes within the IPv4 address family. If you enter an RT,                                                 |

|                | Command or Action                                  | Purpose                                                                                                                                                                                                    |
|----------------|----------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                |                                                    | the following formats are supported: ASN2:NN, ASN4:NN, or IPV4:NN.                                                                                                                                         |
| <b>Step 8</b>  | <b>route-target both {auto   rt} evpn</b>          | Configure the route target (RT) for the import and export of EVPN and IPv4 prefixes within the IPv4 address family. If you enter an RT, the following formats are supported: ASN2:NN, ASN4:NN, or IPV4:NN. |
| <b>Step 9</b>  | <b>route-target both <i>shared-vrf-rt</i></b>      | Configure a special route target (RT) for the import/export of the shared IPv4 prefixes. An additional import/export map for further qualification is supported.                                           |
| <b>Step 10</b> | <b>route-target both <i>shared-vrf-rt</i> evpn</b> | Configure a special route target (RT) for the import/export of the shared IPv4 prefixes. An additional import/export map for further qualification is supported.                                           |
| <b>Step 11</b> | <b>import vrf default map <i>name</i></b>          | Permits all routes, from VRF default, from being imported into the custom VRF according to the specific route-map.                                                                                         |

## Configuring Custom VRF Instance in BGP on the Border Node

This procedure applies equally to IPv6.

### Procedure

|               | Command or Action                                 | Purpose                                                             |
|---------------|---------------------------------------------------|---------------------------------------------------------------------|
| <b>Step 1</b> | <b>configure terminal</b>                         | Enters global configuration mode.                                   |
| <b>Step 2</b> | <b>router bgp <i>autonomous-system-number</i></b> | Configure BGP.                                                      |
| <b>Step 3</b> | <b>vrf <i>vrf-name</i></b>                        | Specify the VRF.                                                    |
| <b>Step 4</b> | <b>address-family ipv4 unicast</b>                | Configure address family for IPv4.                                  |
| <b>Step 5</b> | <b>advertise l2vpn evpn</b>                       | Enable the advertisement of EVPN routes within IPv4 address-family. |
| <b>Step 6</b> | <b>network 0.0.0.0/0</b>                          | Creating IPv4 default-route network statement.                      |
| <b>Step 7</b> | <b>maximum-paths ibgp <i>number</i></b>           | Enabling equal cost multipathing (ECMP) for iBGP prefixes.          |
| <b>Step 8</b> | <b>maximum-paths <i>number</i></b>                | Enabling equal cost multipathing (ECMP) for eBGP prefixes.          |

## Example - Configuration Centralized VRF Route-Leaking - VRF Default with Custom VRF

An example of Centralized VRF route-leaking with VRF default

### Configuring VXLAN BGP EVPN Border Node for VRF Default

The VXLAN BGP EVPN Border Node provides centralized access to VRF default. The leaking configuration is localized such that control-plane leaking and data-path forwarding following the same path. Most significantly is the VRF configuration of the Border Node and the advertisement of the default-route and less specific prefixes (aggregates) into the respective destination VRFs.

```
ip route 0.0.0.0/0 10.9.9.1
!
ip prefix-list PL_DENY_EXPORT seq 5 permit 0.0.0.0/0
!
route-map permit 10
match ip address prefix-list PL_DENY_EXPORT
route-map RM_DENY_EXPORT permit 20
route-map RM_PERMIT_IMPORT permit 10
!
vrf context Blue
 vni 51010
 ip route 0.0.0.0/0 Null0
 rd auto
 address-family ipv4 unicast
 route-target both auto
 route-target both auto evpn
 import vrf default map RM_PERMIT_IMPORT
 export vrf default 100 map RM_DENY_EXPORT allow-vpn
!
vlan 2110
 vn-segment 51010
!
interface Vlan2110
 no shutdown
 mtu 9216
 vrf member Blue
 no ip redirects
 ip forward
!
vrf context Red
 vni 51020
 ip route 0.0.0.0/0 Null0
 rd auto
 address-family ipv4 unicast
 route-target both auto
 route-target both auto evpn
 import vrf default map RM_PERMIT_IMPORT
 export vrf default 100 map RM_DENY_EXPORT allow-vpn
!
vlan 2120
 vn-segment 51020
!
interface Vlan2120
 no shutdown
 mtu 9216
 vrf member Blue
 no ip redirects
 ip forward
!
interface nve1
 no shutdown
 host-reachability protocol bgp
```

**Example - Configuration Centralized VRF Route-Leaking - VRF Default with Custom VRF**

```
source-interface loopback1
member vni 51010 associate-vrf
member vni 51020 associate-vrf
!
router bgp 65002
 address-family ipv4 unicast
 aggregate-address 10.10.0.0/16
 aggregate-address 10.20.0.0/16
 maximum-paths 2
 maximum-paths ibgp 2
 vrf Blue
 address-family ipv4 unicast
 advertise l2vpn evpn
 network 0.0.0.0/0
 maximum-paths ibgp 2
 maximum-paths 2
 vrf Red
 address-family ipv4 unicast
 advertise l2vpn evpn
 network 0.0.0.0/0
 maximum-paths ibgp 2
 maximum-paths 2
```



## CHAPTER 7

# Configuring VXLAN OAM

This chapter contains the following sections:

- [VXLAN OAM Overview, on page 139](#)
- [Guidelines and Limitations for VXLAN NGOAM, on page 142](#)
- [Configuring VXLAN OAM, on page 142](#)
- [Configuring NGOAM Profile, on page 145](#)

## VXLAN OAM Overview

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

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

There are two types of payloads supported:

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

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

**Table 4: VXLAN OAM Tools**

| Category           | Tools              |
|--------------------|--------------------|
| Fault Verification | Loopback Message   |
| Fault Isolation    | Path Trace Message |

| Category    | Tools                                                                                                                                              |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| 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 16: Loopback Message**

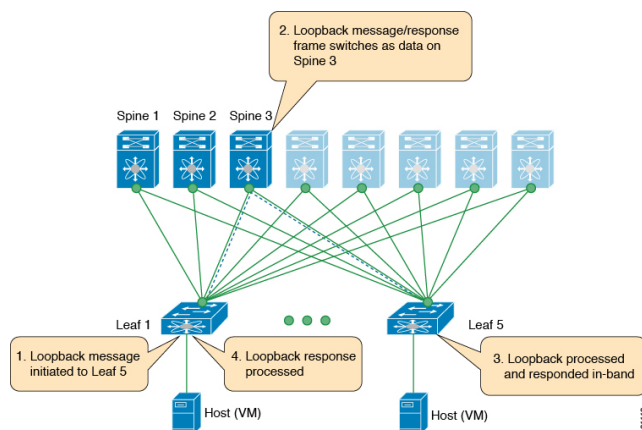
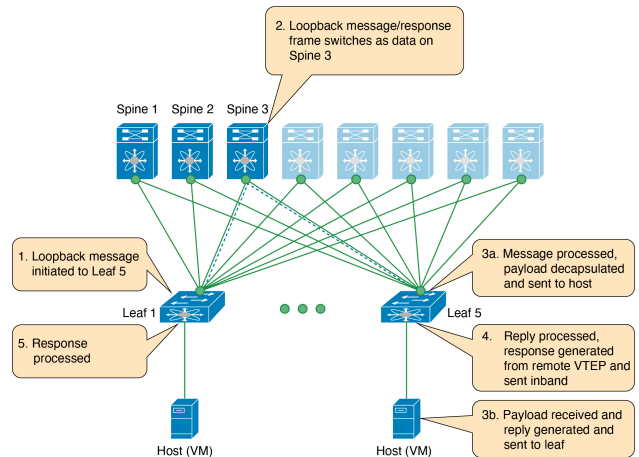




Figure 17: NVO3 Draft Tissa Ping to Remote VM



## Traceroute or Pathtrace Message

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

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

The traceroute and pathtrace messages are similar, except that traceroute uses the ICMP channel, whereas pathtrace use the NVO3 draft Tissa channel. Pathtrace uses the NVO3 draft Tissa channel, carrying additional diagnostic information, for example, interface load and statistics of the hops taken by these messages. If an intermediate device does not support the NVO3 draft Tissa channel, the pathtrace behaves as a simple traceroute and it provides only the hop information.

### Traceroute

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

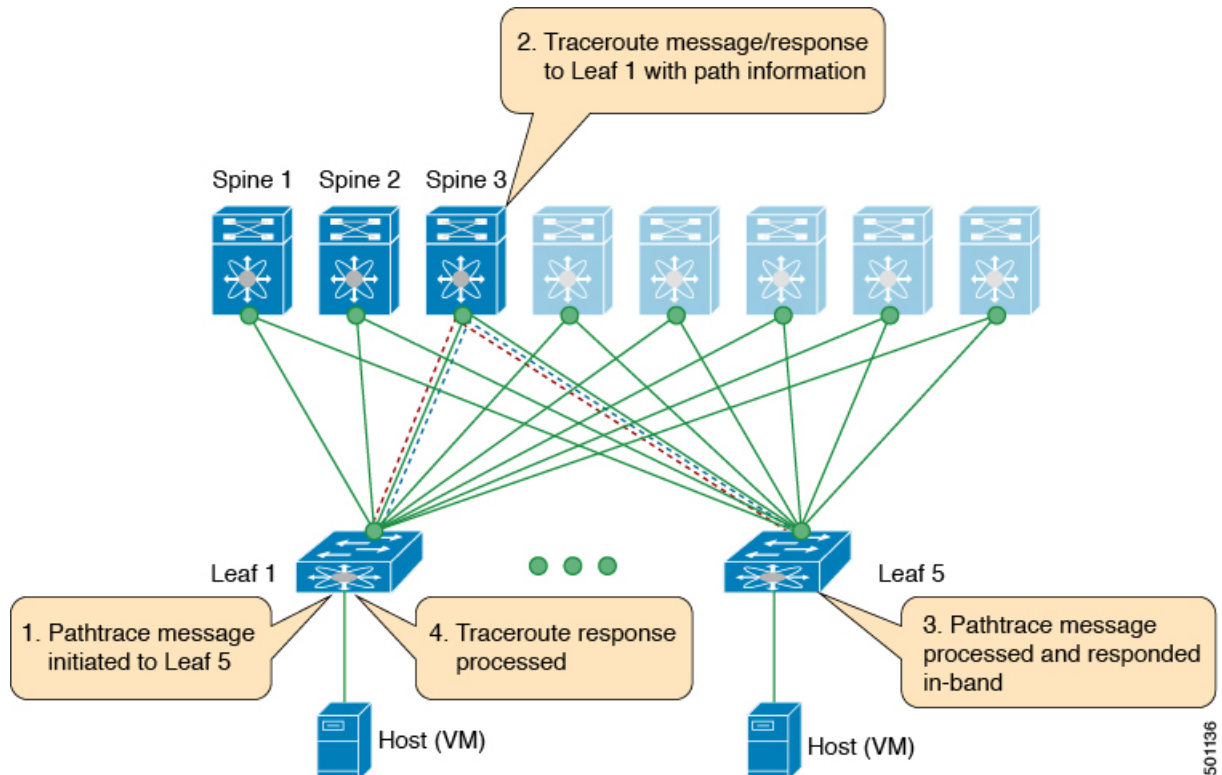
- Traceroute uses the ICMP packets (channel-1), encapsulated in the VXLAN encapsulation to reach the host

### Pathtrace

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

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

Figure 18: Traceroute Message



## Guidelines and Limitations for VXLAN NGOAM

VXLAN NGOAM has the following guidelines and limitations:

- Beginning with Cisco NX-OS Release 9.2(3), support is added for Cisco Nexus 9504 and 9508 switches with -R line cards.

## Configuring VXLAN OAM

### Before you begin

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

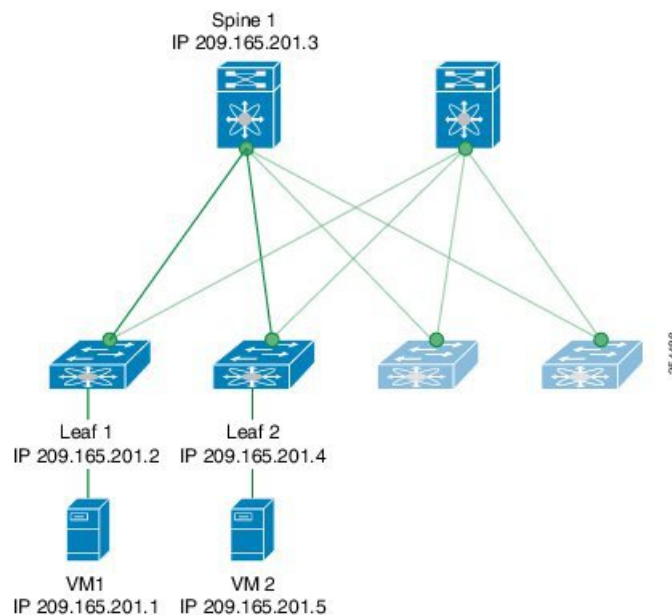
**Procedure**

|               | Command or Action                                                                 | Purpose                                                                                                                                                                                                                                                                                                                                           |
|---------------|-----------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 1</b> | switch# <b>configure terminal</b>                                                 | Enters global configuration mode.                                                                                                                                                                                                                                                                                                                 |
| <b>Step 2</b> | switch(config)# <b>feature ngoam</b>                                              | Enters the NGOAM feature.                                                                                                                                                                                                                                                                                                                         |
| <b>Step 3</b> | switch(config)# <b>hardware access-list tcam region arp-ether 256 double-wide</b> | For Cisco Nexus 9300 platform switches with Network Forwarding Engine (NFE), configure the TCAM region for ARP-ETHER using this command. This step is essential to program the ACL rule in the hardware and it is a prerequisite before installing the ACL rule.<br><br><b>Note</b> Configuring the TCAM region requires the node to be rebooted. |
| <b>Step 4</b> | switch(config)# <b>ngoam install acl</b>                                          | Installs the NGOAM Access Control List (ACL).                                                                                                                                                                                                                                                                                                     |
| <b>Step 5</b> | (Optional) <b>bcm-shell module 1 "fp show group 62"</b>                           | For Cisco Nexus 9300 Series switches with Network Forwarding Engine (NFE), complete this verification step. After entering the command, perform a lookup for entry/eid with data=0x8902 under EtherType.                                                                                                                                          |

**Example**

See the following examples of the configuration topology.

**Figure 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 examples display how to ping from Leaf 1 to VM2 via Spine 1 with channel 1 (unique loopback) and with channel 2 (NVO3 Draft Tissa):

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

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

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

<<<< add space here
switch# ping nve ip unknown vrf vni-31000 payload ip 209.165.201.5 209.165.201.4 payload-end
verify-host
<snip>
Sender handle: 34
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/4/18 ms
Total time elapsed 49 ms
```



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

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

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

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

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

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

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

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

```

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

```

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

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

```

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

```

```

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

```

```

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

```

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

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

```

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

```

```

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

```



#### Note

When the total hop count to final destination is more than 5, the path trace default TTL value is 5. Use **max-ttl** option to finish VXLAN OAM path trace completely.

For example: **pathtrace nve ip unknown vrf vni-31001 payload ip 200.1.1.71 200.1.1.23 payload-end verbose max-ttl 10**

## Configuring NGOAM Profile

Complete the following steps to configure NGOAM profile.

**Procedure**

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



## CHAPTER 8

# Configuring vPC Multi-Homing

This chapter contains the following sections:

- [Advertising Primary IP Address, on page 147](#)
- [BorderPE Switches in a vPC Setup, on page 148](#)
- [DHCP Configuration in a vPC Setup, on page 148](#)
- [IP Prefix Advertisement in vPC Setup, on page 148](#)

## Advertising Primary IP Address

On a vPC enabled leaf or border leaf switch, by default all Layer-3 routes are advertised with the secondary IP address (VIP) of the leaf switch VTEP as the BGP next-hop IP address. Prefix routes and leaf switch generated routes are not synced between vPC leaf switches. Using the VIP as the BGP next-hop for these types of routes can cause traffic to be forwarded to the wrong vPC leaf or border leaf switch and black-holed. The provision to use the primary IP address (PIP) as the next-hop when advertising prefix routes or loopback interface routes in BGP on vPC enabled leaf or border leaf switches allows users to select the PIP as BGP next-hop when advertising these types of routes, so that traffic will always be forwarded to the right vPC enabled leaf or border leaf switch.

The configuration command for advertising the PIP is **advertise-pip**.

The following is a sample configuration:

```
switch(config)# router bgp 65536
 address-family 12vpn evpn
 advertise-pip
 interface nve 1
 advertise virtual-rmac
```

The **advertise-pip** command lets BGP use the PIP as next-hop when advertising externally learned routes or for the redistributed direct routes if vPC is enabled.

VMAC (virtual-mac) is used with VIP and system MAC is used with PIP when the VIP/PIP feature is enabled.

With the **advertise-pip** and **advertise virtual-rmac** commands enabled, type 5 routes are advertised with PIP and type 2 routes are still advertised with VIP. In addition, VMAC will be used with VIP and system MAC will be used with PIP.



**Note** The **advertise-pip** and **advertise-virtual-rmac** commands must be enabled and disabled together for this feature to work properly. If you enable or disable one and not the other, it is considered an invalid configuration. For Cisco Nexus 9504 and 9508 switches with -R line cards, always configure **advertise virtual-rmac** without **advertise-pip**.

## BorderPE Switches in a vPC Setup

The two borderPE switches are configured as a vPC. In a VXLAN vPC deployment, a common, virtual VTEP IP address (secondary loopback IP address) is used for communication. The common, virtual VTEP uses a system specific router MAC address. The Layer-3 prefixes or default route from the borderPE switch is advertised with this common virtual VTEP IP (secondary IP) plus the system specific router MAC address as the next hop.

Entering the **advertise-pip** and **advertise virtual-rmac** commands cause the Layer 3 prefixes or default to be advertised with the primary IP and system-specific router MAC address, the MAC addresses to be advertised with the secondary IP, and a router MAC address derived from the secondary IP address.

## DHCP Configuration in a vPC Setup

When DHCP or DHCPv6 relay function is configured on leaf switches in a vPC setup, and the DHCP server is in the non default, non management VRF, then configure the **advertise-pip** command on the vPC leaf switches. This allows BGP EVPN to advertise Route-type 5 routes with the next-hop using the primary IP address of the VTEP interface.

The following is a sample configuration:

```
switch(config)# router bgp 100
 address-family 12vpn evpn
 advertise-pip
 interface nve 1
 advertise virtual-rmac
```

## IP Prefix Advertisement in vPC Setup

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

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





## CHAPTER 9

# Configuring Multi-Site

This chapter contains the following sections:

- [About VXLAN EVPN Multi-Site, on page 149](#)
- [Guidelines and Limitations for VXLAN EVPN Multi-Site , on page 150](#)
- [Enabling VXLAN EVPN Multi-Site, on page 152](#)
- [Multi-Site with vPC Support, on page 153](#)
- [Configuring VNI Dual Mode, on page 160](#)
- [Configuring Fabric/DCI Link Tracking, on page 161](#)
- [Configuring Fabric External Neighbors, on page 162](#)

## About VXLAN EVPN Multi-Site

The VXLAN EVPN Multi-Site solution interconnects two or more BGP-based Ethernet VPN (EVPN) sites/fabrics (overlay domains) in a scalable fashion over an IP-only network. This solution uses border gateways (BGWs) in anycast or vPC mode to terminate and interconnect two sites. The BGWs provide the network control boundary that is necessary for traffic enforcement and failure containment functionality.

In the BGP control plane, BGP sessions between the BGWs rewrite the next hop information of EVPN routes and reoriginate them.

VXLAN Tunnel Endpoints (VTEPs) are only aware of their overlay domain internal neighbors, including the BGWs. All routes external to the fabric have a next hop on the BGWs for Layer 2 and Layer 3 traffic.

The BGW is the node that interacts with nodes within a site and with nodes that are external to the site. For example, in a leaf-spine data center fabric, it can be a leaf, a spine, or a separate device acting as a gateway to interconnect the sites.

The VXLAN EVPN Multi-Site feature can be conceptualized as multiple site-local EVPN control planes and IP forwarding domains interconnected via a single common EVPN control and IP forwarding domain. Every EVPN node is identified with a unique site-scope identifier. A site-local EVPN domain consists of EVPN nodes with the same site identifier. BGWs on one hand are also part of the site-specific EVPN domain and on the other hand a part of a common EVPN domain to interconnect with BGWs from other sites. For a given site, these BGWs facilitate site-specific nodes to visualize all other sites to be reachable only via them. This means:

- Site-local bridging domains are interconnected only via BGWs with bridging domains from other sites.
- Site-local routing domains are interconnected only via BGWs with routing domains from other sites.

- Site-local flood domains are interconnected only via BGWs with flood domains from other sites.

Selective Advertisement is defined as the configuration of the per-tenant information on the BGW. Specifically, this means IP VRF or MAC VRF (EVPN instance). In cases where external connectivity (VRF-lite) and EVPN Multi-Site coexist on the same BGW, the advertisements are always enabled.

## Guidelines and Limitations for VXLAN EVPN Multi-Site

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

- Cisco Nexus 9332C and 9364C are supported as border gateways.
- VXLAN EVPN Multi-Site is not supported on Cisco Nexus 9500 platform switches with -R line cards.
- 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.
- VXLAN EVPN Multi-Site and Tenant Routed Multicast (TRM) is supported between source and receivers deployed in the same site.
- 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



**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
- Cisco Nexus 9336C-FX2 switch
- In a VXLAN EVPN multisite deployment, when you use the ttag feature, make sure that the ttag is stripped (**ttag-strip**) on BGW's DCI interfaces that connect to the cloud. To elaborate, if the ttag is attached to non-Nexus 9000 devices that do not support ether-type 0x8905, stripping of ttag is required. However, BGW back-to-back model of DCI does not require ttag stripping.
- The number of border gateways per site is limited to four.
- Beginning with Cisco NX-OS Release 9.2(1), Border Gateways (BGWs) in a vPC topology are supported.
- Support for Multicast Flood Domain between inter-site/fabric border gateways is not supported.
- Multicast Underlay between sites is not supported.
- iBGP EVPN Peering between border gateways of different fabrics/sites is not supported.
- The **peer-type fabric-external** command configuration is required only for VXLAN Multi-site BGWs (this command must not be used when peering with non-Cisco equipment).



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

- Anycast mode can support up to four border gateway's per site.
- Anycast mode can only support Layer 3 services attached to local interfaces.
- In Anycast mode, BUM is replicated to each border-leaf and DF election, between border leafs of a particular site decides which border leaf would forward the traffic inter-site traffic (Fabric to DCI and vice versa) for that site.
- In Anycast mode, all the Layer 3 services are advertised in BGP via EVPN Type-5 routes with their physical IP as the next hop.
- vPC mode can support only two border gateways.
- vPC mode can support both Layer 2 hosts and Layer 3 services on local interfaces.
- In vPC mode, BUM is replicated to either of the border-gateway's for traffic coming from external site and hence both the border gateways are forwarders for site external to site internal (DCI to Fabric) direction.
- In vPC mode, BUM is replicated to either of the border gateways for traffic coming from the local site leaf for a VLAN using Ingress Replication (IR) underlay. Both border gateways are forwarders for site internal to site external ( Fabric to DCI) direction for VLANs using the IR underlay.
- In vPC mode, BUM is replicated to both border gateways for traffic coming from the local site leaf for a VLAN using the multicast underlay. Therefore, a decapper/forwarder election happens and the decapsulation winner/forwarder only forwards the site-local traffic to external site border-gateways for VLANs using the multicast underlay.
- In vPC mode, all the Layer 3 services/attachments are advertised in BGP via EVPN Type-5 routes with their virtual IP as next hop. If the VIP/PIP feature is configured, they are advertised with PIP as the next hop.
- If different Anycast Gateway MAC addresses are configured across sites, 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).
- PIM BiDir is not supported for fabric underlay multicast replication with VXLAN Multi-Site.
- PIM is not supported on multisite VXLAN DCI links.
- FEX is not supported on a vPC BGW and Anycast BGW.
- To improve the convergence in case of fabric link failure and avoid issues in case of fabric link flapping, ensure to configure multi-hop BFD between loopbacks of spines and BGWs.

In the specific scenario where a BGW node becomes completely isolated from the fabric due to all its fabric links failing, the use of multi-hop BFD ensures that the BGP sessions between the spines and the isolated BGW can be immediately brought down, without relying on the configured BGP hold-time value.

# Enabling VXLAN EVPN Multi-Site

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

## Procedure

|               | Command or Action                                                                                                                                                        | Purpose                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 1</b> | <b>configure terminal</b><br><br><b>Example:</b><br>switch# <b>configure terminal</b>                                                                                    | Enters global configuration mode.                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>Step 2</b> | <b>evpn multisite border-gateway <i>ms-id</i></b><br><br><b>Example:</b><br>switch(config)# <b>evpn multisite border-gateway 100</b>                                     | Configures 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 BGWs within the same fabric/site.                                                                                                                                                                                                                                                          |
| <b>Step 3</b> | <b>interface nve 1</b><br><br><b>Example:</b><br>switch(config-evpn-msite-bgw)# <b>interface nve 1</b>                                                                   | Creates a VXLAN overlay interface that terminates VXLAN tunnels.<br><br><b>Note</b> Only one NVE interface is allowed on the switch.                                                                                                                                                                                                                                                                                                              |
| <b>Step 4</b> | <b>source-interface loopback <i>src-if</i></b><br><br><b>Example:</b><br>switch(config-if-nve)# <b>source-interface loopback 0</b>                                       | The source interface must be a loopback interface that is configured on the switch with a valid /32 IP address. This /32 IP address must be known by the transient devices in the transport network and the remote VTEPs. This requirement is accomplished by advertising it through a dynamic routing protocol in the transport network.                                                                                                         |
| <b>Step 5</b> | <b>host-reachability protocol bgp</b><br><br><b>Example:</b><br>switch(config-if-nve)# <b>host-reachability protocol bgp</b>                                             | Defines BGP as the mechanism for host reachability advertisement.                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>Step 6</b> | <b>multisite border-gateway interface loopback <i>vi-num</i></b><br><br><b>Example:</b><br>switch(config-if-nve)# <b>multisite border-gateway interface loopback 100</b> | Defines the loopback interface used for the BGW virtual IP address (VIP). The border-gateway interface must be a loopback interface that is configured on the switch with a valid /32 IP address. This /32 IP address must be known by the transient devices in the transport network and the remote VTEPs. This requirement is accomplished by advertising it through a dynamic routing protocol in the transport network. This loopback must be |

|                | Command or Action                                                                                                      | Purpose                                                                                     |
|----------------|------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------|
|                |                                                                                                                        | different than the source interface loopback. The range of <i>vi-num</i> is from 0 to 1023. |
| <b>Step 7</b>  | <b>no shutdown</b><br><br><b>Example:</b><br><code>switch(config-if-nve) # no shutdown</code>                          | Negates the <b>shutdown</b> command.                                                        |
| <b>Step 8</b>  | <b>exit</b><br><br><b>Example:</b><br><code>switch(config-if-nve) # exit</code>                                        | Exits the NVE configuration mode.                                                           |
| <b>Step 9</b>  | <b>interface loopback loopback-number</b><br><br><b>Example:</b><br><code>switch(config) # interface loopback 0</code> | Configures the loopback interface.                                                          |
| <b>Step 10</b> | <b>ip address ip-address</b><br><br><b>Example:</b><br><code>switch(config-if) # ip address<br/>198.0.2.0/32</code>    | Configures the IP address for the loopback interface.                                       |

## Multi-Site with vPC Support

### About Multi-Site with vPC Support

The BGWs can be in a vPC complex. In this case, it is possible to support dually-attached directly-connected hosts that might be bridged or routed as well as dually-attached firewalls or service attachments. The vPC BGWs have vPC-specific multihoming techniques and do not rely on EVPN Type 4 routes for DF election or split horizon.

### Guidelines and Limitations for Multi-Site with vPC Support

Multi-Site with vPC support has the following configuration guidelines and limitations:

- 4000 VNIs for vPC are not supported.
- For BUM with continued VIP use, the MCT link is used as transport upon core isolation or fabric isolation, and for unicast traffic in fabric isolation.
- The routes to remote Multisite BGW loopback addresses must always prioritize the DCI link path over the iBGP protocol between vPC Border Gateway switches configured using the backup SVI. The backup SVI should be used strictly in the event of a DCI link failure.

## Configuring Multi-Site with vPC Support

This procedure describes the configuration of Multi-Site with vPC support:

- Configure vPC domain.
- Configure port channels.
- Configuring vPC Peer Link.

### Procedure

|               | Command or Action                                                                                                                                                           | Purpose                                                                                                                                                   |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 1</b> | <b>configure terminal</b><br><br><b>Example:</b><br><code>switch# configure terminal</code>                                                                                 | Enters global configuration mode.                                                                                                                         |
| <b>Step 2</b> | <b>feature vpc</b><br><br><b>Example:</b><br><code>switch(config)# feature vpc</code>                                                                                       | Enables vPCs on the device.                                                                                                                               |
| <b>Step 3</b> | <b>feature interface-vlan</b><br><br><b>Example:</b><br><code>switch(config)# feature interface-vlan</code>                                                                 | Enables the interface VLAN feature on the device.                                                                                                         |
| <b>Step 4</b> | <b>feature lacp</b><br><br><b>Example:</b><br><code>switch(config)# feature lacp</code>                                                                                     | Enables the LACP feature on the device.                                                                                                                   |
| <b>Step 5</b> | <b>feature pim</b><br><br><b>Example:</b><br><code>switch(config)# feature pim</code>                                                                                       | Enables the PIM feature on the device.                                                                                                                    |
| <b>Step 6</b> | <b>feature ospf</b><br><br><b>Example:</b><br><code>switch(config)# feature ospf</code>                                                                                     | Enables the OSPF feature on the device.                                                                                                                   |
| <b>Step 7</b> | <b>ip pim rp-address <i>address</i> group-list <i>range</i></b><br><br><b>Example:</b><br><code>switch(config)# ip pim rp-address 100.100.100.1 group-list 224.0.0/4</code> | Defines a PIM RP address for the underlay multicast group range.                                                                                          |
| <b>Step 8</b> | <b>vpc domain <i>domain-id</i></b><br><br><b>Example:</b><br><code>switch(config)# vpc domain 1</code>                                                                      | Creates a vPC domain on the device and enters vpn-domain configuration mode for configuration purposes. There is no default. The range is from 1 to 1000. |

|                | Command or Action                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Purpose                                                                                                                                                                                                                                                  |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 9</b>  | <b>peer switch</b><br><b>Example:</b><br><pre>switch(config-vpc-domain) # peer switch</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Defines the peer switch.                                                                                                                                                                                                                                 |
| <b>Step 10</b> | <b>peer gateway</b><br><b>Example:</b><br><pre>switch(config-vpc-domain) # peer gateway</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Enables Layer 3 forwarding for packets destined to the gateway MAC address of the vPC.                                                                                                                                                                   |
| <b>Step 11</b> | <b>peer-keepalive destination ip-address</b><br><b>Example:</b><br><pre>switch(config-vpc-domain) #<br/>peer-keepalive destination 172.28.230.85</pre>                                                                                                                                                                                                                                                                                                                                                                                                                     | <p>Configures the IPv4 address for the remote end of the vPC peer-keepalive link.</p> <p><b>Note</b> The system does not form the vPC peer link until you configure a vPC peer-keepalive link.</p> <p>The management ports and VRF are the defaults.</p> |
| <b>Step 12</b> | <b>ip arp synchronize</b><br><b>Example:</b><br><pre>switch(config-vpc-domain) # ip arp<br/>synchronize</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Enables IP ARP synchronize under the vPC domain to facilitate faster ARP table population following device reload.                                                                                                                                       |
| <b>Step 13</b> | <b>ipv6 nd synchronize</b><br><b>Example:</b><br><pre>switch(config-vpc-domain) # ipv6 nd<br/>synchronize</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Enables IPv6 ND synchronization under the vPC domain to facilitate faster ND table population following device reload.                                                                                                                                   |
| <b>Step 14</b> | <p>Create the vPC peer-link.</p> <b>Example:</b><br><pre>switch(config) # interface port-channel<br/>1<br/>switch(config) # switchport<br/>switch(config) # switchport mode trunk<br/>switch(config) # switchport trunk allowed<br/>vlan 1,10,100-200<br/>switch(config) # mtu 9216<br/>switch(config) # vpc peer-link<br/>switch(config) # no shut</pre> <pre>switch(config) # interface Ethernet 1/1,<br/>1/21<br/>switch(config) # switchport<br/>switch(config) # mtu 9216<br/>switch(config) # channel-group 1 mode<br/>active<br/>switch(config) # no shutdown</pre> | Creates the vPC peer-link port-channel interface and adds two member interfaces to it.                                                                                                                                                                   |
| <b>Step 15</b> | <b>system nve infra-vlans range</b><br><b>Example:</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Defines a non-VXLAN-enabled VLAN as a backup routed path.                                                                                                                                                                                                |

|                | Command or Action                                                                                                                                                                                                                                                                                                                                                                                            | Purpose                                                                                                                                                                                                                                                                                                                          |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | <code>switch(config)# system nve infra-vlans 10</code>                                                                                                                                                                                                                                                                                                                                                       |                                                                                                                                                                                                                                                                                                                                  |
| <b>Step 16</b> | <b>vlan number</b><br><b>Example:</b><br><code>switch(config)# vlan 10</code>                                                                                                                                                                                                                                                                                                                                | Creates the VLAN to be used as an infra-VLAN.                                                                                                                                                                                                                                                                                    |
| <b>Step 17</b> | Create the SVI.<br><b>Example:</b><br><code>switch(config)# interface vlan 10</code><br><code>switch(config)# ip address 10.10.10.1/30</code><br><code>switch(config)# ip router ospf process UNDERLAY area 0</code><br><code>switch(config)# ip pim sparse-mode</code><br><code>switch(config)# no ip redirects</code><br><code>switch(config)# mtu 9216</code><br><code>switch(config)# no shutdown</code> | Creates the SVI used for the backup routed path over the vPC peer-link.                                                                                                                                                                                                                                                          |
| <b>Step 18</b> | (Optional) <b>delay restore interface-vlan seconds</b><br><b>Example:</b><br><code>switch(config-vpc-domain)# delay restore interface-vlan 45</code>                                                                                                                                                                                                                                                         | Enables the delay restore timer for SVIs. We recommend tuning this value when the SVI/VNI scale is high. For example, when the SCI count is 1000, we recommend that you set the delay restore to 45 seconds.                                                                                                                     |
| <b>Step 19</b> | <b>evpn multisite border-gateway ms-id</b><br><b>Example:</b><br><code>switch(config)# evpn multisite border-gateway 100</code>                                                                                                                                                                                                                                                                              | Configures the site ID for a site/fabric. The range of values for <i>ms-id</i> is 1 to 281474976710655. The <i>ms-id</i> must be the same in all BGWs within the same fabric/site.                                                                                                                                               |
| <b>Step 20</b> | <b>interface nve 1</b><br><b>Example:</b><br><code>switch(config-evpn-msite-bgw)# interface nve 1</code>                                                                                                                                                                                                                                                                                                     | Creates a VXLAN overlay interface that terminates VXLAN tunnels.<br><b>Note</b> Only one NVE interface is allowed on the switch.                                                                                                                                                                                                 |
| <b>Step 21</b> | <b>source-interface loopback src-if</b><br><b>Example:</b><br><code>switch(config-if-nve)# source-interface loopback 0</code>                                                                                                                                                                                                                                                                                | Defines the source interface, which must be a loopback interface with a valid /32 IP address. This /32 IP address must be known by the transient devices in the transport network and the remote VTEPs. This requirement is accomplished by advertising the address through a dynamic routing protocol in the transport network. |
| <b>Step 22</b> | <b>host-reachability protocol bgp</b><br><b>Example:</b><br><code>switch(config-if-nve)# host-reachability protocol bgp</code>                                                                                                                                                                                                                                                                               | Defines BGP as the mechanism for host reachability advertisement.                                                                                                                                                                                                                                                                |



|                | Command or Action                                                                                                                                                         | Purpose                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 23</b> | <b>multisite border-gateway interface loopback</b> <i>vi-num</i><br><b>Example:</b><br><pre>switch(config-if-nve) # multisite border-gateway interface loopback 100</pre> | Defines the loopback interface used for the BGW virtual IP address (VIP). The BGW interface must be a loopback interface that is configured on the switch with a valid /32 IP address. This /32 IP address must be known by the transient devices in the transport network and the remote VTEPs. This requirement is accomplished by advertising the address through a dynamic routing protocol in the transport network. This loopback must be different than the source interface loopback. The range of <i>vi-num</i> is from 0 to 1023. |
| <b>Step 24</b> | <b>no shutdown</b><br><b>Example:</b><br><pre>switch(config-if-nve) # no shutdown</pre>                                                                                   | Negates the <b>shutdown</b> command.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>Step 25</b> | <b>exit</b><br><b>Example:</b><br><pre>switch(config-if-nve) # exit</pre>                                                                                                 | Exits the NVE configuration mode.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>Step 26</b> | <b>interface loopback</b> <i>loopback-number</i><br><b>Example:</b><br><pre>switch(config) # interface loopback 0</pre>                                                   | Configures the loopback interface.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <b>Step 27</b> | <b>ip address</b> <i>ip-address</i><br><b>Example:</b><br><pre>switch(config-if) # ip address 198.0.2.0/32</pre>                                                          | Configures the primary IP address for the loopback interface.                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <b>Step 28</b> | <b>ip address</b> <i>ip-address</i> <b>secondary</b><br><b>Example:</b><br><pre>switch(config-if) # ip address 198.0.2.1/32 secondary</pre>                               | Configures the secondary IP address for the loopback interface.                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>Step 29</b> | <b>ip pim sparse-mode</b><br><b>Example:</b><br><pre>switch(config-if) # ip pim sparse-mode</pre>                                                                         | Configures PIM sparse mode on the loopback interface.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |

## Configuring Peer Link as Transport in Case of Link Failure

This procedure describes the configuration of an SVI interface configured with a high IGP cost to ensure it is only used as a backup link.



**Note** This configuration is required to use the peer link as a backup link during fabric and/or DCI link failures.

### Procedure

|               | Command or Action                                                                                                          | Purpose                                                                                                                                                                                                                  |
|---------------|----------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 1</b> | <b>configure terminal</b><br><br><b>Example:</b><br>switch# <b>configure terminal</b>                                      | Enters global configuration mode.                                                                                                                                                                                        |
| <b>Step 2</b> | <b>system nve infra-vlans <i>vlan-range</i></b><br><br><b>Example:</b><br>switch(config)# <b>system nve infra-vlans 10</b> | Specifies VLANs used by all SVI interfaces for uplink and vPC peer-links in VXLAN as infra-VLANs. You should not configure certain combinations of infra-VLANs. For example, 2 and 514, 10 and 522, which are 512 apart. |
| <b>Step 3</b> | <b>interface <i>vlan-id</i></b><br><br><b>Example:</b><br>switch(config)# <b>interface vlan10</b>                          | Configures the interface.                                                                                                                                                                                                |
| <b>Step 4</b> | <b>no shutdown</b><br><br><b>Example:</b><br>switch(config-if)# <b>no shutdown</b>                                         | Negates the <b>shutdown</b> command.                                                                                                                                                                                     |
| <b>Step 5</b> | <b>mtu <i>value</i></b><br><br><b>Example:</b><br>switch(config-if)# <b>mtu 9216</b>                                       | Sets the maximum transmission unit (MTU).                                                                                                                                                                                |
| <b>Step 6</b> | <b>no ip redirects</b><br><br><b>Example:</b><br>switch(config-if)# <b>no ip redirects</b>                                 | Prevents the device from sending redirects.                                                                                                                                                                              |
| <b>Step 7</b> | <b>ip address <i>ip-address/length</i></b><br><br><b>Example:</b><br>switch(config-if)# <b>ip address 35.1.1.2/24</b>      | Configures an IP address for this interface.                                                                                                                                                                             |
| <b>Step 8</b> | <b>no ipv6 redirects</b><br><br><b>Example:</b><br>switch(config-if)# <b>no ipv6 redirects</b>                             | Disables the ICMP redirect messages on BFD-enabled interfaces.                                                                                                                                                           |
| <b>Step 9</b> | <b>ip ospf cost <i>cost</i></b><br><br><b>Example:</b><br>switch(config-if)# <b>ip ospf cost 100</b>                       | Configures the OSPF cost metric for this interface.                                                                                                                                                                      |

|                | Command or Action                                                                                                          | Purpose                                                                          |
|----------------|----------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------|
| <b>Step 10</b> | <b>ip ospf network point-to-point</b><br><br><b>Example:</b><br>switch(config-if)# ip ospf network point-to-point          | Specifies the OSPF point-to-point network.                                       |
| <b>Step 11</b> | <b>ip router ospf instance area area-number</b><br><br><b>Example:</b><br>switch(config-if)# ip router ospf 1 area 0.0.0.0 | Configures the routing process for the IP on an interface and specifies an area. |
| <b>Step 12</b> | <b>ip pim sparse-mode</b><br><br><b>Example:</b><br>switch(config-if)# ip pim sparse-mode                                  | Configures sparse-mode PIM on an interface.                                      |

## Verifying the Multi-Site with vPC Support Configuration

To display Multi-Site with vPC support information, enter one of the following commands:

|                                               |                                                                                                                    |
|-----------------------------------------------|--------------------------------------------------------------------------------------------------------------------|
| <b>show vpc brief</b>                         | Displays general vPC and CC status.                                                                                |
| <b>show vpc consistency-parameters global</b> | Displays the status of those parameters that must be consistent across all vPC interfaces.                         |
| <b>show vpc consistency-parameters vni</b>    | Displays configuration information for VNIs under the NVE interface that must be consistent across both vPC peers. |

Output example for the **show vpc brief** command:

```
switch# show vpc brief
```

Legend:

(\*) - local vPC is down, forwarding via vPC peer-link

```
vPC domain id : 1
Peer status : peer adjacency formed ok (<--- peer up)
vPC keep-alive status : peer is alive
Configuration consistency status : success (<----- CC passed)
Per-vlan consistency status : success (<----- per-VNI CCpassed)
Type-2 consistency status : success
vPC role : secondary
Number of vPCs configured : 1
Peer Gateway : Enabled
Dual-active excluded VLANs : -
Graceful Consistency Check : Enabled
Auto-recovery status : Enabled, timer is off.(timeout = 240s)
Delay-restore status : Timer is off.(timeout = 30s)
Delay-restore SVI status : Timer is off.(timeout = 10s)
Operational Layer3 Peer-router : Disabled
[...]
```

Output example for the **show vpc consistency-parameters global** command:

```
switch# show vpc consistency-parameters global
```

Legend:

Type 1 : vPC will be suspended in case of mismatch

| Name                                                                                                           | Type | Local Value                                                                                       | Peer Value                                                                                        |
|----------------------------------------------------------------------------------------------------------------|------|---------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------|
| [...]                                                                                                          |      |                                                                                                   |                                                                                                   |
| Nve1 Adm St, Src Adm St,<br>Sec IP, Host Reach, VMAC<br>Adv, SA, mcast l2, mcast<br>l3, IR BGP, MS Adm St, Reo | 1    | Up, Up, 2.1.44.5, CP,<br>TRUE, Disabled,<br>0.0.0.0, 0.0.0.0,<br>Disabled, Up,<br>200.200.200.200 | Up, Up, 2.1.44.5, CP,<br>TRUE, Disabled,<br>0.0.0.0, 0.0.0.0,<br>Disabled, Up,<br>200.200.200.200 |
| [...]                                                                                                          |      |                                                                                                   |                                                                                                   |

Output example for the **show vpc consistency-parameters vni** command:

```
switch(config-if-nve-vni)# show vpc consistency-parameters vni
```

Legend:

Type 1 : vPC will be suspended in case of mismatch

| Name                                  | Type | Local Value                           | Peer Value                            |
|---------------------------------------|------|---------------------------------------|---------------------------------------|
| [...]                                 |      |                                       |                                       |
| Nve1 Vni, Mcast, Mode,<br>Type, Flags | 1    | 11577, 234.1.1.1,<br>Mcast, L2, MS IR | 11577, 234.1.1.1,<br>Mcast, L2, MS IR |
| Nve1 Vni, Mcast, Mode,<br>Type, Flags | 1    | 11576, 234.1.1.1,<br>Mcast, L2, MS IR | 11576, 234.1.1.1,<br>Mcast, L2, MS IR |
| [...]                                 |      |                                       |                                       |

## Configuring VNI Dual Mode

This procedure describes the configuration of the BUM traffic domain for a given VLAN. Support exists for using multicast or ingress replication inside the fabric/site and ingress replication across different fabrics/sites.



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

For more information about configuring multicast or ingress replication for a large number of VNIs, see [Example of VXLAN BGP EVPN \(EBGP\), on page 94](#).

### Procedure

|               | Command or Action                                                                     | Purpose                                                          |
|---------------|---------------------------------------------------------------------------------------|------------------------------------------------------------------|
| <b>Step 1</b> | <b>configure terminal</b><br><br><b>Example:</b><br>switch# <b>configure terminal</b> | Enters global configuration mode.                                |
| <b>Step 2</b> | <b>interface nve 1</b><br><br><b>Example:</b>                                         | Creates a VXLAN overlay interface that terminates VXLAN tunnels. |

|               | Command or Action                                                                                                                             | Purpose                                                                                                                                                                                                                                                          |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|               | <code>switch(config)# interface nve 1</code>                                                                                                  | <b>Note</b> Only one NVE interface is allowed on the switch.                                                                                                                                                                                                     |
| <b>Step 3</b> | <b>member vni</b> <i>vni-range</i><br><b>Example:</b><br><code>switch(config-if-nve)# member vni 200</code>                                   | Configures 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.<br><br><b>Note</b> Enter one of the Step 4 or Step 5 commands. |
| <b>Step 4</b> | <b>mcast-group</b> <i>ip-addr</i><br><b>Example:</b><br><code>switch(config-if-nve-vni)# mcast-group 255.0.4.1</code>                         | Configures the NVE Multicast group IP prefix within the fabric.                                                                                                                                                                                                  |
| <b>Step 5</b> | <b>ingress-replication protocol</b> <i>bgp</i><br><b>Example:</b><br><code>switch(config-if-nve-vni)# ingress-replication protocol bgp</code> | Enables BGP EVPN with ingress replication for the VNI within the fabric.                                                                                                                                                                                         |
| <b>Step 6</b> | <b>multisite ingress-replication</b><br><b>Example:</b><br><code>switch(config-if-nve-vni)# multisite ingress-replication</code>              | Defines the Multi-Site BUM replication method for extending the 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 reorigination of EVPN routes either from or to a site if all the DCI/fabric links go down.

### Procedure

|               | Command or Action                                                                                              | Purpose                                                                                                                                          |
|---------------|----------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 1</b> | <b>configure terminal</b><br><b>Example:</b><br><code>switch# configure terminal</code>                        | Enters global configuration mode.                                                                                                                |
| <b>Step 2</b> | <b>interface ethernet</b> <i>port</i><br><b>Example:</b><br><code>switch(config)# interface ethernet1/1</code> | Enters interface configuration mode for the DCI or fabric interface.<br><br><b>Note</b> Enter one of the following commands in Step 3 or Step 4. |

|               | Command or Action                                                                                                                    | Purpose                                                                                                                                              |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 3</b> | <b>evpn multisite dci-tracking</b><br><br><b>Example:</b><br>switch(config-if) # <b>evpn multisite dci-tracking</b>                  | Configures DCI interface tracking.                                                                                                                   |
| <b>Step 4</b> | (Optional) <b>evpn multisite fabric-tracking</b><br><br><b>Example:</b><br>switch(config-if) # <b>evpn multisite fabric-tracking</b> | Configures EVPN Multi-Site fabric tracking.<br><br>The <b>evpn multisite fabric-tracking</b> is mandatory for anycast BGWs and vPC BGW fabric links. |
| <b>Step 5</b> | <b>ip address ip-addr</b><br><br><b>Example:</b><br>switch(config-if) # <b>ip address 192.1.1.1</b>                                  | Configures the IP address.                                                                                                                           |
| <b>Step 6</b> | <b>no shutdown</b><br><br><b>Example:</b><br>switch(config-if) # <b>no shutdown</b>                                                  | Negates the <b>shutdown</b> command.                                                                                                                 |

## Configuring Fabric External Neighbors

This procedure describes the configuration of fabric external/DCI neighbors for communication to other site/fabric BGWs.

### Procedure

|               | Command or Action                                                                                        | Purpose                                                                                             |
|---------------|----------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|
| <b>Step 1</b> | <b>configure terminal</b><br><br><b>Example:</b><br>switch# <b>configure terminal</b>                    | Enters global configuration mode.                                                                   |
| <b>Step 2</b> | <b>router bgp as-num</b><br><br><b>Example:</b><br>switch(config) # <b>router bgp 100</b>                | Configures the autonomous system number.<br>The range for <i>as-num</i> is from 1 to 4,294,967,295. |
| <b>Step 3</b> | <b>neighbor ip-addr</b><br><br><b>Example:</b><br>switch(config-router) # <b>neighbor 100.0.0.1</b>      | Configures a BGP neighbor.                                                                          |
| <b>Step 4</b> | <b>remote-as value</b><br><br><b>Example:</b><br>switch(config-router-neighbor) # <b>remote-as 69000</b> | Configures remote peer's autonomous system number.                                                  |

|               | Command or Action                                                                                                            | Purpose                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------|------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 5</b> | <b>peer-type fabric-external</b><br><b>Example:</b><br><pre>switch(config-router-neighbor) # peer-type fabric-external</pre> | <p>Enables the next hop rewrite for Multi-Site. Defines site external BGP neighbors for EVPN exchange. The default for <b>peer-type</b> is <b>fabric-internal</b>.</p> <p><b>Note</b> The <b>peer-type fabric-external</b> command is required only for VXLAN Multi-Site BGWs. It is not required for pseudo BGWs.</p>                                                                                                                          |
| <b>Step 6</b> | <b>address-family l2vpn evpn</b><br><b>Example:</b><br><pre>switch(config-router-neighbor) # address-family l2vpn evpn</pre> | Configures the address family Layer 2 VPN EVPN under the BGP neighbor.                                                                                                                                                                                                                                                                                                                                                                          |
| <b>Step 7</b> | <b>rewrite-evpn-rt-asn</b><br><b>Example:</b><br><pre>switch(config-router-neighbor) # rewrite-evpn-rt-asn</pre>             | 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. |







## CHAPTER 10

# Configuring Tenant Routed Multicast

This chapter contains the following sections:

- [About Tenant Routed Multicast, on page 165](#)
- [About Tenant Routed Multicast Mixed Mode, on page 167](#)
- [Guidelines and Limitations for Tenant Routed Multicast, on page 167](#)
- [Guidelines and Limitations for Layer 3 Tenant Routed Multicast, on page 168](#)
- [Guidelines and Limitations for Layer 2/Layer 3 Tenant Routed Multicast \(Mixed Mode\), on page 168](#)
- [Rendezvous Point for Tenant Routed Multicast, on page 169](#)
- [Configuring a Rendezvous Point for Tenant Routed Multicast, on page 169](#)
- [Configuring a Rendezvous Point Inside the VXLAN Fabric, on page 170](#)
- [Configuring an External Rendezvous Point, on page 171](#)
- [Configuring Layer 3 Tenant Routed Multicast, on page 173](#)
- [Configuring TRM on the VXLAN EVPN Spine, on page 177](#)
- [Configuring Tenant Routed Multicast in Layer 2/Layer 3 Mixed Mode, on page 179](#)
- [Configuring Layer 2 Tenant Routed Multicast, on page 184](#)
- [Configuring TRM with vPC Support, on page 184](#)

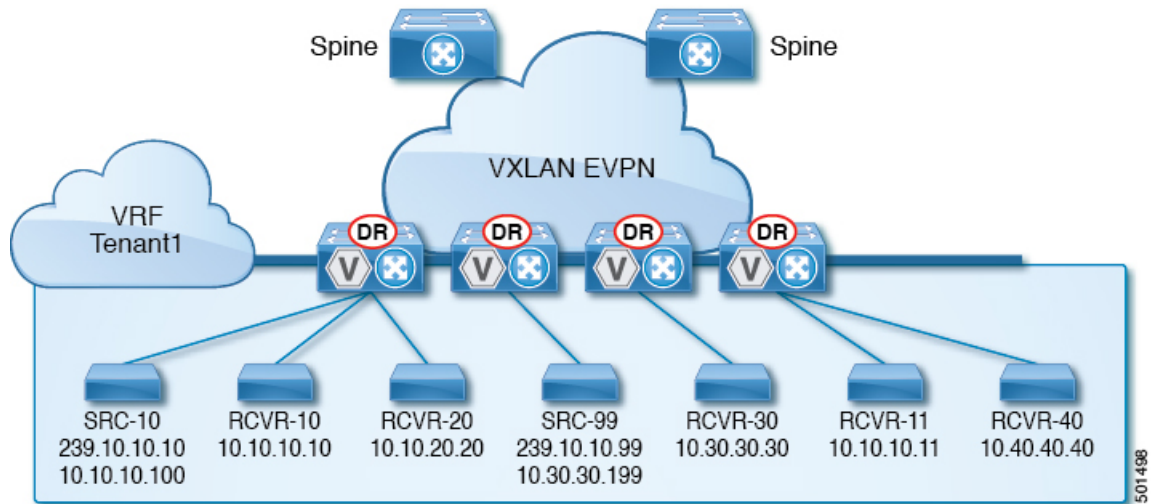
## About Tenant Routed Multicast

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 20: VXLAN EVPN TRM

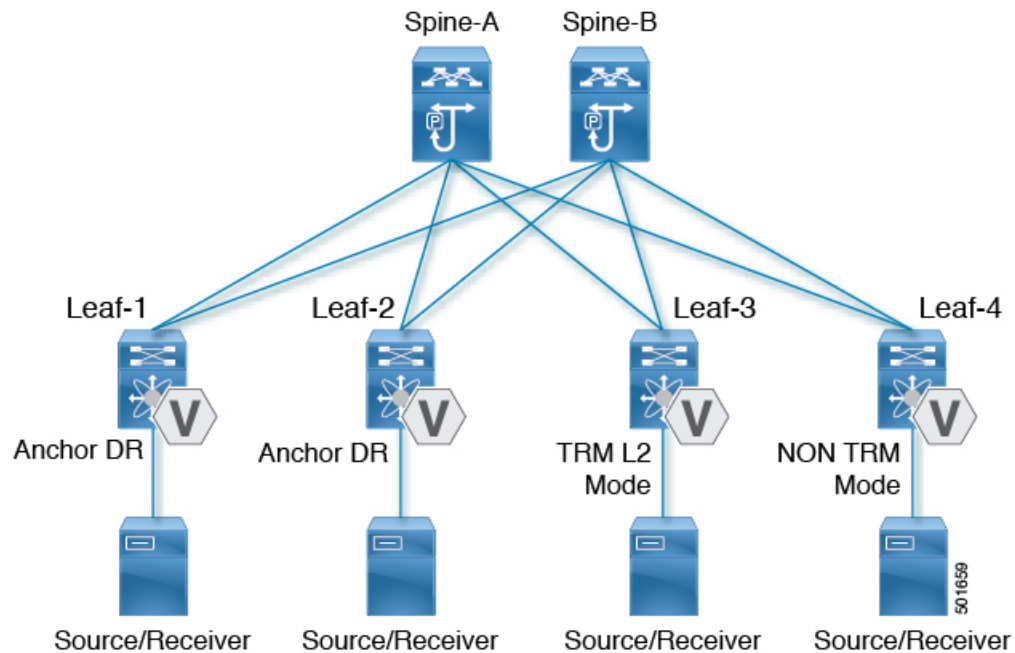


With TRM enabled, multicast forwarding in the underlay is leveraged to replicate VXLAN encapsulated routed multicast traffic. A Default Multicast Distribution Tree (Default-MDT) is built per-VRF. This is an addition to the existing multicast groups for Layer-2 VNI Broadcast, Unknown Unicast, and Layer-2 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 21: TRM Layer 2/Layer 3 Mixed Mode



## Guidelines and Limitations for Tenant Routed Multicast

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

- Tenant Routed Multicast is not supported on Cisco Nexus 9500 platform switches with -R line cards.
- The [Guidelines and Limitations for VXLAN, on page 9](#) also apply to TRM
- With TRM enabled, SVI as a core link is not supported.
- With TRM enabled, Multicast Source/Receiver behind FEX is not supported.
- If TRM is configured, ISSU is disruptive.
- TRM supports IPv4 multicast only.
- TRM requires an IPv4 multicast-based underlay using PIM Any Source Multicast (ASM) which is also known as sparse mode.
- TRM supports overlay PIM ASM and PIM SSM only. PIM BiDir is not supported in the overlay.
- RP has to be configured either internal or external to the fabric.
- The internal RP must be configured on all TRM-enabled VTEPs including the border nodes.
- The external RP must be external to the border nodes.

- The RP must be configured within the VRF pointing to the external RP IP address (static RP). This ensures that unicast and multicast routing is enabled to reach the external RP in the given VRF.
- TRM supports multiple border nodes. Beginning with Cisco NX-OS Release 9.2(3), 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).
- 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.

## Guidelines and Limitations for Layer 3 Tenant Routed Multicast

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

- Layer 3 TRM is supported for Cisco Nexus 9200, 9300-EX, and 9300-FX/FX2/FX3/FXP and 9300-GX platform switches.
- When configuring TRM VXLAN BGP EVPN, the following platforms are supported:
  - Cisco Nexus 9200, 9332C, 9364C, 9300-EX, and 9300-FX/FX2/FX3/FXP platform switches.
  - Cisco Nexus 9500 platform switches with 9700-EX line cards, 9700-FX line cards, or a combination of both line cards.
- Layer 3 TRM and VXLAN EVPN Multi-Site are supported on the same physical switch. For more information, see [Configuring Multi-Site](#).
- TRM with vPC border leafs is supported only for Cisco Nexus 9200, 9300-EX, and 9300-FX/FX2 platform switches and Cisco Nexus 9500 platform switches with -EX/FX line cards. The **advertise-pip** and **advertise virtual-rmac** commands must be enabled on the border leafs to support this functionality. For configuration information, see the "Configuring VIP/PIP" section.
- Well-known local scope multicast (224.0.0.0/24) is excluded from TRM and is bridged.
- When an interface NVE is brought down on the border leaf, the internal overlay RP per VRF must be brought down.

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

Layer 2/Layer 3 Tenant Routed Multicast (TRM) has the following configuration guidelines and limitations:

- All TRM Layer 2/Layer 3 configured switches must be Anchor DR. This is because in TRM Layer 2/Layer 3, you can have switches configured with TRM Layer 2 mode that co-exist in the same topology. This mode is necessary if non-TRM and Layer 2 TRM mode edge devices (VTEPs) are present in the same topology.
- Anchor DR is required to be an RP in the overlay.
- An extra loopback is required for anchor DRs.

- Non-TRM and Layer 2 TRM mode edge devices (VTEPs) require an IGMP snooping querier configured per multicast-enabled VLAN. Every non-TRM and Layer 2 TRM mode edge device (VTEP) requires this IGMP snooping querier configuration because in TRM multicast control-packets are not forwarded over VXLAN.
- The IP address for the IGMP snooping querier can be re-used on non-TRM and Layer 2 TRM mode edge devices (VTEPs).
- The IP address of the IGMP snooping querier in a VPC domain must be different on each VPC member device.
- When interface NVE is brought down on the border leaf, the internal overlay RP per VRF should be brought down.
- The NVE interface must be shut and unshut while configuring the **ip multicast overlay-distributed-dr** command.
- Beginning with Cisco NX-OS Release 9.2(1), TRM with vPC border leafs is supported. Advertise-PIP and Advertise Virtual-Rmac need to be enabled on border leafs to support with functionality. For configuring advertise-pip and advertise virtual-rmac, see the "Configuring VIP/PIP" section.
- Anchor DR is supported only on the following hardware platforms:
  - Cisco Nexus 9200, 9300-EX, and 9300-FX/FX2 platform switches
  - Cisco Nexus 9500 platform switches with 9700-EX line cards, 9700-FX line cards, or a combination of both line cards

## 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)<br>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

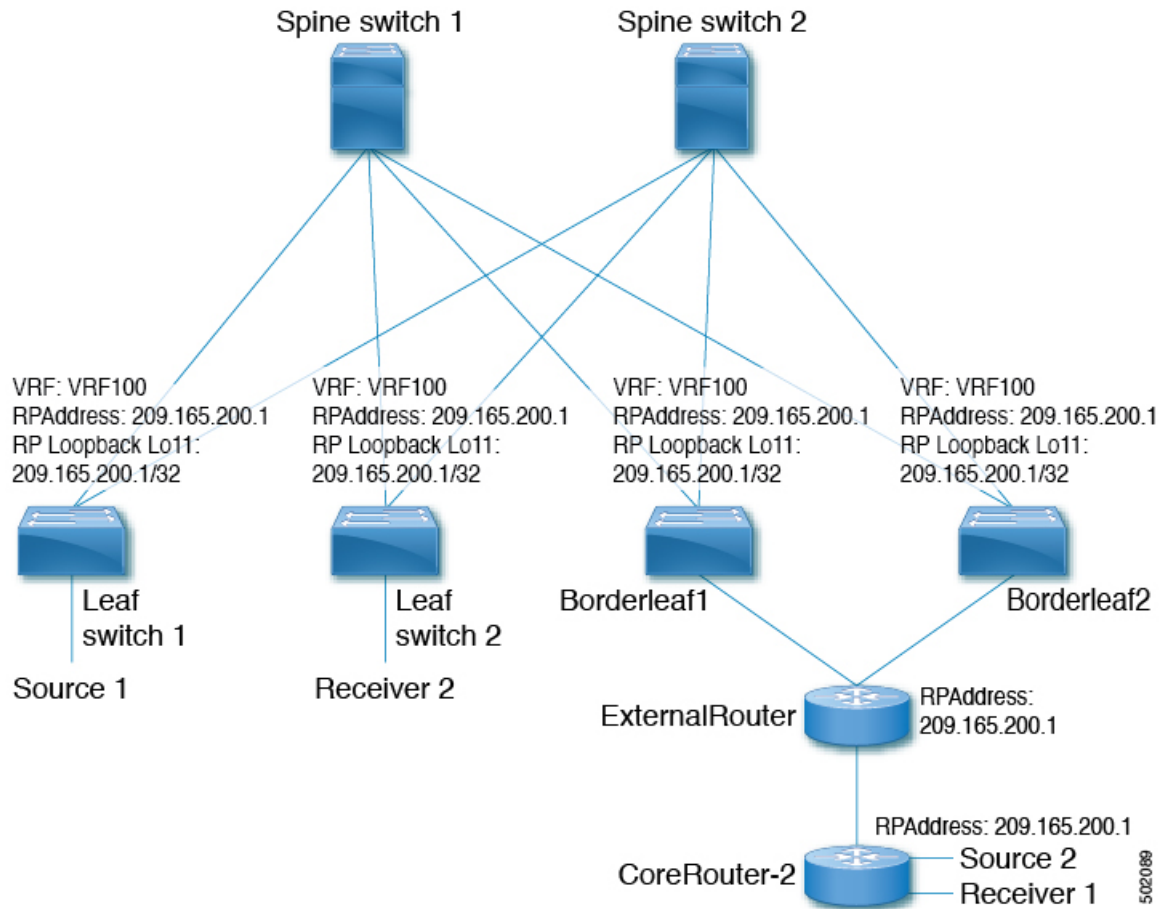
For Tenant Routed Multicast, the following rendezvous point options are supported:

- [Configuring a Rendezvous Point Inside the VXLAN Fabric, on page 170](#)

- [Configuring an External Rendezvous Point, on page 171](#)

# Configuring a Rendezvous Point Inside the VXLAN Fabric

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



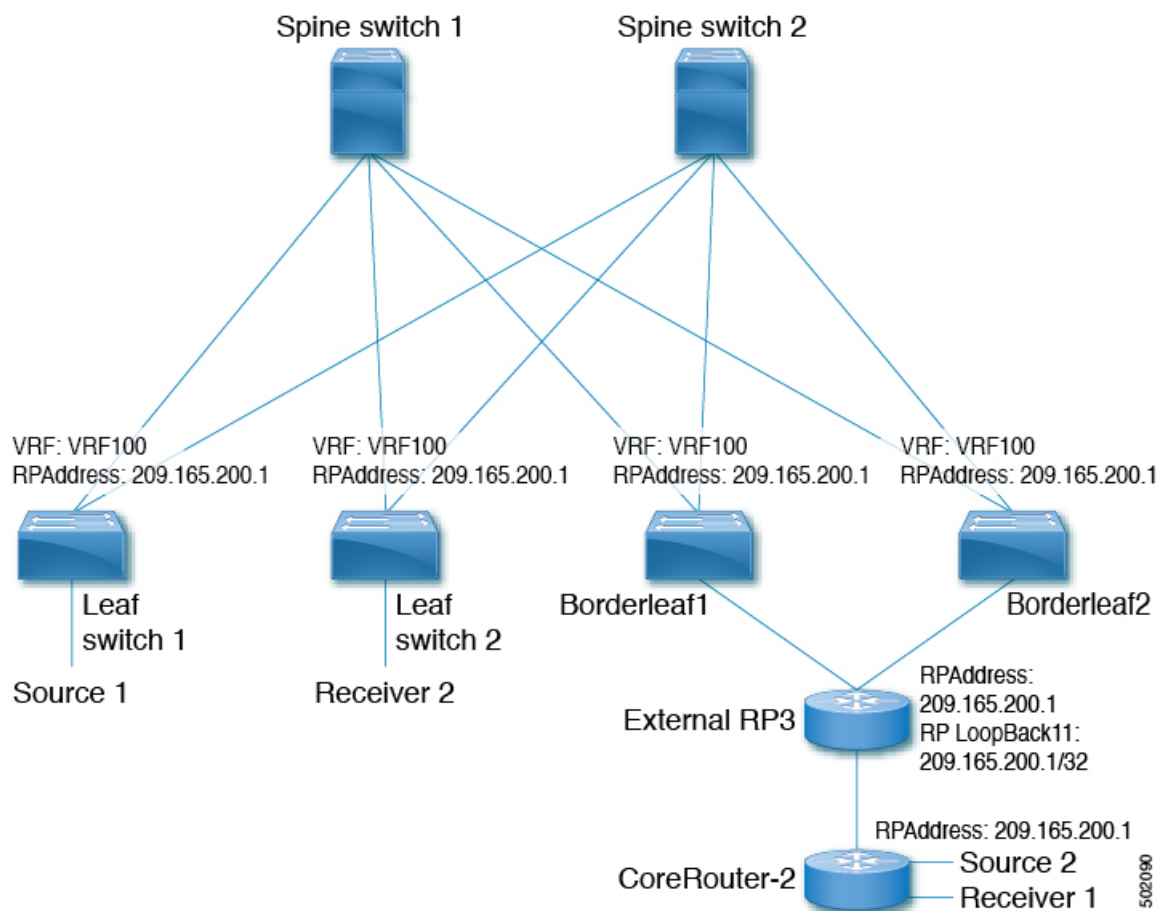
## Procedure

|               | Command or Action                                                                                                         | Purpose                                                                                                         |
|---------------|---------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|
| <b>Step 1</b> | <b>configure terminal</b><br><b>Example:</b><br><code>switch# configure terminal</code>                                   | Enters global configuration mode.                                                                               |
| <b>Step 2</b> | <b>interface loopback <i>loopback_number</i></b><br><b>Example:</b><br><code>switch(config)# interface loopback 11</code> | Configure the loopback interface on all TRM-enabled nodes. This enables the rendezvous point inside the fabric. |

|               | Command or Action                                                                                                                                                                                                     | Purpose                                                                                                                                                           |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 3</b> | <b>vrf member</b> <i>vxlan-number</i><br><br><b>Example:</b><br><code>switch(config-if)# vrf member vrf100</code>                                                                                                     | Configure VRF name.                                                                                                                                               |
| <b>Step 4</b> | <b>ip address</b> <i>ip-address</i><br><br><b>Example:</b><br><code>switch(config-if)# ip address 209.165.200.1/32</code>                                                                                             | Specify IP address.                                                                                                                                               |
| <b>Step 5</b> | <b>ip pim sparse-mode</b><br><br><b>Example:</b><br><code>switch(config-if)# ip pim sparse-mode</code>                                                                                                                | Configure sparse-mode PIM on an interface.                                                                                                                        |
| <b>Step 6</b> | <b>vrf context</b> <i>vrf-name</i><br><br><b>Example:</b><br><code>switch(config-if)# vrf context vrf100</code>                                                                                                       | Create a VXLAN tenant VRF.                                                                                                                                        |
| <b>Step 7</b> | <b>ip pim rp-address</b> <i>ip-address-of-router</i><br><b>group-list</b> <i>group-range-prefix</i><br><br><b>Example:</b><br><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. |

## Configuring an External Rendezvous Point

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



### Procedure

|               | Command or Action                                                                                                                                                                       | Purpose                                                                                                                                                              |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 1</b> | <b>configure terminal</b><br><b>Example:</b><br><pre>switch# configure terminal</pre>                                                                                                   | Enter configuration mode.                                                                                                                                            |
| <b>Step 2</b> | <b>vrf context vrf100</b><br><b>Example:</b><br><pre>switch(config)# vrf context vrf100</pre>                                                                                           | Enter configuration mode.                                                                                                                                            |
| <b>Step 3</b> | <b>ip pim rp-address ip-address-of-router group-list group-range-prefix</b><br><b>Example:</b><br><pre>switch(config-vrf)# ip pim rp-address 209.165.200.1 group-list 224.0.0.0/4</pre> | 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 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                                                                                                                |
|---------------|-------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------|
| <b>Step 1</b> | <b>configure terminal</b><br><b>Example:</b><br>switch# <b>configure terminal</b>                                             | Enter configuration mode.                                                                                              |
| <b>Step 2</b> | <b>feature ngmvpn</b><br><b>Example:</b><br>switch(config)# <b>feature ngmvpn</b>                                             | Enables the Next-Generation Multicast VPN (ngMVPN) control plane. New address family commands become available in BGP. |
| <b>Step 3</b> | <b>ip igmp snooping vxlan</b><br><b>Example:</b><br>switch(config)# <b>ip igmp snooping vxlan</b>                             | Configure IGMP snooping for VXLAN VLANs.                                                                               |
| <b>Step 4</b> | <b>interface nve1</b><br><b>Example:</b><br>switch(config)# <b>interface nve 1</b>                                            | Configure the NVE interface.                                                                                           |
| <b>Step 5</b> | <b>member vni vni-range associate-vrf</b><br><b>Example:</b><br>switch(config-if-nve)# <b>member vni 200100 associate-vrf</b> | Configure the Layer 3 virtual network identifier. The range of <i>vni-range</i> is from 1 to 16,777,214.               |
| <b>Step 6</b> | <b>mcast-group ip-prefix</b><br><b>Example:</b>                                                                               | 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><b>Note</b> We recommend that underlay multicast groups for Layer 2 VNI, default MDT, and data MDT not be shared. Use separate, non-overlapping groups.</p> |
| <b>Step 7</b>  | <pre>exit</pre> <p><b>Example:</b></p> <pre>switch(config-if-nve-vni) # exit</pre>                                               | Exits command mode.                                                                                                                                                                                                                                                                         |
| <b>Step 8</b>  | <pre>exit</pre> <p><b>Example:</b></p> <pre>switch(config-if) # exit</pre>                                                       | Exits command mode.                                                                                                                                                                                                                                                                         |
| <b>Step 9</b>  | <pre>router bgp &lt;as-number&gt;</pre> <p><b>Example:</b></p> <pre>switch(config) # router bgp 100</pre>                        | Set autonomous system number.                                                                                                                                                                                                                                                               |
| <b>Step 10</b> | <pre>neighbor ip-addr</pre> <p><b>Example:</b></p> <pre>switch(config-router) # neighbor 1.1.1.1</pre>                           | Configure IP address of the neighbor.                                                                                                                                                                                                                                                       |
| <b>Step 11</b> | <pre>address-family ipv4 mvpn</pre> <p><b>Example:</b></p> <pre>switch(config-router-neighbor) # address-family ipv4 mvpn</pre>  | Configure multicast VPN.                                                                                                                                                                                                                                                                    |
| <b>Step 12</b> | <pre>send-community extended</pre> <p><b>Example:</b></p> <pre>switch(config-router-neighbor-af) # send-community extended</pre> | Enables ngMVPN for address family signalization. The <b>send community extended</b> command ensures that extended communities are exchanged for this address family.                                                                                                                        |
| <b>Step 13</b> | <pre>exit</pre> <p><b>Example:</b></p> <pre>switch(config-router-neighbor-af) # exit</pre>                                       | Exits command mode.                                                                                                                                                                                                                                                                         |
| <b>Step 14</b> | <pre>exit</pre> <p><b>Example:</b></p> <pre>switch(config-router) # exit</pre>                                                   | Exits command mode.                                                                                                                                                                                                                                                                         |

|                | Command or Action                                                                                                                                                                                                | Purpose                                                                                                                                                                                                                                                                                                                       |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 15</b> | <b>vrf context</b> <i>vrf_name</i><br><b>Example:</b><br><pre>switch(config-router) # vrf context vrf100</pre>                                                                                                   | Configures VRF name.                                                                                                                                                                                                                                                                                                          |
| <b>Step 16</b> | <b>ip pim rp-address</b> <i>ip-address-of-router</i><br><b>group-list</b> <i>group-range-prefix</i><br><b>Example:</b><br><pre>switch(config-vrf) # ip pim rp-address 209.165.201.1 group-list 226.0.0.0/8</pre> | <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 <a href="#">Configuring a Rendezvous Point for Tenant Routed Multicast, on page 169</a> section.</p> |
| <b>Step 17</b> | <b>address-family ipv4 unicast</b><br><b>Example:</b><br><pre>switch(config-vrf) # address-family ipv4 unicast</pre>                                                                                             | Configure unicast address family.                                                                                                                                                                                                                                                                                             |
| <b>Step 18</b> | <b>route-target both auto mvpn</b><br><b>Example:</b><br><pre>switch(config-vrf-af-ipv4) # route-target both auto mvpn</pre>                                                                                     | <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>                                                      |
| <b>Step 19</b> | <b>ip multicast overlay-spt-only</b><br><b>Example:</b><br><pre>switch(config) # ip multicast overlay-spt-only</pre>                                                                                             | Gratuitously originate (S,A) route when the source is locally connected. The <b>ip multicast overlay-spt-only</b> command is enabled by default on all MVPN-enabled Cisco Nexus 9000 Series switches (typically leaf node).                                                                                                   |
| <b>Step 20</b> | <b>interface</b> <i>vlan_id</i><br><b>Example:</b><br><pre>switch(config) # interface vlan11</pre>                                                                                                               | Configures the first-hop gateway (distributed anycast gateway for the Layer 2 VNI. No router PIM peering must ever happen with this interface.                                                                                                                                                                                |
| <b>Step 21</b> | <b>no shutdown</b><br><b>Example:</b><br><pre>switch(config-if) # no shutdown</pre>                                                                                                                              | Disables an interface.                                                                                                                                                                                                                                                                                                        |
| <b>Step 22</b> | <b>vrf member</b> <i>vrf-num</i><br><b>Example:</b><br><pre>switch(config-if) # vrf member vrf100</pre>                                                                                                          | Configure VRF name.                                                                                                                                                                                                                                                                                                           |
| <b>Step 23</b> | <b>ipv6 address</b> <i>ipv6_address</i><br><b>Example:</b>                                                                                                                                                       | Configure IP address.                                                                                                                                                                                                                                                                                                         |

|                | Command or Action                                                                                                                          | Purpose                                                                                                                                                                                                                                                                                                                       |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | <code>switch(config-if)# ip address 11.1.1.1/24</code>                                                                                     |                                                                                                                                                                                                                                                                                                                               |
| <b>Step 24</b> | <b>ipv6 pim sparse-mode</b><br><b>Example:</b><br><code>switch(config-if)# ip pim sparse-mode</code>                                       | Enables IGMP and PIM on the SVI. This is required if multicast sources and/or receivers exist in this VLAN.                                                                                                                                                                                                                   |
| <b>Step 25</b> | <b>fabric forwarding mode anycast-gateway</b><br><b>Example:</b><br><code>switch(config-if)# fabric forwarding mode anycast-gateway</code> | Configure Anycast Gateway Forwarding Mode.                                                                                                                                                                                                                                                                                    |
| <b>Step 26</b> | <b>ip pim neighbor-policy NONE*</b><br><b>Example:</b><br><code>switch(config-if)# ip pim neighbor-policy NONE*</code>                     | <p>Creates an IP PIM neighbor policy to avoid PIM neighborship with PIM routers within the VLAN. The <b>none</b> keyword is a configured route map to deny any ipv4 addresses to avoid establishing PIM neighborship policy using anycast IP.</p> <p><b>Note</b> Do not use Distributed Anycast Gateway for PIM Peerings.</p> |
| <b>Step 27</b> | <b>exit</b><br><b>Example:</b><br><code>switch(config-if)# exit</code>                                                                     | Exits command mode.                                                                                                                                                                                                                                                                                                           |
| <b>Step 28</b> | <b>interface vlan_id</b><br><b>Example:</b><br><code>switch(config)# interface vlan100</code>                                              | Configure Layer 3 VNI.                                                                                                                                                                                                                                                                                                        |
| <b>Step 29</b> | <b>no shutdown</b><br><b>Example:</b><br><code>switch(config-if)# no shutdown</code>                                                       | Disable an interface.                                                                                                                                                                                                                                                                                                         |
| <b>Step 30</b> | <b>vrf member vrf100</b><br><b>Example:</b><br><code>switch(config-if)# vrf member vrf100</code>                                           | Configure VRF name.                                                                                                                                                                                                                                                                                                           |
| <b>Step 31</b> | <b>ip forward</b><br><b>Example:</b><br><code>switch(config-if)# ip forward</code>                                                         | Enable IP forwarding on interface.                                                                                                                                                                                                                                                                                            |
| <b>Step 32</b> | <b>ip pim sparse-mode</b><br><b>Example:</b><br><code>switch(config-if)# ip pim sparse-mode</code>                                         | Configure sparse-mode PIM on interface. There is no PIM peering happening in the Layer-3 VNI, but this command must be present for forwarding.                                                                                                                                                                                |

# Configuring TRM on the VXLAN EVPN Spine

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

## Before you begin

The VXLAN BGP EVPN spine must be configured. See [Configuring iBGP for EVPN on the Spine](#), on page 77.

## Procedure

|               | Command or Action                                                                                                     | Purpose                                                                                                                                                                                                  |
|---------------|-----------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 1</b> | <b>configure terminal</b><br><br><b>Example:</b><br>switch# <b>configure terminal</b>                                 | Enter configuration mode.                                                                                                                                                                                |
| <b>Step 2</b> | <b>route-map permitall permit 10</b><br><br><b>Example:</b><br>switch(config)# <b>route-map permitall permit 10</b>   | Configure the route-map.<br><br><b>Note</b> The route-map keeps the next-hop unchanged for EVPN routes <ul style="list-style-type: none"> <li>• Required for eBGP</li> <li>• Options for iBGP</li> </ul> |
| <b>Step 3</b> | <b>set ip next-hop unchanged</b><br><br><b>Example:</b><br>switch(config-route-map)# <b>set ip next-hop unchanged</b> | Set next hop address.<br><br><b>Note</b> The route-map keeps the next-hop unchanged for EVPN routes <ul style="list-style-type: none"> <li>• Required for eBGP</li> <li>• Options for iBGP</li> </ul>    |
| <b>Step 4</b> | <b>exit</b><br><br><b>Example:</b><br>switch(config-route-map)# <b>exit</b>                                           | Return to exec mode.                                                                                                                                                                                     |
| <b>Step 5</b> | <b>router bgp [autonomous system] number</b><br><br><b>Example:</b><br>switch(config)# <b>router bgp 65002</b>        | Specify BGP.                                                                                                                                                                                             |
| <b>Step 6</b> | <b>address-family ipv4 mvpn</b><br><br><b>Example:</b><br>switch(config-router)# <b>address-family ipv4 mvpn</b>      | Configure the address family IPv4 MVPN under the BGP.                                                                                                                                                    |

|                | Command or Action                                                                                                                | Purpose                                                                                                                                                                                                                                                    |
|----------------|----------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 7</b>  | <b>retain route-target all</b><br><b>Example:</b><br><pre>switch(config-router-af) # retain route-target all</pre>               | Configure retain route-target all under address-family IPv4 MVPN [global].<br><br><b>Note</b> 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.      |
| <b>Step 8</b>  | <b>neighbor ip-address [remote-as number]</b><br><b>Example:</b><br><pre>switch(config-router-af) # neighbor 100.100.100.1</pre> | Define neighbor.                                                                                                                                                                                                                                           |
| <b>Step 9</b>  | <b>address-family ipv4 mvpn</b><br><b>Example:</b><br><pre>switch(config-router-neighbor) # address-family ipv4 mvpn</pre>       | Configure address family IPv4 MVPN under the BGP neighbor.                                                                                                                                                                                                 |
| <b>Step 10</b> | <b>disable-peer-as-check</b><br><b>Example:</b><br><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.<br><br><b>Note</b> Required for eBGP.                  |
| <b>Step 11</b> | <b>rewrite-rt-asn</b><br><b>Example:</b><br><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 <b>rewrite-rt-asn</b> command is required if the route target auto feature is being used to configure EVPN route targets. |
| <b>Step 12</b> | <b>send-community extended</b><br><b>Example:</b><br><pre>switch(config-router-neighbor-af) # send-community extended</pre>      | Configures community for BGP neighbors.                                                                                                                                                                                                                    |
| <b>Step 13</b> | <b>route-reflector-client</b><br><b>Example:</b><br><pre>switch(config-router-neighbor-af) # route-reflector-client</pre>        | Configure route reflector.<br><br><b>Note</b> Required for iBGP with route-reflector.                                                                                                                                                                      |
| <b>Step 14</b> | <b>route-map permitall out</b><br><b>Example:</b><br><pre>switch(config-router-neighbor-af) # route-map permitall out</pre>      | Applies route-map to keep the next-hop unchanged.<br><br><b>Note</b> 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 Cisco Nexus 9000 Series switches (1st generation). Only the Cisco Nexus 9000-EX and 9000-FX switches can do Layer 2/Layer 3 mode (Anchor-DR).

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

All Cisco Nexus 9300-EX and 9300-FX platform switches must be in Layer 2/Layer 3 mode.

## Before you begin

VXLAN EVPN must be configured.

The rendezvous point (RP) must be configured.

## Procedure

|               | Command or Action                                                                                               | Purpose                                                                                                                                                                                                                  |
|---------------|-----------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 1</b> | <b>configure terminal</b><br><b>Example:</b><br>switch# <b>configure terminal</b>                               | Enter configuration mode.                                                                                                                                                                                                |
| <b>Step 2</b> | <b>feature ngmvpn</b><br><b>Example:</b><br>switch(config)# <b>feature ngmvpn</b>                               | Enables the Next-Generation Multicast VPN (ngMVPN) control plane. New address family commands become available in BGP.                                                                                                   |
| <b>Step 3</b> | <b>advertise evpn multicast</b><br><b>Example:</b><br>switch(config)# <b>advertise evpn multicast</b>           | Advertises IMET and SMET routes into BGP EVPN towards non-TRM capable switches.                                                                                                                                          |
| <b>Step 4</b> | <b>ip igmp snooping vxlan</b><br><b>Example:</b><br>switch(config)# <b>ip igmp snooping vxlan</b>               | Configure IGMP snooping for VXLAN VLANs.                                                                                                                                                                                 |
| <b>Step 5</b> | <b>ip multicast overlay-spt-only</b><br><b>Example:</b><br>switch(config)# <b>ip multicast overlay-spt-only</b> | Gratuitously originate (S,A) route when source is locally connected. The <b>ip multicast overlay-spt-only</b> command is enabled by default on all MVPN-enabled Cisco Nexus 9000 Series switches (typically leaf nodes). |
| <b>Step 6</b> | <b>ip multicast overlay-distributed-dr</b><br><b>Example:</b>                                                   | Enables distributed anchor DR function on this VTEP.                                                                                                                                                                     |

|                | Command or Action                                                                                                                    | Purpose                                                                                                  |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|
|                | <code>switch(config)# ip multicast overlay-distributed-dr</code>                                                                     | <b>Note</b> The NVE interface must be shut and unshut while configuring this command.                    |
| <b>Step 7</b>  | <b>interface nve1</b><br><b>Example:</b><br><code>switch(config)# interface nve 1</code>                                             | Configure the NVE interface.                                                                             |
| <b>Step 8</b>  | <b>[no] shutdown</b><br><b>Example:</b><br><code>switch(config-if-nve)# shutdown</code>                                              | Shuts down the NVE interface. The <b>no shutdown</b> command brings up the interface.                    |
| <b>Step 9</b>  | <b>member vni vni-range associate-vrf</b><br><b>Example:</b><br><code>switch(config-if-nve)# member vni 200100 associate-vrf</code>  | Configure the Layer 3 virtual network identifier. The range of <i>vni-range</i> is from 1 to 16,777,214. |
| <b>Step 10</b> | <b>mcast-group ip-prefix</b><br><b>Example:</b><br><code>switch(config-if-nve-vni)# mcast-group 225.3.3.3</code>                     | Configures the multicast group on distributed anchor DR.                                                 |
| <b>Step 11</b> | <b>exit</b><br><b>Example:</b><br><code>switch(config-if-nve-vni)# exit</code>                                                       | Exits command mode.                                                                                      |
| <b>Step 12</b> | <b>interface loopback loopback_number</b><br><b>Example:</b><br><code>switch(config-if-nve)# interface loopback 10</code>            | Configure the loopback interface on all distributed anchor DR devices.                                   |
| <b>Step 13</b> | <b>ip address ip_address</b><br><b>Example:</b><br><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.                          |
| <b>Step 14</b> | <b>ip router ospf process-tag area ospf-id</b><br><b>Example:</b><br><code>switch(config-if)# ip router ospf 100 area 0.0.0.0</code> | OSPF area ID in IP address format.                                                                       |
| <b>Step 15</b> | <b>ip pim sparse-mode</b><br><b>Example:</b><br><code>switch(config-if)# ip pim sparse-mode</code>                                   | Configure sparse-mode PIM on interface.                                                                  |



|                | Command or Action                                                                                                                                                       | Purpose                                                                                                                                                                                                                                                                                                                                                               |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 16</b> | <b>interface nve1</b><br><b>Example:</b><br>switch(config-if)# <b>interface nve1</b>                                                                                    | Configure NVE interface.                                                                                                                                                                                                                                                                                                                                              |
| <b>Step 17</b> | <b>shutdown</b><br><b>Example:</b><br>switch(config-if-nve)# <b>shutdown</b>                                                                                            | Disable the interface.                                                                                                                                                                                                                                                                                                                                                |
| <b>Step 18</b> | <b>mcast-routing override source-interface loopback int-num</b><br><b>Example:</b><br>switch(config-if-nve)# <b>mcast-routing override source-interface loopback 10</b> | Enables that TRM is using a different loopback interface than the VTEPs default source-interface.<br><br>The <i>loopback10</i> variable must be configured on every TRM-enabled VTEP (Anchor DR) in the underlay with the same IP address. This loopback and the respective <b>override</b> command are needed to serve TRM VTEPs in co-existence with non-TRM VTEPs. |
| <b>Step 19</b> | <b>exit</b><br><b>Example:</b><br>switch(config-if-nve)# <b>exit</b>                                                                                                    | Exits command mode.                                                                                                                                                                                                                                                                                                                                                   |
| <b>Step 20</b> | <b>router bgp 100</b><br><b>Example:</b><br>switch(config)# <b>router bgp 100</b>                                                                                       | Set autonomous system number.                                                                                                                                                                                                                                                                                                                                         |
| <b>Step 21</b> | <b>neighbor ip-addr</b><br><b>Example:</b><br>switch(config-router)# <b>neighbor 1.1.1.1</b>                                                                            | Configure IP address of the neighbor.                                                                                                                                                                                                                                                                                                                                 |
| <b>Step 22</b> | <b>address-family ipv4 mvpn</b><br><b>Example:</b><br>switch(config-router-neighbor)# <b>address-family ipv4 mvpn</b>                                                   | Configure multicast VPN.                                                                                                                                                                                                                                                                                                                                              |
| <b>Step 23</b> | <b>send-community extended</b><br><b>Example:</b><br>switch(config-router-neighbor-af)# <b>send-community extended</b>                                                  | Send community attribute.                                                                                                                                                                                                                                                                                                                                             |
| <b>Step 24</b> | <b>exit</b><br><b>Example:</b><br>switch(config-router-neighbor-af)# <b>exit</b>                                                                                        | Exits command mode.                                                                                                                                                                                                                                                                                                                                                   |

|                | Command or Action                                                                                                                                                                              | Purpose                                                                                                                                                                                                                                                                                                                              |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 25</b> | <b>exit</b><br><br><b>Example:</b><br><code>switch(config-router) # exit</code>                                                                                                                | Exits command mode.                                                                                                                                                                                                                                                                                                                  |
| <b>Step 26</b> | <b>vrf vrf_name vrf100</b><br><br><b>Example:</b><br><code>switch(config) # vrf context vrf100</code>                                                                                          | Configure VRF name.                                                                                                                                                                                                                                                                                                                  |
| <b>Step 27</b> | <b>ip pim rp-address ip-address-of-router group-list group-range-prefix</b><br><br><b>Example:</b><br><code>switch(config-vrf) # ip pim rp-address 209.165.201.1 group-list 226.0.0.0/8</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 of the edge devices (VTEPs) for a fully distributed RP.<br><br>For overlay RP placement options, see the <a href="#">Configuring a Rendezvous Point for Tenant Routed Multicast, on page 169</a> - Internal RP section. |
| <b>Step 28</b> | <b>address-family ipv4 unicast</b><br><br><b>Example:</b><br><code>switch(config-vrf) # address-family ipv4 unicast</code>                                                                     | Configure unicast address family.                                                                                                                                                                                                                                                                                                    |
| <b>Step 29</b> | <b>route-target both auto mvpn</b><br><br><b>Example:</b><br><code>switch(config-vrf-af-ipv4) # route-target both auto mvpn</code>                                                             | Specify target for mvpn routes.                                                                                                                                                                                                                                                                                                      |
| <b>Step 30</b> | <b>exit</b><br><br><b>Example:</b><br><code>switch(config-vrf-af-ipv4) # exit</code>                                                                                                           | Exits command mode.                                                                                                                                                                                                                                                                                                                  |
| <b>Step 31</b> | <b>exit</b><br><br><b>Example:</b><br><code>switch(config-vrf) # exit</code>                                                                                                                   | Exits command mode.                                                                                                                                                                                                                                                                                                                  |
| <b>Step 32</b> | <b>interface vlan_id</b><br><br><b>Example:</b><br><code>switch(config) # interface vlan11</code>                                                                                              | Configure Layer 2 VNI.                                                                                                                                                                                                                                                                                                               |
| <b>Step 33</b> | <b>no shutdown</b><br><br><b>Example:</b><br><code>switch(config-if) # no shutdown</code>                                                                                                      | Disable an interface.                                                                                                                                                                                                                                                                                                                |
| <b>Step 34</b> | <b>vrf member vrf100</b><br><br><b>Example:</b>                                                                                                                                                | Configure VRF name.                                                                                                                                                                                                                                                                                                                  |

|                | Command or Action                                                                                                                           | Purpose                                                                                                                                        |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|
|                | <code>switch(config-if) # vrf member vrf100</code>                                                                                          |                                                                                                                                                |
| <b>Step 35</b> | <b>ip address <i>ip_address</i></b><br><b>Example:</b><br><code>switch(config-if) # ip address 11.1.1.1/24</code>                           | Configure IP address.                                                                                                                          |
| <b>Step 36</b> | <b>ip pim sparse-mode</b><br><b>Example:</b><br><code>e</code><br><code>switch(config-if) # ip pim sparse-mode</code>                       | Configure sparse-mode PIM on the interface.                                                                                                    |
| <b>Step 37</b> | <b>fabric forwarding mode anycast-gateway</b><br><b>Example:</b><br><code>switch(config-if) # fabric forwarding mode anycast-gateway</code> | Configure Anycast Gateway Forwarding Mode.                                                                                                     |
| <b>Step 38</b> | <b>ip pim neighbor-policy NONE*</b><br><b>Example:</b><br><code>switch(config-if) # ip pim neighbor-policy NONE*</code>                     | The <b>none</b> keyword is a configured route map to deny any IPv4 addresses to avoid establishing a PIM neighborhood policy using anycase IP. |
| <b>Step 39</b> | <b>exit</b><br><b>Example:</b><br><code>switch(config-if) # exit</code>                                                                     | Exits command mode.                                                                                                                            |
| <b>Step 40</b> | <b>interface <i>vlan_id</i></b><br><b>Example:</b><br><code>switch(config) # interface vlan100</code>                                       | Configure Layer 3 VNI.                                                                                                                         |
| <b>Step 41</b> | <b>no shutdown</b><br><b>Example:</b><br><code>switch(config-if) # no shutdown</code>                                                       | Disable an interface.                                                                                                                          |
| <b>Step 42</b> | <b>vrf member vrf100</b><br><b>Example:</b><br><code>switch(config-if) # vrf member vrf100</code>                                           | Configure VRF name.                                                                                                                            |
| <b>Step 43</b> | <b>ip forward</b><br><b>Example:</b><br><code>switch(config-if) # ip forward</code>                                                         | Enable IP forwarding on interface.                                                                                                             |
| <b>Step 44</b> | <b>ip pim sparse-mode</b><br><b>Example:</b><br><code>switch(config-if) # ip pim sparse-mode</code>                                         | Configure sparse-mode PIM on the interface.                                                                                                    |

## Configuring Layer 2 Tenant Routed Multicast

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

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

### Before you begin

VXLAN EVPN must be configured.

### Procedure

|               | Command or Action                                                                                                                                                 | Purpose                                                                |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------|
| <b>Step 1</b> | <b>configure terminal</b><br><br><b>Example:</b><br><code>switch# configure terminal</code>                                                                       | Enter configuration mode.                                              |
| <b>Step 2</b> | <b>feature ngmvpn</b><br><br><b>Example:</b><br><code>switch(config)# feature ngmvpn</code>                                                                       | Enables EVPN/MVPN feature.                                             |
| <b>Step 3</b> | <b>advertise evpn multicast</b><br><br><b>Example:</b><br><code>switch(config)# advertise evpn multicast</code>                                                   | Advertise L2 multicast capability.                                     |
| <b>Step 4</b> | <b>ip igmp snooping vxlan</b><br><br><b>Example:</b><br><code>switch(config)# ip igmp snooping vxlan</code>                                                       | Configure IGMP snooping for VXLANs.                                    |
| <b>Step 5</b> | <b>vlan configuration <i>vlan-id</i></b><br><br><b>Example:</b><br><code>switch(config)# vlan configuration 101</code>                                            | Enter configuration mode for VLAN 101.                                 |
| <b>Step 6</b> | <b>ip igmp snooping querier <i>querier-ip-address</i></b><br><br><b>Example:</b><br><code>switch(config-vlan-config)# ip igmp<br/>snooping querier 2.2.2.2</code> | Configure IGMP snooping querier for each multicast-enabled VXLAN VLAN. |

## Configuring TRM with vPC Support

This section provides steps to configure TRM with vPC support. Beginning with Cisco NX-OS Release 10.1(2), TRM Multisite with vPC BGW is supported.

## Procedure

|                | Command or Action                                                                                                                                                 | Purpose                                                                                                                                                   |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 1</b>  | <b>configure terminal</b><br><b>Example:</b><br>switch# <b>configure terminal</b>                                                                                 | Enter global configuration mode.                                                                                                                          |
| <b>Step 2</b>  | <b>feature vpc</b><br><b>Example:</b><br>switch(config)# <b>feature vpc</b>                                                                                       | Enables vPCs on the device.                                                                                                                               |
| <b>Step 3</b>  | <b>feature interface-vlan</b><br><b>Example:</b><br>switch(config)# <b>feature interface-vlan</b>                                                                 | Enables the interface VLAN feature on the device.                                                                                                         |
| <b>Step 4</b>  | <b>feature lacp</b><br><b>Example:</b><br>switch(config)# <b>feature lacp</b>                                                                                     | Enables the LACP feature on the device.                                                                                                                   |
| <b>Step 5</b>  | <b>feature pim</b><br><b>Example:</b><br>switch(config)# <b>feature pim</b>                                                                                       | Enables the PIM feature on the device.                                                                                                                    |
| <b>Step 6</b>  | <b>feature ospf</b><br><b>Example:</b><br>switch(config)# <b>feature ospf</b>                                                                                     | Enables the OSPF feature on the device.                                                                                                                   |
| <b>Step 7</b>  | <b>ip pim rp-address <i>address</i> group-list <i>range</i></b><br><b>Example:</b><br>switch(config)# <b>ip pim rp-address 100.100.100.1 group-list 224.0.0/4</b> | Defines a PIM RP address for the underlay multicast group range.                                                                                          |
| <b>Step 8</b>  | <b>vpc domain <i>domain-id</i></b><br><b>Example:</b><br>switch(config)# <b>vpc domain 1</b>                                                                      | Creates a vPC domain on the device and enters vpn-domain configuration mode for configuration purposes. There is no default. The range is from 1 to 1000. |
| <b>Step 9</b>  | <b>peer switch</b><br><b>Example:</b><br>switch(config-vpc-domain)# <b>peer switch</b>                                                                            | Defines the peer switch.                                                                                                                                  |
| <b>Step 10</b> | <b>peer gateway</b><br><b>Example:</b><br>switch(config-vpc-domain)# <b>peer gateway</b>                                                                          | To enable Layer 3 forwarding for packets destined to the gateway MAC address of the virtual port channel (vPC), use the <b>peer-gateway</b> command.      |

|                | Command or Action                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Purpose                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 11</b> | <b>peer-keepalive destination</b> <i>ipaddress</i><br><b>Example:</b><br><pre>switch(config-vpc-domain)# peer-keepalive destination 172.28.230.85</pre>                                                                                                                                                                                                                                                                                                                                      | <p>Configures the IPv4 address for the remote end of the vPC peer-keepalive link.</p> <p><b>Note</b> The system does not form the vPC peer link until you configure a vPC peer-keepalive link.</p> <p>The management ports and VRF are the defaults.</p> <p><b>Note</b> We recommend that you configure a separate VRF and use a Layer 3 port from each vPC peer device in that VRF for the vPC peer-keepalive link.</p> <p>For more information about creating and configuring VRFs, see the <a href="#">Cisco Nexus 9000 NX-OS Series Unicast Routing Config Guide, 9.3(x)</a>.</p> |
| <b>Step 12</b> | <b>ip arp synchronize</b><br><b>Example:</b><br><pre>switch(config-vpc-domain)# ip arp synchronize</pre>                                                                                                                                                                                                                                                                                                                                                                                     | Enables IP ARP synchronize under the vPC Domain to facilitate faster ARP table population following device reload.                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>Step 13</b> | <b>ipv6 nd synchronize</b><br><b>Example:</b><br><pre>switch(config-vpc-domain)# ipv6 nd synchronize</pre>                                                                                                                                                                                                                                                                                                                                                                                   | Enables IPv6 nd synchronization under the vPC domain to facilitate faster nd table population following device reload.                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>Step 14</b> | <p>Create vPC peer-link.</p> <p><b>Example:</b></p> <pre>switch(config)# interface port-channel 1 switch(config)# switchport switch(config)# switchport mode trunk switch(config)# switchport trunk allowed vlan 1,10,100-200 switch(config)# mtu 9216 switch(config)# vpc peer-link switch(config)# no shut  switch(config)# interface Ethernet 1/1, 1/21 switch(config)# switchport switch(config)# mtu 9216 switch(config)# channel-group 1 mode active switch(config)# no shutdown</pre> | Creates the vPC peer-link port-channel interface and adds two member interfaces to it.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |

|                | Command or Action                                                                                                                                                                                                                                                                                          | Purpose                                                                                                                                                                                                                                |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 15</b> | <b>system nve infra-vlans</b> <i>range</i><br><b>Example:</b><br><pre>switch(config)# system nve infra-vlans 10</pre>                                                                                                                                                                                      | Defines a non-VXLAN enabled VLAN as a backup routed path.                                                                                                                                                                              |
| <b>Step 16</b> | <b>vlan</b> <i>number</i><br><b>Example:</b><br><pre>switch(config)# vlan 10</pre>                                                                                                                                                                                                                         | Creates the VLAN to be used as an infra-VLAN.                                                                                                                                                                                          |
| <b>Step 17</b> | Create the SVI.<br><b>Example:</b><br><pre>switch(config)# interface vlan 10 switch(config)# ip address 10.10.10.1/30 switch(config)# ip router ospf process UNDERLAY area 0 switch(config)# ip pim sparse-mode switch(config)# no ip redirects switch(config)# mtu 9216 switch(config)# no shutdown</pre> | Creates the SVI used for the backup routed path over the vPC peer-link.                                                                                                                                                                |
| <b>Step 18</b> | (Optional) <b>delay restore interface-vlan</b> <i>seconds</i><br><b>Example:</b><br><pre>switch(config-vpc-domain)# delay restore interface-vlan 45</pre>                                                                                                                                                  | Enables the delay restore timer for SVIs. We recommend tuning this value when the SVI/VNI scale is high. For example, when the SCI count is 1000, we recommend that you set the delay restore for <b>interface-vlan</b> to 45 seconds. |







## CHAPTER 11

# Configuring Cross Connect

This chapter contains the following sections:

- [About VXLAN Cross Connect, on page 189](#)
- [Guidelines and Limitations for VXLAN Cross Connect, on page 190](#)
- [Configuring VXLAN Cross Connect, on page 191](#)
- [Verifying VXLAN Cross Connect Configuration, on page 193](#)
- [Configuring NGOAM for VXLAN Cross Connect, on page 194](#)
- [Verifying NGOAM for VXLAN Cross Connect, on page 194](#)
- [NGOAM Authentication, on page 195](#)
- [Guidelines and Limitations for Q-in-VNI, on page 196](#)
- [Configuring Q-in-VNI, on page 198](#)
- [Configuring Selective Q-in-VNI, on page 199](#)
- [Configuring Q-in-VNI with LACP Tunneling, on page 201](#)
- [Selective Q-in-VNI with Multiple Provider VLANs, on page 204](#)
- [Configuring QinQ-QinVNI, on page 207](#)
- [Removing a VNI, on page 209](#)

## About VXLAN Cross Connect

This feature provides point-to-point tunneling of data and control packet from one VTEP to another. Every attachment circuit will be part of a unique provider VNI. BGP EVPN signaling will discover these end-points based on how the provider VNI is stretched in the fabric. All inner customer .1q tags will be preserved, as is, and packets will be encapsulated in the provider VNI at the encapsulation VTEP. On the decapsulation end-point, the provider VNI will forward the packet to its attachment circuit while preserving all customer .1q tags in the packets.



---

**Note** Cross Connect and xconnect are synonymous.

---

Beginning with Cisco NX-OS Release 9.2(3), support added for vPC Fabric Peering.

VXLAN Cross Connect enables VXLAN point-to-point functionality on the following switches:

- Cisco Nexus 9332PQ
- Cisco Nexus 9336C-FX2

- Cisco Nexus 9372PX
- Cisco Nexus 9372PX-E
- Cisco Nexus 9372TX
- Cisco Nexus 9372TX-E
- Cisco Nexus 93120TX
- Cisco Nexus 93108TC-EX
- Cisco Nexus 93108TC-FX
- Cisco Nexus 93180LC-EX
- Cisco Nexus 93180YC-EX
- Cisco Nexus 93180YC-FX
- Cisco Nexus 93240YC-FX2

VXLAN Cross Connect enables tunneling of all control frames (CDP, LLDP, LACP, STP, BFD, and PAGP) and data across the VXLAN cloud.

## Guidelines and Limitations for VXLAN Cross Connect

VXLAN Cross Connect has the following guidelines and limitations:

- When an upgrade is performed non-disruptively from Cisco NX-OS Release 7.0(3)I7(4) to Cisco NX-OS Release 9.2(x) code, and if a VLAN is created and configured as xconnect, you must enter the **copy running-config startup-config** command and reload the switch. If the box was upgraded disruptively to Cisco NX-OS Release 9.2(x) code, a reload is not needed on configuring a VLAN as xconnect.
- MAC learning will be disabled on the xconnect VNIs and none of the host MAC will be learned on the tunnel access ports.
- Only supported on a BGP EVPN topology.
- LACP bundling of attachment circuits is not supported.
- Only one attachment circuit can be configured for a provider VNI on a given VTEP.
- A VNI can only be stretched in a point-to-point fashion. Point-to-multipoint is not supported.
- SVI on an xconnect VLAN is not supported.
- ARP suppression is not supported on an xconnect VLAN VNI.
- Xconnect is not supported on the following switches:
  - Cisco Nexus 9504
  - Cisco Nexus 9508
  - Cisco Nexus 9516

- Scale of xconnect VLANs depends on the number of ports available on the switch. Every xconnect VLAN can tunnel all 4k customer VLANs.
- Xconnect or Crossconnect feature on vpc-vtep needs backup-svi as native VLAN on the vPC peer-link.
- Make sure that the NGOAM xconnect hb-interval is set to 5000 milliseconds on all VTEPs before attempting ISSU/patch activation to avoid link flaps.
- Before activating the patch for the cfs process, you must move the NGOAM xconnect hb-interval to the maximum value of 5000 milliseconds. This prevents interface flaps during the patch activation.
- The vPC orphan tunneled port per VNI should be either on the vPC primary switch or secondary switch, but not both.
- Configuring a static MAC on xconnect tunnel interfaces is not supported.
- xconnect is not supported on FEX ports.
- On vpc-vtep, spanning tree must be disabled on both vPC peers for xconnect VLANs.
- Xconnect access ports need to be flapped after disabling NGOAM on all the VTEPs.
- After deleting and adding a VLAN, or removing xconnect from a VLAN, physical ports need to be flapped with NGOAM.
- VXLAN Cross Connect is not supported as part of multi-site solution.

## Configuring VXLAN Cross Connect

This procedure describes how to configure the VXLAN Cross Connect feature.

### Procedure

|               | Command or Action                                                                                          | Purpose                                                                      |
|---------------|------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------|
| <b>Step 1</b> | <b>configure terminal</b><br><br><b>Example:</b><br><code>switch# configure terminal</code>                | Enters global configuration mode.                                            |
| <b>Step 2</b> | <b>vlan <i>vlan-id</i></b><br><br><b>Example:</b><br><code>switch(config)# vlan 10</code>                  | Specifies VLAN.                                                              |
| <b>Step 3</b> | <b>vn-segment <i>vnid</i></b><br><br><b>Example:</b><br><code>switch(config-vlan)# vn-segment 10010</code> | Specifies VXLAN VNID (Virtual Network Identifier).                           |
| <b>Step 4</b> | <b>xconnect</b><br><br><b>Example:</b><br><code>switch(config-vlan)# xconnect</code>                       | Defines the provider VLAN with the attached VNI to be in cross connect mode. |

|               | Command or Action                                                                                                    | Purpose                                                                                                                                                                                         |
|---------------|----------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 5</b> | <b>exit</b><br><br><b>Example:</b><br>switch(config-vlan)# <b>exit</b>                                               | Exits command mode.                                                                                                                                                                             |
| <b>Step 6</b> | <b>interface type port</b><br><br><b>Example:</b><br>switch(config)# <b>interface ethernet 1/1</b>                   | Enters interface configuration mode.                                                                                                                                                            |
| <b>Step 7</b> | <b>switchport mode dot1q-tunnel</b><br><br><b>Example:</b><br>switch(config-if)# <b>switchport mode dot1q-tunnel</b> | Creates a 802.1q tunnel on the port. The port will do down and reinitialize (port flap) when the interface mode is changed. BPDU filtering is enabled and CDP is disabled on tunnel interfaces. |
| <b>Step 8</b> | <b>switchport access vlan vlan-id</b><br><br><b>Example:</b><br>switch(config-if)# <b>switchport access vlan 10</b>  | Sets the interface access VLAN.                                                                                                                                                                 |
| <b>Step 9</b> | <b>exit</b><br><br><b>Example:</b><br>switch(config-vlan)# <b>exit</b>                                               | Exits command mode.                                                                                                                                                                             |

### Example

This example shows how to configure VXLAN Cross Connect.

```
switch# configure terminal
switch(config)# vlan 10
switch(config)# vn-segment 10010
switch(config)# xconnect
switch(config)# vlan 20
switch(config)# vn-segment 10020
switch(config)# xconnect
switch(config)# vlan 30
switch(config)# vn-segment 10030
switch(config)# xconnect
```

This example shows how to configure access ports:

```
switch# configure terminal
switch(config)# interface ethernet1/1
switch(config-if)# switchport mode dot1q-tunnel
switch(config-if)# switchport access vlan 10
switch(config-if)# exit
switch(config)# interface ethernet1/2
switch(config-if)# switchport mode dot1q-tunnel
switch(config-if)# switchport access vlan 20
switch(config-if)# exit
switch(config)# interface ethernet1/3
switch(config-if)# switchport mode dot1q-tunnel
```

```
switch(config-if)# switchport access vlan 30
```

## Verifying VXLAN Cross Connect Configuration

To display the status for the VXLAN Cross Connect configuration, enter one of the following commands:

**Table 5: Display VXLAN Cross Connect Information**

| Command                                            | Purpose                            |
|----------------------------------------------------|------------------------------------|
| <b>show running-config vlan</b> <i>session-num</i> | Displays VLAN information.         |
| <b>show nve vni</b>                                | Displays VXLAN VNI status.         |
| <b>show nve vni</b> <i>session-num</i>             | Displays VXLAN VNI status per VNI. |

Example of the **show run vlan 503** command:

```
switch(config)# sh run vlan 503

!Command: show running-config vlan 503
!Running configuration last done at: Mon Jul 9 13:46:03 2018
!Time: Tue Jul 10 14:12:04 2018

version 9.2(1) Bios:version 07.64
vlan 503
vlan 503
 vn-segment 5503
 xconnect
```

Example of the **show nve vni 5503** command:

```
switch(config)# sh nve vni 5503
Codes: CP - Control Plane DP - Data Plane
 UC - Unconfigured SA - Suppress ARP
 SU - Suppress Unknown Unicast
Interface VNI Multicast-group State Mode Type [BD/VRF] Flags

nve1 5503 225.5.0.3 Up CP L2 [503] SA Xconn
```

Example of the **show nve vni** command:

```
switch(config)# sh nve vni
Codes: CP - Control Plane DP - Data Plane
 UC - Unconfigured SA - Suppress ARP
 SU - Suppress Unknown Unicast
Interface VNI Multicast-group State Mode Type [BD/VRF] Flags

nve1 5501 225.5.0.1 Up CP L2 [501] SA
nve1 5502 225.5.0.2 Up CP L2 [502] SA
nve1 5503 225.5.0.3 Up CP L2 [503] SA Xconn
nve1 5504 UnicastBGP Up CP L2 [504] SA Xconn
nve1 5505 225.5.0.5 Up CP L2 [505] SA Xconn
nve1 5506 UnicastBGP Up CP L2 [506] SA Xconn
nve1 5507 225.5.0.7 Up CP L2 [507] SA Xconn
nve1 5510 225.5.0.10 Up CP L2 [510] SA Xconn
nve1 5511 225.5.0.11 Up CP L2 [511] SA Xconn
```

|      |      |            |    |    |    |       |    |       |
|------|------|------------|----|----|----|-------|----|-------|
| nve1 | 5512 | 225.5.0.12 | Up | CP | L2 | [512] | SA | Xconn |
| nve1 | 5513 | UnicastBGP | Up | CP | L2 | [513] | SA | Xconn |
| nve1 | 5514 | 225.5.0.14 | Up | CP | L2 | [514] | SA | Xconn |
| nve1 | 5515 | UnicastBGP | Up | CP | L2 | [515] | SA | Xconn |
| nve1 | 5516 | UnicastBGP | Up | CP | L2 | [516] | SA | Xconn |
| nve1 | 5517 | UnicastBGP | Up | CP | L2 | [517] | SA | Xconn |
| nve1 | 5518 | UnicastBGP | Up | CP | L2 | [518] | SA | Xconn |

## Configuring NGOAM for VXLAN Cross Connect

This procedure describes how to configure NGOAM for VXLAN Cross Connect.

### Procedure

|               | Command or Action                                                                                                                            | Purpose                                                                                                |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------|
| <b>Step 1</b> | <b>configure terminal</b>                                                                                                                    | Enters global configuration mode.                                                                      |
| <b>Step 2</b> | <b>feature ngoam</b><br><br><b>Example:</b><br><code>switch(config)# feature ngoam</code>                                                    | Enters the NGOAM feature.                                                                              |
| <b>Step 3</b> | <b>ngoam install acl</b><br><br><b>Example:</b><br><code>switch(config)# ngoam install acl</code>                                            | Installs NGOAM Access Control List (ACL).                                                              |
| <b>Step 4</b> | (Optional) <b>ngoam xconnect hb-interval interval</b><br><br><b>Example:</b><br><code>switch(config)# ngoam xconnect hb-interval 5000</code> | Configures the heart beat interval. Range of <i>interval</i> is 150 to 5000. The default value is 190. |

## Verifying NGOAM for VXLAN Cross Connect

To display the NGOAM status for the VXLAN Cross Connect configuration, enter one of the following commands:

**Table 6: Display VXLAN Cross Connect Information**

| Command                                        | Purpose                                                 |
|------------------------------------------------|---------------------------------------------------------|
| <b>show ngoam xconnect session all</b>         | Displays the summary of xconnect sessions.              |
| <b>show ngoam xconnect session session-num</b> | Displays detailed xconnect information for the session. |

Example of the **show ngoam xconnect session all** command:

```
switch(config)# sh ngoam xconnect session all
```

```

States: LD = Local interface down, RD = Remote interface Down
HB = Heartbeat lost, DB = Database/Routes not present
* - Showing Vpc-peer interface info

```

| Vlan | Peer-ip/vni    | XC-State | Local-if/State | Rmt-if/State |
|------|----------------|----------|----------------|--------------|
| 507  | 6.6.6.6 / 5507 | Active   | Eth1/7 / UP    | Eth1/5 / UP  |
| 508  | 7.7.7.7 / 5508 | Active   | Eth1/8 / UP    | Eth1/5 / UP  |
| 509  | 7.7.7.7 / 5509 | Active   | Eth1/9 / UP    | Eth1/9 / UP  |
| 510  | 6.6.6.6 / 5510 | Active   | Po303 / UP     | Po103 / UP   |
| 513  | 6.6.6.6 / 5513 | Active   | Eth1/6 / UP    | Eth1/8 / UP  |

Example of the **show ngoam xconnect session 507** command:

```

switch(config)# sh ngoam xconnect session 507
Vlan ID: 507
Peer IP: 6.6.6.6 VNI : 5507
State: Active
Last state update: 07/09/2018 13:47:03.849
Local interface: Eth1/7 State: UP
Local vpc interface Unknown State: DOWN
Remote interface: Eth1/5 State: UP
Remote vpc interface: Unknown State: DOWN
switch(config)#

```

## NGOAM Authentication

NGOAM provides the interface statistics in the pathtrace response. NGOAM authenticates the pathtrace requests to provide the statistics by using the HMAC MD5 authentication mechanism.

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

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

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

```

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

```

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

```
switch# pathtrace nve ip 12.0.22.1 profile 1 vni 31000 req-stats ver
Path trace Request to peer ip 12.0.22.1 source ip 11.0.22.1
Hop Code ReplyIP IngressI/f EgressI/f State
=====
 1 !Reply from 55.55.55.2, Eth5/7/1 Eth5/7/2 UP / UP
 Input Stats: PktRate:0 ByteRate:0 Load:0 Bytes:339573434 unicast:14657 mcast:307581
bcast:67 discards:0 errors:3 unknown:0 bandwidth:42949672970000000
 Output Stats: PktRate:0 ByteRate:0 load:0 bytes:237399176 unicast:2929 mcast:535710
bcast:10408 discards:0 errors:0 bandwidth:42949672970000000
 2 !Reply from 12.0.22.1, Eth1/7 Unknown UP / DOWN
 Input Stats: PktRate:0 ByteRate:0 Load:0 Bytes:4213416 unicast:275 mcast:4366 bcast:3
discards:0 errors:0 unknown:0 bandwidth:42949672970000000
switch# conf t
switch(config)# no ngoam authentication-key 123456789
switch(config)# end
```

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

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

## Guidelines and Limitations for Q-in-VNI

Q-in-VNI has the following guidelines and limitations:

- Q-in-VNI and selective Q-in-VNI are supported with VXLAN Flood and Learn with Ingress Replication and VXLAN EVPN with Ingress Replication.
- Q-in-VNI, selective Q-in-VNI, and QinQ-QinVNI are not supported with the multicast underlay on Cisco Nexus 9000-EX platform switches.
- The **system dot1q-tunnel transit** command is required when running this feature on vPC VTEPs.
- Port VLAN mapping and Q-in-VNI cannot coexist on the same port.
- Port VLAN mapping and Q-in-VNI cannot coexist on a switch if the **system dot1q-tunnel transit** command is enabled.
- For proper operation during L3 uplink failure scenarios on vPC VTEPs, configure a backup SVI and enter the **system nve infra-vlans backup-svi-vlan** command. On Cisco Nexus 9000-EX platform switches, the backup SVI VLAN needs to be the native VLAN on the peer-link.



- Q-in-VNI only supports VXLAN bridging. It does not support VXLAN routing.
- The dot1q tunnel mode does not support ALE ports on Cisco Nexus 9300 Series and Cisco Nexus 9500 platform switches.
- Q-in-VNI does not support FEX.
- When configuring access ports and trunk ports for Cisco Nexus 9000 Series switches with a Network Forwarding Engine (NFE) or a Leaf Spine Engine (LSE), you can have access ports, trunk ports, and dot1q ports on different interfaces on the same switch.
- You cannot have the same VLAN configured for both dot1q and trunk ports/access ports.
- Disable ARP suppression on the provider VNI for ARP traffic originated from a customer VLAN in order to flow.

```
switch(config)# interface nve 1
switch(config-if-nve)# member VNI 10000011
switch(config-if-nve-vni)# no suppress-arp
```

- Cisco Nexus 9300 platform switches support single tag. You can enable it by entering the **no overlay-encapsulation vxlan-with-tag** command for the NVE interface:

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

```
!Command: show running-config interface nve1
!Time: Wed Jul 20 23:26:25 2016
```

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

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

- Cisco Nexus 9500 platform switches do not support single tag. They support only double tag.
- Cisco Nexus 9300-EX platform switches do not support double tag. They support only single tag.
- Cisco Nexus 9300-EX platform switches do not support traffic between ports configured for Q-in-VNI and ports configured for trunk.
- Q-in-VNI cannot coexist with a VTEP that has Layer 3 subinterfaces configured.
- When VLAN1 is configured as the native VLAN with selective Q-in-VNI with the multiple provider tag, traffic on the native VLAN gets dropped. Do not configure VLAN1 as the native VLAN when the port is configured with selective Q-in-VNI. When VLAN1 is configured as a customer VLAN, the traffic on VLAN1 gets dropped.
- The base port mode must be a dot1q tunnel port with an access VLAN configured.
- VNI mapping is required for the access VLAN on the port.
- If you have Q-in-VNI on one Cisco Nexus 9300-EX Series switch VTEP and trunk on another Cisco Nexus 9300-EX Series switch VTEP, the bidirectional traffic will not be sent between the two ports.

- Cisco Nexus 9300-EX Series of switches performing VXLAN and Q-in-Q, a mix of provider interface and VXLAN uplinks is not considered. The VXLAN uplinks have to be separated from the Q-in-Q provider or customer interface.

For vPC use cases, the following considerations must be made when VXLAN and Q-in-Q are used on the same switch.

- The vPC peer-link has to be specifically configured as a provider interface to ensure orphan-to-orphan port communication. In these cases, the traffic is sent with two IEEE 802.1q tags (double dot1q tagging). The inner dot1q is the customer VLAN ID while the outer dot1q is the provider VLAN ID (access VLAN).
- The vPC peer-link is used as backup path for the VXLAN encapsulated traffic in the case of an uplink failure. In Q-in-Q, the vPC peer-link also acts as the provider interface (orphan-to-orphan port communication). In this combination, use the native VLAN as the backup VLAN for traffic to handle uplink failure scenarios. Also make sure the backup VLAN is configured as a system infra VLAN (system nve infra-vlans).
- Q-in-VNI does not support vPC Fabric Peering.
- BPDU filter is required for Selective Q-in-VNI, as we do not support tunneling STP BPDU.

## Configuring Q-in-VNI

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

### Procedure

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

### Example

The following is an example of configuring Q-in-VNI:

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

```
switch(config-if) # spanning-tree bpdudfilter enable
switch(config-if) #
```

## Configuring Selective Q-in-VNI

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

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

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

- Selective Q-in-VNI is supported on both vPC and non-vPC ports on Cisco Nexus 9300-EX and 9300-FX/FXP/FX2 platform switches. This feature is not supported on Cisco Nexus 9200 and 9300 platform switches.
- Selective Q-in-VNI does not support vPC Fabric Peering.
- Configuring selective Q-in-VNI on one VTEP and configuring plain Q-in-VNI on the VXLAN peer is supported. Configuring one port with selective Q-in-VNI and the other port with plain Q-in-VNI on the same switch is supported.
- Selective Q-in-VNI is an ingress VLAN tag-policing feature. Only ingress VLAN tag policing is performed with respect to the selective Q-in-VNI configured range.

For example, selective Q-in-VNI customer VLAN range of 100-200 is configured on VTEP1 and customer VLAN range of 200-300 is configured on VTEP2. When traffic with VLAN tag of 175 is sent from VTEP1 to VTEP2, the traffic is accepted on VTEP1, since the VLAN is in the configured range and it is forwarded to the VTEP2. On VTEP2, even though VLAN tag 175 is not part of the configured range, the packet egresses out of the selective Q-in-VNI port. If a packet is sent with VLAN tag 300 from VTEP1, it is dropped because 300 is not in VTEP1's selective Q-in-VNI configured range.

- Port VLAN mapping and selective Q-in-VNI cannot coexist on the same port.
- Port VLAN mapping and selective Q-in-VNI cannot coexist on a switch if the **system dot1q-tunnel transit** command is enabled.
- Configure the **system dot1q-tunnel transit** command on vPC switches with selective Q-in-VNI configurations. This command is required to retain the inner Q-tag as the packet goes over the vPC peer link when one of the vPC peers has an orphan port. With this CLI configuration, the **vlan dot1Q tag native** functionality does not work.
- The native VLAN configured on the selective Q-in-VNI port cannot be a part of the customer VLAN range. If the native VLAN is part of the customer VLAN range, the configuration is rejected.

The provider VLAN can overlap with the customer VLAN range. For example, **switchport vlan mapping 100-1000 dot1q-tunnel 200**.

- By default, the native VLAN on any port is VLAN 1. If VLAN 1 is configured as part of the customer VLAN range using the **switchport vlan mapping <range>dot1q-tunnel <sp-vlan>** CLI command, the traffic with customer VLAN 1 is not carried over as VLAN 1 is the native VLAN on the port. If customer wants VLAN 1 traffic to be carried over the VXLAN cloud, they should configure a dummy native VLAN on the port whose value is outside the customer VLAN range.
- To remove some VLANs or a range of VLANs from the configured switchport VLAN mapping range on the selective Q-in-VNI port, use the **no** form of the **switchport vlan mapping <range>dot1q-tunnel <sp-vlan>** command.

For example, VLAN 100-1000 is configured on the port. To remove VLAN 200-300 from the configured range, use the **no switchport vlan mapping <200-300> dot1q-tunnel <sp-vlan>** command.

```
interface Ethernet1/32
 switchport
 switchport mode trunk
 switchport trunk native vlan 4049
 switchport vlan mapping 100-1000 dot1q-tunnel 21
 switchport trunk allowed vlan 21,4049
 spanning-tree 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
 spanning-tree bpdupfilter enable
 no shutdown
```

See the following configuration examples.

- See the following example for the provider VLAN configuration:

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

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

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

- See the following example for the interface nve configuration:

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

- See the following example for configuring an SVI in the native VLAN to routed traffic.

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

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

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

- Disable ARP suppression on the provider VNI for ARP traffic originated from a customer VLAN in order to flow.

```
switch(config)# interface nve 1
switch(config-if-nve)# member VNI 10000011
switch(config-if-nve-vni)# no suppress-arp
```

## Configuring Q-in-VNI with LACP Tunneling

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

### Procedure

|               | Command or Action                                                                | Purpose                                                          |
|---------------|----------------------------------------------------------------------------------|------------------------------------------------------------------|
| <b>Step 1</b> | <b>configure terminal</b>                                                        | Enters global configuration mode.                                |
| <b>Step 2</b> | <b>interface</b> <i>type port</i>                                                | Enters interface configuration mode.                             |
| <b>Step 3</b> | <b>switchport mode dot1q-tunnel</b>                                              | Enables dot1q-tunnel mode.                                       |
| <b>Step 4</b> | <b>switchport access vlan</b> <i>vlan-id</i>                                     | Specifies the port assigned to a VLAN.                           |
| <b>Step 5</b> | <b>interface nve</b> <i>x</i>                                                    | Creates a VXLAN overlay interface that terminates VXLAN tunnels. |
| <b>Step 6</b> | <b>overlay-encapsulation vxlan-with-tag</b><br><b>tunnel-control-frames lacp</b> | Enables Q-in-VNI for LACP tunneling.                             |

|  | Command or Action | Purpose                                                                                                                                                                                                                  |
|--|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  |                   | <b>Note</b> Use this form of the command for NX-OS 7.0(3)I3(1) and later releases.<br><br>For NX-OS 7.0(3)I2(2) and earlier releases, use the <b>overlay-encapsulation vxlan-with-tag tunnel-control-frames</b> command. |

### 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 bpdupfilter 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):

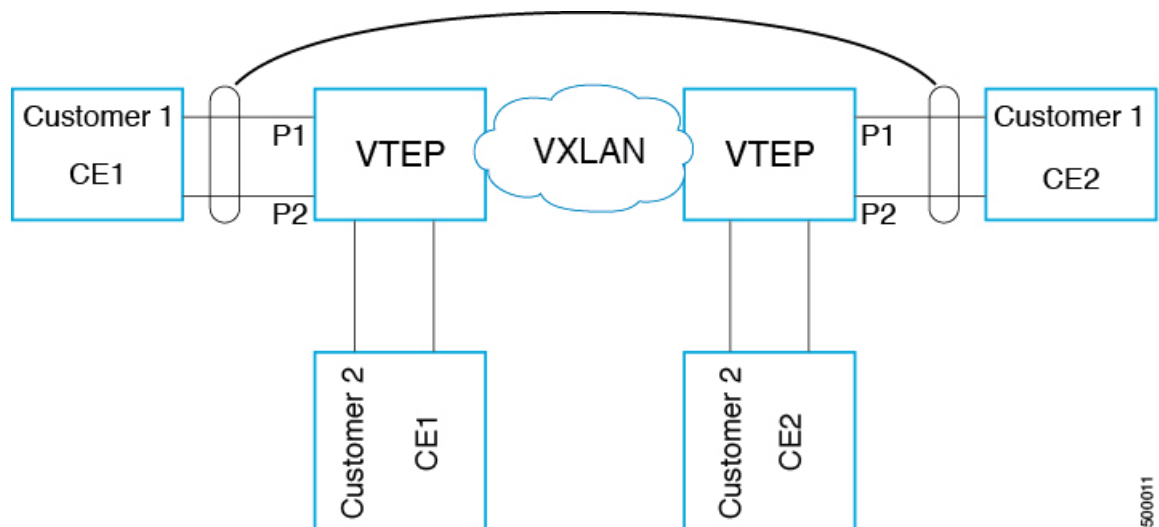
```
switch# config terminal
switch(config)# interface ethernet 1/4
switch(config-if)# switchport mode dot1q-tunnel
switch(config-if)# switchport access vlan 10
switch(config-if)# spanning-tree bpdupfilter enable
switch(config-if)# interface nve1
switch(config-if)# overlay-encapsulation vxlan-with-tag tunnel-control-frames lacp
```

**Note**

- STP is disabled on VNI mapped VLANs.
- No spanning-tree VLAN  $\diamond$  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 22: 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.

## Selective Q-in-VNI with Multiple Provider VLANs

### About Selective Q-in-VNI with Multiple Provider VLANs

Selective Q-in-VNI with multiple provider VLANs is a VXLAN tunneling feature. This feature allows a user specific range of customer VLANs on a port to be associated with one specific provider VLAN. It also enables you to have multiple customer-VLAN to provider-VLAN mappings on a port. Packets that come in with a VLAN tag which matches any of the configured customer VLANs on the port are tunneled across the VXLAN fabric using the properties of the service provider VNI. The VXLAN encapsulated packet carries the customer VLAN tag as part of the Layer 2 header of the inner packet.

### Guidelines and Limitations for Selective Q-in-VNI with Multiple Provider VLANs

Selective Q-in-VNI with multiple provider VLANs has the following guidelines and limitations:

- All the existing guidelines and limitations for [Selective Q-in-VNI](#) apply.
- This feature is supported with VXLAN BGP EVPN IR mode only.
- When enabling multiple provider VLANs on a vPC port channel, make sure that the configuration is consistent across the vPC peers.
- Port VLAN mapping and selective Q-in-VNI cannot coexist on the same port.
- Port VLAN mapping and selective Q-in-VNI cannot coexist on a switch if the **system dot1q-tunnel transit** command is enabled.
- The **system dot1q-tunnel transit** command is required when using this feature on vPC VTEPs.



- For proper operation during Layer 3 uplink failure scenarios on vPC VTEPs, configure the backup SVI and enter the **system nve infra-vlans backup-svi-vlan** command. On Cisco Nexus 9000-EX platform switches, the backup SVI VLAN must be the native VLAN on the peer-link.
- As a best practice, do not allow provider VLANs on a regular trunk.
- We recommend not creating or allowing customer VLANs on the switch where customer-VLAN to provider-VLAN mapping is configured.
- We do not support specific native VLAN configuration when the **switchport vlan mapping all dot1q-tunnel** command is entered.
- Selective Q-in-VNI with a multiple provider tag does not support vPC Fabric Peering.
- Disable ARP suppression on the provider VNI for ARP traffic originated from a customer VLAN in order to flow.

```
switch(config)# interface nve 1
switch(config-if-nve)# member VNI 10000011
switch(config-if-nve-vni)# no suppress-arp
```

- All incoming traffic should be tagged when the interface is configured with the **switchport vlan mapping all dot1q-tunnel** command.

## Configuring Selective Q-in-VNI with Multiple Provider VLANs

You can configure selective Q-in-VNI with multiple provider VLANs.

### Before you begin

You must configure provider VLANs and associate the VLAN to a vn-segment.

### Procedure

- Step 1** Enter global configuration mode.

```
switch# configure terminal
```

- Step 2** Configure Layer 2 VLANs and associate them to a vn-segment.

```
switch(config)# vlan 10
vn-segment 10000010
switch(config)# vlan 20
vn-segment 10000020
```

- Step 3** Enter interface configuration mode where the traffic comes in with a dot1Q VLAN tag.

```
switch(config)# interf port-channel 10
switch(config-if)# switchport
switch(config-if)# switchport mode trunk
switch(config-if)# switchport trunk native vlan 3962
switch(config-if)# switchport vlan mapping 2-400 dot1q-tunnel 10
switch(config-if)# switchport vlan mapping 401-800 dot1q-tunnel 20
switch(config-if)# switchport vlan mapping 801-1200 dot1q-tunnel 30
switch(config-if)# switchport vlan mapping 1201-1600 dot1q-tunnel 40
switch(config-if)# switchport vlan mapping 1601-2000 dot1q-tunnel 50
switch(config-if)# switchport vlan mapping 2001-2400 dot1q-tunnel 60
switch(config-if)# switchport vlan mapping 2401-2800 dot1q-tunnel 70
```

```

switch(config-if)# switchport vlan mapping 2801-3200 dot1q-tunnel 80
switch(config-if)# switchport vlan mapping 3201-3600 dot1q-tunnel 90
switch(config-if)# switchport vlan mapping 3601-3960 dot1q-tunnel 100
switch(config-if)# switchport trunk allowed vlan 10,20,30,40,50,60,70,80,90,100,3961-3967

```

## Example

This example shows how to configure Selective QinVni with multiple provider VLANs:

```

switch# show run vlan 121
vlan 121
vlan 121
 vn-segment 10000021

switch#
switch# sh run interf port-channel 5

interface port-channel5
 description VPC PO
 switchport
 switchport mode trunk
 switchport trunk native vlan 504
 switchport vlan mapping 11 dot1q-tunnel 111
 switchport vlan mapping 12 dot1q-tunnel 112
 switchport vlan mapping 13 dot1q-tunnel 113
 switchport vlan mapping 14 dot1q-tunnel 114
 switchport vlan mapping 15 dot1q-tunnel 115
 switchport vlan mapping 16 dot1q-tunnel 116
 switchport vlan mapping 17 dot1q-tunnel 117
 switchport vlan mapping 18 dot1q-tunnel 118
 switchport vlan mapping 19 dot1q-tunnel 119
 switchport vlan mapping 20 dot1q-tunnel 120
 switchport trunk allowed vlan 111-120,500-505
 vpc 5

switch#

switch# sh spanning-tree vlan 111

VLAN0111
 Spanning tree enabled protocol rstp
 Root ID Priority 32879
 Address 7079.b3cf.956d
 This bridge is the root
 Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

 Bridge ID Priority 32879 (priority 32768 sys-id-ext 111)
 Address 7079.b3cf.956d
 Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Interface Role Sts Cost Prio.Nbr Type

Po1 Desg FWD 1 128.4096 (vPC peer-link) Network P2p
Po5 Desg FWD 1 128.4100 (vPC) P2p
Eth1/7/2 Desg FWD 10 128.26 P2p

switch#

switch# sh vlan internal info mapping | b Po5
ifindex Po5(0x16000004)

```

```

vlan mapping enabled: TRUE
vlan translation mapping information (count=10):
 Original Vlan Translated Vlan

 11 111
 12 112
 13 113
 14 114
 15 115
 16 116
 17 117
 18 118
 19 119
 20 120
switch#

switch# sh consistency-checker vxlan selective-qinvni interface port-channel 5
Performing port specific checks for intf port-channel5
Port specific selective QinVNI checks for interface port-channel5 : PASS
Performing port specific checks for intf port-channel5
Port specific selective QinVNI checks for interface port-channel5 : PASS

switch#

```

## Configuring QinQ-QinVNI

### Overview for QinQ-QinVNI

- QinQ-QinVNI is a VXLAN tunneling feature that allows you to configure a trunk port as a multi-tag port to preserve the customer VLANs that are carried across the network.
- On a port that is configured as multi-tag, packets are expected with multiple-tags or at least one tag. When multi-tag packets ingress on this port, the outer-most or first tag is treated as provider-tag or provider-vlan. The remaining tags are treated as customer-tag or customer-vlan.
- This feature is supported on both vPC and non-vPC ports.
- Ensure that the **switchport trunk allow-multi-tag** command is configured on both of the vPC-peers. It is a type 1 consistency check.
- This feature is supported with VXLAN Flood and Learn and VXLAN EVPN.

### Guidelines and Limitations for QinQ-QinVNI

QinQ-QinVNI has the following guidelines and limitations:

- This feature is supported on the Cisco Nexus 9300-FX/FX2 switches.
- This feature supports vPC Fabric Peering, beginning with Cisco NX-OS Release 9.2(3). For more information, see the [Configuring vPC Fabric Peering, on page 257](#) chapter.
- 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                                           |
|---------------|---------------------------------------------------------------------------------------------------------------------|---------------------------------------------------|
| <b>Step 1</b> | <b>configure terminal</b><br><b>Example:</b><br><code>switch# configure terminal</code>                             | Enters global configuration mode.                 |
| <b>Step 2</b> | <b>interface ethernet <i>slot/port</i></b><br><b>Example:</b><br><code>switch(config)# interface ethernet1/7</code> | Specifies the interface that you are configuring. |
| <b>Step 3</b> | <b>switchport</b><br><b>Example:</b><br><code>switch(config-if)# switchport</code>                                  | Configures it as a Layer 2 port.                  |
| <b>Step 4</b> | <b>switchport mode trunk</b><br><b>Example:</b><br><code>switch(config-if)# switchport mode trunk</code>            | Sets the interface as a Layer 2 trunk port.       |

|               | Command or Action                                                                                                                                 | Purpose                                                                                                                                                                                                               |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 5</b> | <b>switchport trunk native vlan <i>vlan-id</i></b><br><b>Example:</b><br><pre>switch(config-inf)# switchport trunk native vlan 30</pre>           | Sets the native VLAN for the 802.1Q trunk. Valid values are from 1 to 4094. The default value is VLAN1.                                                                                                               |
| <b>Step 6</b> | <b>switchport trunk allowed vlan <i>vlan-list</i></b><br><b>Example:</b><br><pre>switch(config-inf)# switchport trunk allowed vlan 10,20,30</pre> | 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.  |
| <b>Step 7</b> | <b>switchport trunk allow-multi-tag</b><br><b>Example:</b><br><pre>switch(config-inf)# switchport trunk allow-multi-tag</pre>                     | 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

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





## CHAPTER 12

# Configuring Port VLAN Mapping

---

This chapter contains the following sections:

- [About Translating Incoming VLANs, on page 211](#)
- [Guidelines and Limitations for Port VLAN Mapping, on page 212](#)
- [Configuring Port VLAN Mapping on a Trunk Port, on page 214](#)
- [Configuring Inner VLAN and Outer VLAN Mapping on a Trunk Port, on page 216](#)

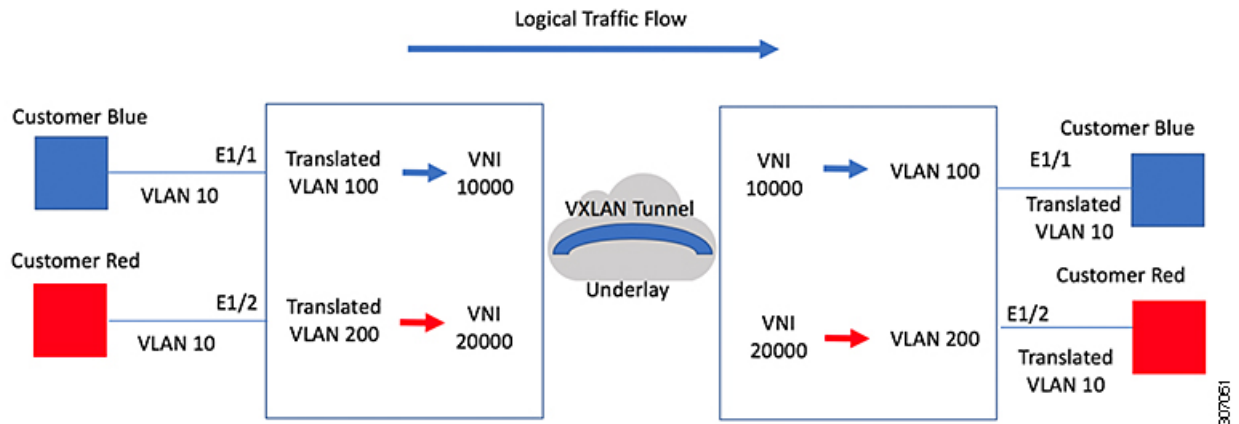
## About Translating Incoming VLANs

Sometimes a VLAN translation is required or desired. One such use case is when a service provider has multiple customers connecting to the same physical switch using the same VLAN encapsulation, but they are not and should not be on the same Layer 2 segment. In such cases translating the incoming VLAN to a unique VLAN that is then mapped to a VNI is the right way to extending the segment. In the figure below two customers, Blue and Red are both connecting to the leaf using VLAN 10 as their encapsulation.

Customers Blue and Red should not be on the same VNI. In this example VLAN 10 for Customer Blue (on interface E1/1) is mapped/translated to VLAN 100, and VLAN 10 for customer Red (on interface E1/2) is mapped to VLAN 200. In turn, VLAN 100 is mapped to VNI 10000 and VLAN 200 is mapped to VNI 20000.

On the other leaf, this mapping is applied in reverse. Incoming VXLAN encapsulated traffic on VNI 10000 is mapped to VLAN 100 which in turn is mapped to VLAN 10 on Interface E1/1. VXLAN encapsulated traffic on VNI 20000 is mapped to VLAN 200 which in turn is mapped to VLAN 10 on Interface E1/2.

Figure 23: Logical Traffic Flow



You can configure VLAN translation between the ingress (incoming) VLAN and a local (translated) VLAN on a port. For the traffic arriving on the interface where VLAN translation is enabled, the incoming VLAN is mapped to a translated VLAN that is VXLAN enabled.

On the underlay, this is mapped to a VNI, the inner dot1q is deleted, and switched over to the VXLAN network. On the egress switch, the VNI is mapped to a translated VLAN. On the outgoing interface, where VLAN translation is configured, the traffic is converted to the original VLAN and egressed out. Refer to the VLAN counters on the translated VLAN for the traffic counters and not on the ingress VLAN. Port VLAN (PV) mapping is an access side feature and is supported with both multicast and ingress replication for flood and learn and MP-BGP EVPN mode for VXLAN.

## Guidelines and Limitations for Port VLAN Mapping

The following are the guidelines and Limitations for Port VLAN Mapping:

- Beginning with Cisco NX-OS Release 9.2(3), support is added for vPC Fabric Peering. For more information, see the [Configuring vPC Fabric Peering, on page 257](#) chapter.
- VLAN translation is supported only on VXLAN enabled VLANs
- The ingress (incoming) VLAN does not need to be configured on the switch as a VLAN. The translated VLAN needs to be configured and a vn-segment mapping given to it. An NVE interface with VNI mapping is essential for the same.
- All Layer 2 source address learning and Layer 2 MAC destination lookup occurs on the translated VLAN. Refer to the VLAN counters on the translated VLAN and not on the ingress (incoming) VLAN.
- Port VLAN mapping is supported on Cisco Nexus 9300 platform switches. Port VLAN mapping is supported on Cisco Nexus 9300-EX platform switches.
- Cisco Nexus 9300, and 9500 switches support switching and routing on overlapped VLAN interfaces; only VLAN-mapping switching is applicable for Cisco Nexus 9500 with EX/FX line cards and 9300-EX/FX/FX2 platform switches.
- On Cisco Nexus 9300 Series switches with NFE ASIC, PV routing is not supported on 40 G ALE ports.



- PV routing supports configuring an SVI on the translated VLAN for flood and learn and BGP EVPN mode for VXLAN.
- VLAN translation (mapping) is supported on Cisco Nexus 9000 Series switches with a Network Forwarding Engine (NFE).
- When changing a property on a translated VLAN, the port that has a mapping configuration with that VLAN as the translated VLAN, must be flapped to ensure correct behavior.

```
Int eth 1/1
switchport vlan mapping 101 10
.
.
.

/****Deleting vn-segment from vlan 10.****/
/****Adding vn-segment back.****/
/****Flap Eth 1/1 to ensure correct behavior.****/
```

- The following example shows incoming VLAN 10 being mapped to local VLAN 100. Local VLAN 100 will be the one mapped to a VXLAN VNI.

```
interface ethernet1/1
switchport vlan mapping 10 100
```

- The following is an example of overlapping VLAN for PV translation. In the first statement, VLAN-102 is a translated VLAN with VNI mapping. In the second statement, VLAN-102 the VLAN where it is translated to VLAN-103 with VNI mapping.

```
interface ethernet1/1
switchport vlan mapping 101 102
switchport vlan mapping 102 103/
```

- When adding a member to an existing port channel using the force command, the "mapping enable" configuration must be consistent. For example:

```
Int po 101
switchport vlan mapping enable
switchport vlan mapping 101 10
switchport trunk allowed vlan 10

int eth 1/8
/****No configuration****/
```




---

**Note** The **switchport vlan mapping enable** command is supported only when the port mode is trunk.

---

- Port VLAN mapping is not supported on Cisco Nexus 9200 platform switches.
- VLAN mapping helps with VLAN localization to a port, scoping the VLANs per port. A typical use case is in the service provider environment where the service provider leaf switch has different customers with overlapping VLANs that come in on different ports. For example, customer A has VLAN 10 coming in on Eth 1/1 and customer B has VLAN 10 coming in on Eth 2/2.

In this scenario, you can map the customer VLAN to a provider VLAN and map that to a Layer 2 VNI. There is an operational benefit in terminating different customer VLANs and mapping them to the fabric-managed VLANs, L2 VNIs.

- An NVE interface with VNI mapping must be configured for Port VLAN translation to work.

# Configuring Port VLAN Mapping on a Trunk Port

## Before you begin

- Ensure that the physical or port channel on which you want to implement VLAN translation is configured as a Layer 2 trunk port.
- Ensure that the translated VLANs are created on the switch and are also added to the Layer 2 trunk ports trunk-allowed VLAN vlan-list.



**Note** As a best practice, do not add the ingress VLAN ID to the switchport allowed vlan-list under the interface.

- Ensure that all translated VLANs are VXLAN enabled.

## Procedure

|               | Command or Action                                                                                                                                     | Purpose                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 1</b> | <b>configure terminal</b><br><br><b>Example:</b><br><pre>switch# configure terminal</pre>                                                             | Enters global configuration mode.                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>Step 2</b> | <b>interface type/port</b><br><br><b>Example:</b><br><pre>switch(config)# interface Ethernet1/1</pre>                                                 | Specifies the interface that you are configuring.                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>Step 3</b> | <b>[no] switchport vlan mapping enable</b><br><br><b>Example:</b><br><pre>switch(config-if)# [no] switchport vlan mapping enable</pre>                | Enables VLAN translation on the switch port. VLAN translation is disabled by default.<br><br><b>Note</b> Use the <b>no</b> form of this command to disable VLAN translation.                                                                                                                                                                                                                                                                                             |
| <b>Step 4</b> | <b>[no] switchport vlan mapping vlan-id translated-vlan-id</b><br><br><b>Example:</b><br><pre>switch(config-if)# switchport vlan mapping 10 100</pre> | Translates a VLAN to another VLAN. <ul style="list-style-type: none"> <li>• The range for both the <i>vlan-id</i> and <i>translated-vlan-id</i> arguments are from 1 to 4094.</li> <li>• 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.</li> </ul> |

|               | Command or Action                                                                                                                           | Purpose                                                                                                                                                                                                                                                                                                                                                                                                                            |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|               |                                                                                                                                             | <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 local translated VLAN. On the outgoing interface, where VLAN translation is configured, the traffic is converted to the original VLAN and egresses out.</p> <p><b>Note</b> Use the <b>no</b> form of this command to clear the mappings between a pair of VLANs.</p> |
| <b>Step 5</b> | <p><b>[no] switchport vlan mapping all</b></p> <p><b>Example:</b></p> <pre>switch(config-if)# switchport vlan mapping all</pre>             | Removes all VLAN mappings configured on the interface.                                                                                                                                                                                                                                                                                                                                                                             |
| <b>Step 6</b> | <p><b>copy running-config startup-config</b></p> <p><b>Example:</b></p> <pre>switch(config-if)# copy running-config startup-config</pre>    | <p>Copies the running configuration to the startup configuration.</p> <p><b>Note</b> The VLAN translation configuration does not become effective until the switch port becomes an operational trunk port.</p>                                                                                                                                                                                                                     |
| <b>Step 7</b> | <p><b>show interface [if-identifier] vlan mapping</b></p> <p><b>Example:</b></p> <pre>switch# show interface ethernet1/1 vlan mapping</pre> | Displays VLAN mapping information for a range of interfaces or for a specific interface.                                                                                                                                                                                                                                                                                                                                           |

### Example

This example shows how to configure VLAN translation between (the ingress) VLAN 10 and (the local) VLAN 100. The show vlan counters command output shows the statistic counters as translated VLAN instead of customer VLAN.

```
switch# configure terminal
switch(config)# interface ethernet1/1
switch(config-if)# switchport vlan mapping enable
switch(config-if)# switchport vlan mapping 10 100
switch(config-if)# switchport trunk allowed vlan 100
switch(config-if)# show interface ethernet1/1 vlan mapping
Interface eth1/1:
Original VLAN Translated VLAN

10 100

switch(config-if)# show vlan counters
Vlan Id :100
Unicast Octets In :292442462
Unicast Packets In :1950525
Multicast Octets In :14619624
```

```

Multicast Packets In :91088
Broadcast Octets In :14619624
Broadcast Packets In :91088
Unicast Octets Out :304012656
Unicast Packets Out :2061976
L3 Unicast Octets In :0
L3 Unicast Packets In :0

```

## Configuring Inner VLAN and Outer VLAN Mapping on a Trunk Port

Configuring Inner VLAN and Outer VLAN Mapping on a Trunk Port is applicable only for Cisco Nexus 9300 platforms and not supported on Cisco Nexus 9200, 9300-EX, 9300-FX, 9300-FX2, 9300-FX3, 9300-GX, 9300-GX2, 9364C, 9332C platforms.

You can configure VLAN translation from an inner VLAN and an outer VLAN to a local (translated) VLAN on a port. For the double tag VLAN traffic arriving on the interfaces where VLAN translation is enabled, the inner VLAN and outer VLAN are mapped to a translated VLAN that is VXLAN enabled.

Notes for configuring inner VLAN and outer VLAN mapping:

- Inner and outer VLAN cannot be on the trunk allowed list on a port where inner VLAN and outer VLAN is configured.

For example:

```

switchport vlan mapping 11 inner 12 111
switchport trunk allowed vlan 11-12,111 /***Not valid because 11 is outer VLAN and 12
is inner VLAN.*** /

```

- On the same port, no two mapping (translation) configurations can have the same outer (or original) or translated VLAN. Multiple inner VLAN and outer VLAN mapping configurations can have the same inner VLAN.

For example:

```

switchport vlan mapping 101 inner 102 1001
switchport vlan mapping 101 inner 103 1002 /***Not valid because 101 is already used
as an original VLAN.*** /
switchport vlan mapping 111 inner 104 1001 /***Not valid because 1001 is already used
as a translated VLAN.*** /
switchport vlan mapping 106 inner 102 1003 /***Valid because inner vlan can be the
same.*** /

```

- When a packet comes double-tagged on a port which is enabled with the inner option, only bridging is supported.
- VXLAN PV routing is not supported for double-tagged frames.

### Procedure

|               | Command or Action         | Purpose                           |
|---------------|---------------------------|-----------------------------------|
| <b>Step 1</b> | <b>configure terminal</b> | Enters global configuration mode. |

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

### Example

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

```
switch# configure terminal
switch(config)# interface ethernet1/1
switch(config-if)# switchport mode trunk
switch(config-if)# switchport vlan mapping enable
switch(config-if)# switchport vlan mapping 11 inner 12 111
switch(config-if)# switchport trunk allowed vlan 101-170
switch(config-if)# no shutdown
```

```
switch(config-if)# show mac address-table dynamic vlan 111
```

Legend:

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

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





## CHAPTER 13

# Configuring IGMP Snooping

This chapter contains the following sections:

- [Configuring IGMP Snooping Over VXLAN, on page 219](#)

## Configuring IGMP Snooping Over VXLAN

### Overview of IGMP Snooping Over VXLAN

By default, multicast traffic over VXLAN is flooded in the VNI/VLAN like any broadcast and unknown unicast traffic. With IGMP snooping enabled, each VTEP can snoop IGMP reports and only forward multicast traffic towards interested receivers.

The configuration of IGMP snooping is the same in VXLAN as in the configuration of IGMP snooping in a regular VLAN domain. For more information on IGMP snooping, see the *Configuring IGMP Snooping* section in the [Cisco Nexus 9000 Series NX-OS Multicast Routing Configuration Guide, Release 7.x](#).

### Guidelines and Limitations for IGMP Snooping Over VXLAN

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

- IGMP snooping over VXLAN is not supported on VLANs with FEX member ports.
- IGMP snooping over VXLAN is supported with both IR and multicast underlay.
- IGMP snooping over VXLAN is supported in BGP EVPN topologies, not flood and learn topologies.

## Configuring IGMP Snooping Over VXLAN

### Procedure

|        | Command or Action                 | Purpose                           |
|--------|-----------------------------------|-----------------------------------|
| Step 1 | switch# <b>configure terminal</b> | Enters global configuration mode. |

|               | Command or Action                                                      | Purpose                                                                                                                                                                                 |
|---------------|------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 2</b> | switch(config)# <b>ip igmp snooping vxlan</b>                          | Enables IGMP snooping for VXLAN VLANs. You have to explicitly configure this command to enable snooping for VXLAN VLANs.                                                                |
| <b>Step 3</b> | switch(config)# <b>ip igmp snooping disable-nve-static-router-port</b> | Configures IGMP snooping over VXLAN to not include NVE as static mrouter port using this global CLI command. IGMP snooping over VXLAN has the NVE interface as mrouter port by default. |





## CHAPTER 14

# Configuring VLANs

---

This chapter contains the following sections:

- [About Private VLANs over VXLAN, on page 221](#)
- [Guidelines and Limitations for Private VLANs over VXLAN, on page 222](#)
- [Configuration Example for Private VLANs, on page 222](#)

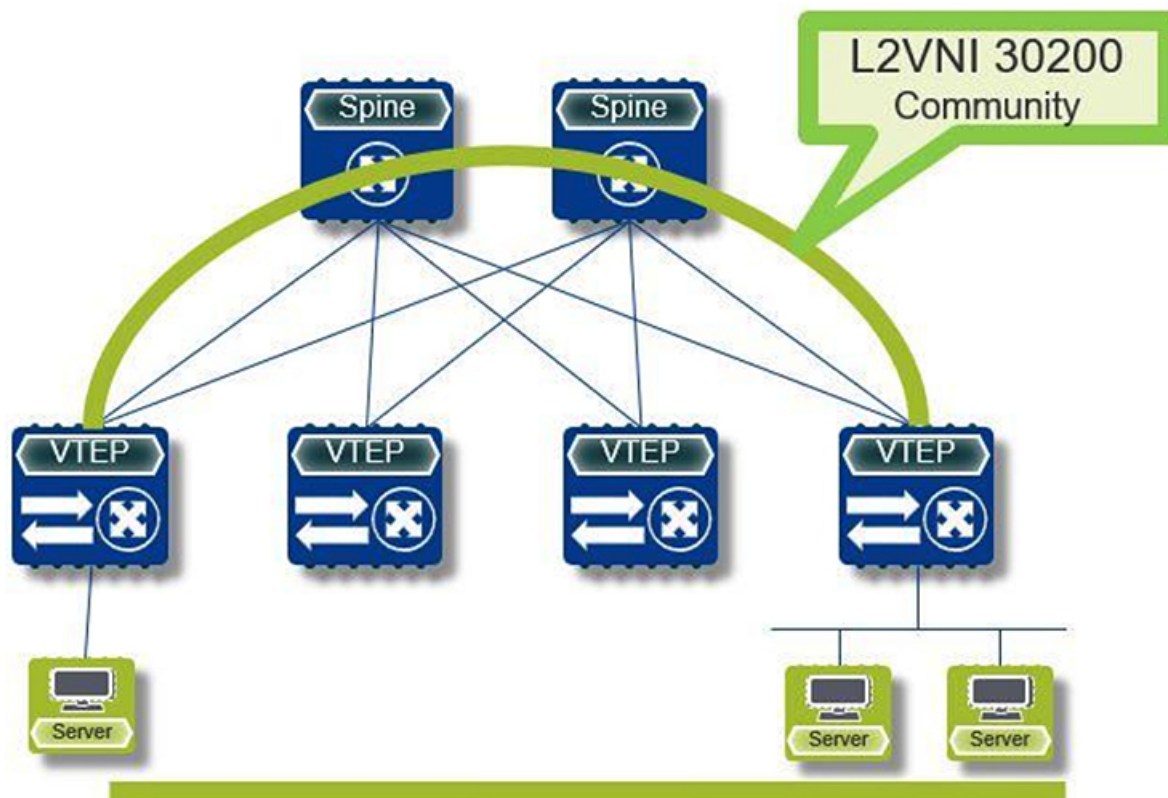
## About Private VLANs over VXLAN

Beginning with Cisco NX-OS Release 9.2(3), support added for vPC Fabric Peering. For more information, see the [Configuring vPC Fabric Peering, on page 257](#) chapter.

The private VLAN feature allows segmenting the Layer 2 broadcast domain of a VLAN into subdomains. A subdomain is represented by a pair of private VLANs: a primary VLAN and a secondary VLAN. A private VLAN domain can have multiple private VLAN pairs, one pair for each subdomain. All VLAN pairs in a private VLAN domain share the same primary VLAN. The secondary VLAN ID differentiates one subdomain from another.

Private VLANs over VXLAN extends private VLAN across VXLAN. The secondary VLAN can exist on multiple VTEPs across VXLAN. MAC address learning happens over the primary VLAN and advertises via BGP EVPN. When traffic is encapsulated, the VNI used is that of the secondary VLAN. The feature also supports Anycast Gateway. Anycast Gateway must be defined using the primary VLAN.

Figure 24: L2VNI 30200 Community



307054

## Guidelines and Limitations for Private VLANs over VXLAN

Private VLANs over VXLAN has the following configuration guidelines and limitations:

- The following platforms support private VLANs over VXLAN:
  - Cisco Nexus 9300-EX platform switches
  - Cisco Nexus 9300-FX/FX2 platform switches
- Flood and learn underlay is not supported.
- Fabric Extenders (FEX) VLAN cannot be mapped to a private VLAN.
- vPC Fabric Peering supports private VLANs.

## Configuration Example for Private VLANs

The following is a private VLAN configuration example:

```
vlan 500
 private-vlan primary
 private-vlan association 501-503
```

```

 vn-segment 5000
vlan 501
 private-vlan isolated
 vn-segment 5001
vlan 502
 private-vlan community
 vn-segment 5002
vlan 503
 private-vlan community
 vn-segment 5003

vlan 1001
 !L3 VNI for tenant VRF
 vn-segment 900001

interface Vlan500
 no shutdown
 private-vlan mapping 501-503
 vrf member vxlan-900001
 no ip redirects
 ip address 50.1.1.1/8
 ipv6 address 50::1:1:1/64
 no ipv6 redirects
 fabric forwarding mode anycast-gateway

interface Vlan1001
 no shutdown
 vrf member vxlan-900001
 no ip redirects
 ip forward
 ipv6 forward
 ipv6 address use-link-local-only
 no ipv6 redirects

interface nve 1
 no shutdown
 host-reachability protocol bgp
 source-interface loopback0
 member vni 5000
 mcast-group 225.5.0.1
 member vni 5001
 mcast-group 225.5.0.2
 member vni 5002
 ingress-replication protocol bgp
 member vni 5003
 mcast-group 225.5.0.4
 member vni 900001 associate-vrf

```




---

**Note** If you use an external gateway, the interface towards the external router must be configured as a PVLAN promiscuous port

---

```

interface ethernet 2/1
 switchport
 switchport mode private-vlan trunk promiscuous
 switchport private-vlan mapping trunk 500 199,200,201
exit

```





## CHAPTER 15

# Configuring Policy-Based Redirect

---

This chapter contains the following sections:

- [Service Redirection in VXLAN EVPN Fabrics, on page 225](#)
- [Guidelines and Limitations for Policy-Based Redirect, on page 225](#)
- [Enabling the Policy-Based Redirect Feature, on page 226](#)
- [Configuring a Route Policy, on page 226](#)
- [Verifying the Policy-Based Redirect Configuration, on page 228](#)
- [Configuration Example for Policy-Based Redirect, on page 228](#)

## Service Redirection in VXLAN EVPN Fabrics

Today, insertion of service appliances (also referred to as service nodes or service endpoints) such as firewalls, load-balancers, etc are needed to secure and optimize applications within a data center. This section describes the Layer 4-Layer 7 service insertion and redirection features offered on VXLAN EVPN fabrics that provides sophisticated mechanisms to onboard and selectively redirect traffic to these services.

## Guidelines and Limitations for Policy-Based Redirect

The following guidelines and limitations apply to PBR over VXLAN.

- The following platforms support PBR over VXLAN:
  - Cisco Nexus 9332C and 9364C switches
  - Cisco Nexus 9300-EX switches
  - Cisco Nexus 9300-FX/FX2 switches
  - Cisco Nexus 9504 and 9508 switches with -EX/FX line cards
- PBR over VXLAN doesn't support the following features: VTEP ECMP, and the **load-share** keyword in the **set {ip | ipv6} next-hop ip-address** command.

# Enabling the Policy-Based Redirect Feature

To configure basic PBR, in cases where the advanced (and recommended) ePBR functions are not deployed, see the following sections:

- [Enabling the Policy-Based Redirect Feature, on page 226](#)
- [Configuring a Route Policy, on page 226](#)
- [Verifying the Policy-Based Redirect Configuration, on page 228](#)
- [Configuration Example for Policy-Based Redirect, on page 228](#)

## Before you begin

Enable the policy-based redirect feature before you can configure a route policy.

## Procedure

|               | Command or Action                                                                                                                              | Purpose                                   |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------|
| <b>Step 1</b> | <b>configure terminal</b><br><br><b>Example:</b><br><code>switch# configure terminal</code>                                                    | Enters global configuration mode.         |
| <b>Step 2</b> | <b>[no] feature pbr</b><br><br><b>Example:</b><br><code>switch(config)# feature pbr</code>                                                     | Enables the policy-based routing feature. |
| <b>Step 3</b> | (Optional) <b>show feature</b><br><br><b>Example:</b><br><code>switch(config)# show feature</code>                                             | Displays enabled and disabled features.   |
| <b>Step 4</b> | (Optional) <b>copy running-config startup-config</b><br><br><b>Example:</b><br><code>switch(config)# copy running-config startup-config</code> | Saves this configuration change.          |

# Configuring a Route Policy

You can use route maps in policy-based routing to assign routing policies to the inbound interface. Cisco NX-OS routes the packets when it finds a next hop and an interface.



**Note** The switch has a RACL TCAM region by default for IPv4 traffic.

### Before you begin

Configure the RACL TCAM region (using TCAM carving) before you apply the policy-based routing policy. For instructions, see the “Configuring ACL TCAM Region Sizes” section in the [Cisco Nexus 9000 Series NX-OS Security Configuration Guide, Release 9.2\(x\)](#).

### Procedure

|               | Command or Action                                                                                                                                                  | Purpose                                                                                                                                                                                          |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 1</b> | <b>configure terminal</b><br><br><b>Example:</b><br>switch# <b>configure terminal</b>                                                                              | Enters global configuration mode.                                                                                                                                                                |
| <b>Step 2</b> | <b>interface type slot/port</b><br><br><b>Example:</b><br>switch(config)# <b>interface ethernet 1/2</b>                                                            | Enters interface configuration mode.                                                                                                                                                             |
| <b>Step 3</b> | <b>{ip   ipv6} policy route-map map-name</b><br><br><b>Example:</b><br>switch(config-if)# <b>ip policy route-map Testmap</b>                                       | Assigns a route map for IPv4 or IPv6 policy-based routing to the interface.                                                                                                                      |
| <b>Step 4</b> | <b>route-map map-name [permit   deny] [seq]</b><br><br><b>Example:</b><br>switch(config-if)# <b>route-map Testmap</b>                                              | Creates a route map or enters route-map configuration mode for an existing route map. Use <i>seq</i> to order the entries in a route map.                                                        |
| <b>Step 5</b> | <b>match {ip   ipv6} address access-list-name name [name...]</b><br><br><b>Example:</b><br>switch(config-route-map)# <b>match ip address access-list-name ACL1</b> | Matches an IPv4 or IPv6 address against one or more IPv4 or IPv6 access control lists (ACLs). This command is used for policy-based routing and is ignored by route filtering or redistribution. |
| <b>Step 6</b> | <b>set ip next-hop address1</b><br><br><b>Example:</b><br>switch(config-route-map)# <b>set ip next-hop 192.0.2.1</b>                                               | Sets the IPv4 next-hop address for policy-based routing.                                                                                                                                         |
| <b>Step 7</b> | <b>set ipv6 next-hop address1</b><br><br><b>Example:</b><br>switch(config-route-map)# <b>set ipv6 next-hop 2001:0DB8::1</b>                                        | Sets the IPv6 next-hop address for policy-based routing.                                                                                                                                         |
| <b>Step 8</b> | (Optional) <b>set interface null0</b><br><br><b>Example:</b><br>switch(config-route-map)# <b>set interface null0</b>                                               | Sets the interface that is used for routing. Use the <b>null0</b> interface to drop packets.                                                                                                     |

|               | Command or Action                                                                                                                                      | Purpose                          |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------|
| <b>Step 9</b> | (Optional) <b>copy running-config startup-config</b><br><br><b>Example:</b><br><pre>switch(config-route-map)# copy running-config startup-config</pre> | Saves this configuration change. |

## Verifying the Policy-Based Redirect Configuration

To display the policy-based redirect configuration information, perform one of the following tasks:

| Command                                     | Purpose                                            |
|---------------------------------------------|----------------------------------------------------|
| <b>show [ip   ipv6] policy [name]</b>       | Displays information about an IPv4 or IPv6 policy. |
| <b>show route-map [name] pbr-statistics</b> | Displays policy statistics.                        |

Use the **route-map map-name pbr-statistics** command to enable policy statistics. Use the **clear route-map map-name pbr-statistics** command to clear these policy statistics.

## Configuration Example for Policy-Based Redirect

Perform the following configuration on all tenant VTEPs, excluding the service VTEP.

```
feature pbr

ipv6 access-list IPV6_App_group_1
10 permit ipv6 any 2001:10:1:1::0/64

ip access-list IPV4_App_group_1
10 permit ip any 10.1.1.0/24

ipv6 access-list IPV6_App_group_2
10 permit ipv6 any 2001:20:1:1::0/64

ip access-list IPV4_App_group_2
10 permit ip any 20.1.1.0/24

route-map IPV6_PBR_Appgroup1 permit 10
 match ipv6 address IPV6_App_group_2
 set ipv6 next-hop 2001:100:1:1::20 (next hop is that of the firewall)

route-map IPV4_PBR_Appgroup1 permit 10
 match ip address IPV4_App_group_2
 set ip next-hop 10.100.1.20 (next hop is that of the firewall)

route-map IPV6_PBR_Appgroup2 permit 10
 match ipv6 address IPV6_App_group1
 set ipv6 next-hop 2001:100:1:1::20 (next hop is that of the firewall)

route-map IPV4_PBR_Appgroup2 permit 10
 match ip address IPV4_App_group_1
 set ip next-hop 10.100.1.20 (next hop is that of the firewall)
```



```

interface Vlan10
! tenant SVI appgroup 1
vrf member appgroup
 ip address 10.1.1.1/24
 no ip redirect
 ipv6 address 2001:10:1:1::1/64
 no ipv6 redirects
 fabric forwarding mode anycast-gateway
ip policy route-map IPV4_PBR_Appgroup1
ipv6 policy route-map IPV6_PBR_Appgroup1
interface Vlan20
! tenant SVI appgroup 2
vrf member appgroup
 ip address 20.1.1.1/24
 no ip redirect
 ipv6 address 2001:20:1:1::1/64
 no ipv6 redirects
 fabric forwarding mode anycast-gateway
ip policy route-map IPV4_PBR_Appgroup2
ipv6 policy route-map IPV6_PBR_Appgroup2

```

On the service VTEP, the PBR policy is applied on the tenant VRF SVI. This ensures the traffic post decapsulation will be redirected to firewall.

```
feature pbr
```

```

ipv6 access-list IPV6_App_group_1
10 permit ipv6 any 2001:10:1:1::0/64

```

```

ip access-list IPV4_App_group_1
10 permit ip any 10.1.1.0/24

```

```

ipv6 access-list IPV6_App_group_2
10 permit ipv6 any 2001:20:1:1::0/64

```

```

ip access-list IPV4_App_group_2
10 permit ip any 20.1.1.0/24

```

```

route-map IPV6_PBR_Appgroup1 permit 10
 match ipv6 address IPV6_App_group_2
 set ipv6 next-hop 2001:100:1:1::20 (next hop is that of the firewall)

```

```

route-map IPV6_PBR_Appgroup permit 20
 match ipv6 address IPV6_App_group1
 set ipv6 next-hop 2001:100:1:1::20 (next hop is that of the firewall)

```

```

route-map IPV4_PBR_Appgroup permit 10
 match ip address IPV4_App_group_2
 set ip next-hop 10.100.1.20 (next hop is that of the firewall)

```

```

route-map IPV4_PBR_Appgroup permit 20
 match ip address IPV4_App_group_1
 set ip next-hop 10.100.1.20 (next hop is that of the firewall)

```

```

interface vlan1000
!L3VNI SVI for Tenant VRF
vrf member appgroup
ip forward
ipv6 forward
ipv6 ipv6 address use-link-local-only
ip policy route-map IPV4_PBR_Appgroup
ipv6 policy route-map IPV6_PBR_Appgroup

```





# CHAPTER 16

## Configuring ACL

This chapter contains the following sections:

- [About Access Control Lists, on page 231](#)
- [Guidelines and Limitations for VXLAN ACLs, on page 233](#)
- [VXLAN Tunnel Encapsulation Switch, on page 234](#)
- [VXLAN Tunnel Decapsulation Switch, on page 238](#)

## About Access Control Lists

**Table 7: ACL Options That Can Be Used for VXLAN Traffic on Cisco Nexus 92300YC, 92160YC-X, 93120TX, 9332PQ, and 9348GC-FXP Switches**

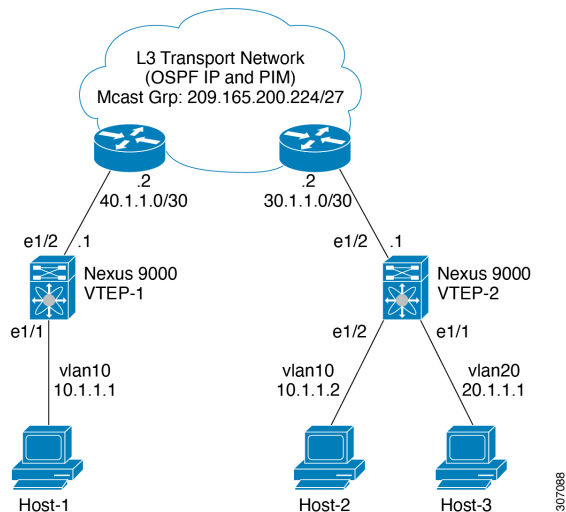
| Scenario | ACL Direction | ACL Type | VTEP Type    | Port Type           | Flow Direction                              | Traffic Type                      | Supported |
|----------|---------------|----------|--------------|---------------------|---------------------------------------------|-----------------------------------|-----------|
| 1        | Ingress       | PACL     | Ingress VTEP | L2 port             | Access to Network<br>[GROUPencap direction] | Native L2 traffic<br>[GROUPinner] | YES       |
| 2        |               | VACL     | Ingress VTEP | VLAN                | Access to Network<br>[GROUPencap direction] | Native L2 traffic<br>[GROUPinner] | YES       |
| 3        | Ingress       | RACL     | Ingress VTEP | Tenant L3 SVI       | Access to Network<br>[GROUPencap direction] | Native L3 traffic<br>[GROUPinner] | YES       |
| 4        | Egress        | RACL     | Ingress VTEP | uplink L3/L3-PO/SVI | Access to Network<br>[GROUPencap direction] | VXLAN encap<br>[GROUPouter]       | NO        |

| Scenario | ACL Direction | ACL Type | VTEP Type   | Port Type           | Flow Direction                               | Traffic Type                           | Supported |
|----------|---------------|----------|-------------|---------------------|----------------------------------------------|----------------------------------------|-----------|
| 5        | Ingress       | RACL     | Egress VTEP | Uplink L3/L3-PO/SVI | Network to Access<br>[GROUP:decap direction] | VXLAN encap<br>[GROUP:outer]           | NO        |
| 6        | Egress        | PACL     | Egress VTEP | L2 port             | Network to Access<br>[GROUP:decap direction] | Native L2 traffic<br>[GROUP:inner]     | NO        |
| 7a       |               | VACL     | Egress VTEP | VLAN                | Network to Access<br>[GROUP:decap direction] | Native L2 traffic<br>[GROUP:inner]     | YES       |
| 7b       |               | VACL     | Egress VTEP | Destination VLAN    | Network to Access<br>[GROUP:decap direction] | Native L3 traffic<br>[GROUP:inner]     | YES       |
| 8        | Egress        | RACL     | Egress VTEP | Tenant L3 SVI       | Network to Access<br>[GROUP:decap direction] | Post-decap L3 traffic<br>[GROUP:inner] | YES       |

ACL implementation for VXLAN is the same as regular IP traffic. The host traffic is not encapsulated in the ingress direction at the encapsulation switch. The implementation is a bit different for the VXLAN encapsulated traffic at the decapsulation switch as the ACL classification is based on the inner payload. The supported ACL scenarios for VXLAN are explained in the following topics and the unsupported cases are also covered for both encapsulation and decapsulation switches.

All scenarios that are mentioned in the previous table are explained with the following host details:

Figure 25: Port ACL on VXLAN Encap Switch



- Host-1: 10.1.1.1/24 VLAN-10
- Host-2: 10.1.1.2/24 VLAN-10
- Host-3: 20.1.1.1/24 VLAN-20
- Case 1: Layer 2 traffic/L2 VNI that flows between Host-1 and Host-2 on VLAN-10.
- Case 2: Layer 3 traffic/L3 VNI that flows between Host-1 and Host-3 on VLAN-10 and VLAN-20.

## Guidelines and Limitations for VXLAN ACLs

VXLAN ACLs have the following guidelines and limitations:

- A router ACL (RACL) on an SVI of the incoming VLAN-10 and the uplink port (eth1/2) does not support filtering the encapsulated VXLAN traffic with outer or inner headers in an egress direction. The limitation also applies to the Layer 3 port-channel uplink interfaces.
- A router ACL (RACL) on an SVI and the Layer 3 uplink ports is not supported to filter the encapsulated VXLAN traffic with outer or inner headers in an ingress direction. This limitation also applies to the Layer 3 port-channel uplink interfaces.
- A port ACL (PACL) cannot be applied on the Layer 2 port to which a host is connected. Cisco NX-OS does not support a PACL in the egress direction.

# VXLAN Tunnel Encapsulation Switch

## Port ACL on the Access Port on Ingress

You can apply a port ACL (PACL) on the Layer 2 trunk or access port that a host is connected on the encapsulating switch. As the incoming traffic from access to the network is normal IP traffic. The ACL that is being applied on the Layer 2 port can filter it as it does for any IP traffic in the non-VXLAN environment.

The **ing-ifacl** TCAM region must be carved as follows:

### Procedure

|               | Command or Action                                                                                                                                                   | Purpose                                                                                                                                                                                                                                                                                                  |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 1</b> | <b>configure terminal</b><br><br><b>Example:</b><br><code>switch# configure terminal</code>                                                                         | Enters global configuration mode.                                                                                                                                                                                                                                                                        |
| <b>Step 2</b> | <b>hardware access-list tcam region ing-ifacl 256</b><br><br><b>Example:</b><br><code>switch(config)# hardware access-list tcam region ing-ifacl 256</code>         | Attaches the UDFs to the <b>ing-ifacl</b> TCAM region, which applies to IPv4 or IPv6 port ACLs.                                                                                                                                                                                                          |
| <b>Step 3</b> | <b>ip access-list name</b><br><br><b>Example:</b><br><code>switch(config)# ip access list PACL_On_Host_Port</code>                                                  | Creates an IPv4 ACL and enters IP ACL configuration mode. The name arguments can be up to 64 characters.                                                                                                                                                                                                 |
| <b>Step 4</b> | <b>sequence-number permit ip source-address destination-address</b><br><br><b>Example:</b><br><code>switch(config-acl)# 10 permit ip 10.1.1.1/32 10.1.1.2/32</code> | Creates an ACL rule that permits or denies IPv4 traffic matching its condition.<br><br>The <i>source-address destination-address</i> arguments can be the IP address with a network wildcard, the IP address and variable-length subnet mask, the host address, and <b>any</b> to designate any address. |
| <b>Step 5</b> | <b>exit</b><br><br><b>Example:</b><br><code>switch(config-acl)# exit</code>                                                                                         | Exits IP ACL configuration mode.                                                                                                                                                                                                                                                                         |
| <b>Step 6</b> | <b>interface ethernet slot/port</b><br><br><b>Example:</b><br><code>switch(config)# interface ethernet1/1</code>                                                    | Enters interface configuration mode.                                                                                                                                                                                                                                                                     |

|                | Command or Action                                                                                                                                    | Purpose                                                                                                                                                                                                                  |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 7</b>  | <b>ip port access-group</b> <i>acl-name</i><br><b>Example:</b><br>switch(config-if) # <b>ip port access-group</b> <b>PACL_On_Host_Port</b> <b>in</b> | Applies a Layer 2 PACL to the interface. Only inbound filtering is supported with port ACLs. You can apply one port ACL to an interface.                                                                                 |
| <b>Step 8</b>  | <b>switchport</b><br><b>Example:</b><br>switch(config-if) # <b>switchport</b>                                                                        | Configures the interface as a Layer 2 interface.                                                                                                                                                                         |
| <b>Step 9</b>  | <b>switchport mode trunk</b><br><b>Example:</b><br>switch(config-if) # <b>switchport mode trunk</b>                                                  | Configures the interface as a Layer 2 trunk port.                                                                                                                                                                        |
| <b>Step 10</b> | <b>switchport trunk allowed vlan</b> <i>vlan-list</i><br><b>Example:</b><br>switch(config-if) # <b>switchport trunk allowed vlan</b> <b>10,20</b>    | Sets the allowed VLANs for the trunk interface. The default is to allow all VLANs on the trunk interface, 1 through 3967 and 4048 through 4094. VLANs 3968 through 4047 are the default VLANs reserved for internal use. |
| <b>Step 11</b> | <b>no shutdown</b><br><b>Example:</b><br>switch(config-if) # <b>no shutdown</b>                                                                      | Negates the <b>shutdown</b> command.                                                                                                                                                                                     |

## VLAN ACL on the Server VLAN

A VLAN ACL (VACL) can be applied on the incoming VLAN-10 that the host is connected to on the encaps switch. As the incoming traffic from access to network is normal IP traffic, the ACL that is being applied to VLAN-10 can filter it as it does for any IP traffic in the non-VXLAN environment. For more information on VACL, see [About Access Control Lists, on page 231](#).

### Procedure

|               | Command or Action                                                                                                         | Purpose                                                                                                  |
|---------------|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|
| <b>Step 1</b> | <b>configure terminal</b><br><b>Example:</b><br>switch# <b>configure terminal</b>                                         | Enters global configuration mode.                                                                        |
| <b>Step 2</b> | <b>ip access-list</b> <i>name</i><br><b>Example:</b><br>switch(config) # <b>ip access list</b> <b>Vacl_On_Source_VLAN</b> | Creates an IPv4 ACL and enters IP ACL configuration mode. The name arguments can be up to 64 characters. |
| <b>Step 3</b> | <i>sequence-number</i> <b>permit ip</b> <i>source-address</i> <i>destination-address</i>                                  | Creates an ACL rule that permits or denies IPv4 traffic matching its condition.                          |

|               | Command or Action                                                                                                                                                 | Purpose                                                                                                                                                                                                                                                                                                        |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|               | <b>Example:</b><br><pre>switch(config-acl)# 10 permit ip 10.1.1.1 10.1.1.2</pre>                                                                                  | The <i>source-address destination-address</i> arguments can be the IP address with a network wildcard, the IP address and variable-length subnet mask, the host address, and <b>any</b> to designate any address.                                                                                              |
| <b>Step 4</b> | <b>vlan access-map</b> <i>map-name</i> [ <i>sequence-number</i> ]<br><br><b>Example:</b><br><pre>switch(config-acl)# vlan access-map Vacl_on_Source_Vlan 10</pre> | Enters VLAN access-map configuration mode for the VLAN access map specified. If the VLAN access map does not exist, the device creates it.<br><br>If you do not specify a sequence number, the device creates a new entry whose sequence number is 10 greater than the last sequence number in the access map. |
| <b>Step 5</b> | <b>match ip address</b> <i>ip-access-list</i><br><br><b>Example:</b><br><pre>switch(config-acl)# match ip address Vacl_on_Source_Vlan</pre>                       | Specifies an ACL for the access-map entry.                                                                                                                                                                                                                                                                     |
| <b>Step 6</b> | <b>action forward</b><br><br><b>Example:</b><br><pre>switch(config-acl)# action forward</pre>                                                                     | Specifies the action that the device applies to traffic that matches the ACL.                                                                                                                                                                                                                                  |
| <b>Step 7</b> | <b>vlan access-map</b> <i>name</i><br><br><b>Example:</b><br><pre>switch(config-acl)# vlan access map Vacl_on_Source_Vlan</pre>                                   | Enters VLAN access-map configuration mode for the VLAN access map specified.                                                                                                                                                                                                                                   |

## Routed ACL on an SVI on Ingress

A router ACL (RACL) in the ingress direction can be applied on an SVI of the incoming VLAN-10 that the host that connects to the encapsulating switch. As the incoming traffic from access to network is normal IP traffic, the ACL that is being applied on SVI 10 can filter it as it does for any IP traffic in the non-VXLAN environment.

The **ing-racl** TCAM region must be carved as follows:

### Procedure

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



|                | Command or Action                                                                                                                                                 | Purpose                                                                                                                                                                                                                                                                                                         |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 2</b>  | <b>hardware access-list tcam region ing-ifacl 256</b><br><br><b>Example:</b><br><pre>switch(config)# hardware access-list tcam region ing-ifacl 256</pre>         | Attaches the UDFs to the <b>ing-racl</b> TCAM region, which applies to IPv4 or IPv6 port ACLs.                                                                                                                                                                                                                  |
| <b>Step 3</b>  | <b>ip access-list name</b><br><br><b>Example:</b><br><pre>switch(config)# ip access list PACL_On_Host_Port</pre>                                                  | Creates an IPv4 ACL and enters IP ACL configuration mode. The name arguments can be up to 64 characters.                                                                                                                                                                                                        |
| <b>Step 4</b>  | <i>sequence-number permit ip source-address destination-address</i><br><br><b>Example:</b><br><pre>switch(config-acl)# 10 permit ip 10.1.1.1/32 10.1.1.2/32</pre> | <p>Creates an ACL rule that permits or denies IPv4 traffic matching its condition.</p> <p>The <i>source-address destination-address</i> arguments can be the IP address with a network wildcard, the IP address and variable-length subnet mask, the host address, and <b>any</b> to designate any address.</p> |
| <b>Step 5</b>  | <b>exit</b><br><br><b>Example:</b><br><pre>switch(config-acl)# exit</pre>                                                                                         | Exits IP ACL configuration mode.                                                                                                                                                                                                                                                                                |
| <b>Step 6</b>  | <b>interface ethernet slot/port</b><br><br><b>Example:</b><br><pre>switch(config)# interface ethernet1/1</pre>                                                    | Enters interface configuration mode.                                                                                                                                                                                                                                                                            |
| <b>Step 7</b>  | <b>no shutdown</b><br><br><b>Example:</b><br><pre>switch(config-if)# no shutdown</pre>                                                                            | Negates <b>shutdown</b> command.                                                                                                                                                                                                                                                                                |
| <b>Step 8</b>  | <b>ip access-group pacl-name in</b><br><br><b>Example:</b><br><pre>switch(config-if)# ip port access-group Racl_On_Source_Vlan_SVI in</pre>                       | Applies a Layer 2 PACL to the interface. Only inbound filtering is supported with port ACLs. You can apply one port ACL to an interface.                                                                                                                                                                        |
| <b>Step 9</b>  | <b>vrf member vxlan-number</b><br><br><b>Example:</b><br><pre>switch(config-if)# vrf member Cust-A</pre>                                                          | Configure SVI for host.                                                                                                                                                                                                                                                                                         |
| <b>Step 10</b> | <b>no ip redirects</b><br><br><b>Example:</b><br><pre>switch(config-if)# no ip redirects</pre>                                                                    | Prevents the device from sending redirects.                                                                                                                                                                                                                                                                     |
| <b>Step 11</b> | <b>ip address ip-address</b><br><br><b>Example:</b>                                                                                                               | Configures an IP address for this interface.                                                                                                                                                                                                                                                                    |

|                | Command or Action                                                                                                                          | Purpose                                                        |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------|
|                | <code>switch(config-if)# ip address 10.1.1.10</code>                                                                                       |                                                                |
| <b>Step 12</b> | <b>no ipv6 redirects</b><br><b>Example:</b><br><code>switch(config-if)# no ipv6 redirects</code>                                           | Disables the ICMP redirect messages on BFD-enabled interfaces. |
| <b>Step 13</b> | <b>fabric forwarding mode anycast-gateway</b><br><b>Example:</b><br><code>switch(config-if)# fabric forwarding mode anycast-gateway</code> | Configure Anycast gateway forwarding mode.                     |

## Routed ACL on the Uplink on Egress

A RACL on an SVI of the incoming VLAN-10 and the uplink port (eth1/2) is not supported to filter the encapsulated VXLAN traffic with an outer or inner header in an egress direction. This limitation also applies to the Layer 3 port-channel uplink interfaces.

## VXLAN Tunnel Decapsulation Switch

### Routed ACL on the Uplink on Ingress

A RACL on a SVI and the Layer 3 uplink ports is not supported to filter the encapsulated VXLAN traffic with outer or inner header in an ingress direction. This limitation also applies to the Layer 3 port-channel uplink interfaces.

### Port ACL on the Access Port on Egress

Do not apply a PACL on the Layer 2 port to which a host is connected. Cisco Nexus 9000 Series switches do not support a PACL in the egress direction.

### VLAN ACL for the Layer 2 VNI Traffic

A VLAN ACL (VACL) can be applied on VLAN-10 to filter with the inner header when the Layer 2 VNI traffic is flowing from Host-1 to Host-2. For more information on VACL, see [About Access Control Lists, on page 231](#).

The VACL TCAM region must be carved as follows:

#### Procedure

|               | Command or Action                            | Purpose                           |
|---------------|----------------------------------------------|-----------------------------------|
| <b>Step 1</b> | <b>configure terminal</b><br><b>Example:</b> | Enters global configuration mode. |

|               | Command or Action                                                                                                                                                          | Purpose                                                                                                                                                                                                                                                                                                               |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|               | <code>switch# configure terminal</code>                                                                                                                                    |                                                                                                                                                                                                                                                                                                                       |
| <b>Step 2</b> | <b>hardware access-list tcam region vACL 256</b><br><br><b>Example:</b><br><code>switch(config)# hardware access-list tcam region vACL 256</code>                          | Changes the ACL TCAM region size.                                                                                                                                                                                                                                                                                     |
| <b>Step 3</b> | <b>ip access-list name</b><br><br><b>Example:</b><br><code>switch(config)# ip access list VXLAN-L2-VNI</code>                                                              | Creates an IPv4 ACL and enters IP ACL configuration mode. The name arguments can be up to 64 characters.                                                                                                                                                                                                              |
| <b>Step 4</b> | <b>statistics per-entry</b><br><br><b>Example:</b><br><code>switch(config-acl)# statistics per-entry</code>                                                                | Specifies that the device maintains global statistics for packets that match the rules in the VACL.                                                                                                                                                                                                                   |
| <b>Step 5</b> | <b>sequence-number permit ip source-address destination-address</b><br><br><b>Example:</b><br><code>switch(config-acl)# 10 permit ip 10.1.1.1/32 10.1.1.2/32</code>        | <p>Creates an ACL rule that permits or denies IPv4 traffic matching its condition.</p> <p>The <i>source-address destination-address</i> arguments can be the IP address with a network wildcard, the IP address and variable-length subnet mask, the host address, and <b>any</b> to designate any address.</p>       |
| <b>Step 6</b> | <b>sequence-number permit protocol source-address destination-address</b><br><br><b>Example:</b><br><code>switch(config-acl)# 20 permit tcp 10.1.1.2/32 10.1.1.1/32</code> | <p>Creates an ACL rule that permits or denies IPv4 traffic matching its condition.</p> <p>The <i>source-address destination-address</i> arguments can be the IP address with a network wildcard, the IP address and variable-length subnet mask, the host address, and <b>any</b> to designate any address.</p>       |
| <b>Step 7</b> | <b>exit</b><br><br><b>Example:</b><br><code>switch(config-acl)# exit</code>                                                                                                | Exit ACL configuration mode.                                                                                                                                                                                                                                                                                          |
| <b>Step 8</b> | <b>vlan access-map map-name [sequence-number]</b><br><br><b>Example:</b><br><code>switch(config)# vlan access-map VXLAN-L2-VNI 10</code>                                   | <p>Enters VLAN access-map configuration mode for the VLAN access map specified. If the VLAN access map does not exist, the device creates it.</p> <p>If you do not specify a sequence number, the device creates a new entry whose sequence number is 10 greater than the last sequence number in the access map.</p> |
| <b>Step 9</b> | <b>match ip address list-name</b><br><br><b>Example:</b>                                                                                                                   | Configure the IP list name.                                                                                                                                                                                                                                                                                           |

|  | Command or Action                                                 | Purpose |
|--|-------------------------------------------------------------------|---------|
|  | <code>switch(config-access-map)# match ip<br/>VXLAN-L2-VNI</code> |         |

## VLAN ACL for the Layer 3 VNI Traffic

A VLAN ACL (VACL) can be applied on the destination VLAN-20 to filter with the inner header when the Layer 3 VNI traffic is flowing from Host-1 to Host-3. It slightly differs from the previous case as the VACL for the Layer 3 traffic is accounted on the egress on the system. The keyword **output** must be used while dumping the VACL entries for the Layer 3 VNI traffic. For more information on VACL, see [About Access Control Lists, on page 231](#).

The VACL TCAM region must be carved as follows.

### Procedure

|               | Command or Action                                                                                                                                                              | Purpose                                                                                                                                                                                                                                                                                                  |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 1</b> | <b>configure terminal</b><br><br><b>Example:</b><br><code>switch# configure terminal</code>                                                                                    | Enters global configuration mode.                                                                                                                                                                                                                                                                        |
| <b>Step 2</b> | <b>hardware access-list tcam region vacl 256</b><br><br><b>Example:</b><br><code>switch(config)# hardware access-list tcam<br/>region vacl 256</code>                          | Changes the ACL TCAM region size.                                                                                                                                                                                                                                                                        |
| <b>Step 3</b> | <b>ip access-list name</b><br><br><b>Example:</b><br><code>switch(config)# ip access list<br/>VXLAN-L3-VNI</code>                                                              | Creates an IPv4 ACL and enters IP ACL configuration mode. The name arguments can be up to 64 characters.                                                                                                                                                                                                 |
| <b>Step 4</b> | <b>statistics per-entry</b><br><br><b>Example:</b><br><code>switch(config)# statistics per-entry</code>                                                                        | Specifies that the device maintains global statistics for packets that match the rules in the VACL.                                                                                                                                                                                                      |
| <b>Step 5</b> | <b>sequence-number permit ip source-address destination-address</b><br><br><b>Example:</b><br><code>switch(config-acl)# 10 permit ip<br/>10.1.1.1/32 20.1.1.1/32</code>        | Creates an ACL rule that permits or denies IPv4 traffic matching its condition.<br><br>The <i>source-address destination-address</i> arguments can be the IP address with a network wildcard, the IP address and variable-length subnet mask, the host address, and <b>any</b> to designate any address. |
| <b>Step 6</b> | <b>sequence-number permit protocol source-address destination-address</b><br><br><b>Example:</b><br><code>switch(config-acl)# 20 permit tcp<br/>20.1.1.1/32 10.1.1.1/32</code> | Configures the ACL to redirect-specific HTTP methods to a server.                                                                                                                                                                                                                                        |

|               | Command or Action                                                                                                                                          | Purpose                                                                                                                                                                                                                                                                                                        |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 7</b> | <b>vlan access-map</b> <i>map-name</i> [ <i>sequence-number</i> ]<br><br><b>Example:</b><br><pre>switch(config-acl)# vlan access-map VXLAN-L3-VNI 10</pre> | Enters VLAN access-map configuration mode for the VLAN access map specified. If the VLAN access map does not exist, the device creates it.<br><br>If you do not specify a sequence number, the device creates a new entry whose sequence number is 10 greater than the last sequence number in the access map. |
| <b>Step 8</b> | <b>action forward</b><br><br><b>Example:</b><br><pre>switch(config-acl)# action forward</pre>                                                              | Specifies the action that the device applies to traffic that matches the ACL.                                                                                                                                                                                                                                  |

## Routed ACL on an SVI on Egress

A router ACL (RACL) on the egress direction can be applied on an SVI of the destination VLAN-20 that Host-3 is connected to on the decap switch to filter with the inner header for traffic flows from the network to access which is normal post-decapsulated IP traffic post. The ACL that is being applied on SVI 20 can filter it as it does for any IP traffic in the non-VXLAN environment. For more information on ACL, see [About Access Control Lists, on page 231](#).

The egr-racl TCAM region must be carved as follows:

### Procedure

|               | Command or Action                                                                                                                                                                      | Purpose                                                                                                                                                                                                                 |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 1</b> | <b>configure terminal</b><br><br><b>Example:</b><br><pre>switch# configure terminal</pre>                                                                                              | Enters global configuration mode.                                                                                                                                                                                       |
| <b>Step 2</b> | <b>hardware access-list tcam region egr-racl 256</b><br><br><b>Example:</b><br><pre>switch(config)# hardware access-list tcam region egr-racl 256</pre>                                | Changes the ACL TCAM region size.                                                                                                                                                                                       |
| <b>Step 3</b> | <b>ip access-list</b> <i>name</i><br><br><b>Example:</b><br><pre>switch(config)# ip access-list Racl_on_Source_Vlan_SVI</pre>                                                          | Creates an IPv4 ACL and enters IP ACL configuration mode. The name arguments can be up to 64 characters.                                                                                                                |
| <b>Step 4</b> | <i>sequence-number</i> <b>permit ip</b> <i>source-address</i> <i>destination-address</i><br><br><b>Example:</b><br><pre>switch(config-acl)# 10 permit ip 10.1.1.1/32 20.1.1.1/32</pre> | Creates an ACL rule that permits or denies IPv4 traffic matching its condition.<br><br>The <i>source-address</i> <i>destination-address</i> arguments can be the IP address with a network wildcard, the IP address and |

|                | Command or Action                                                                                                                                 | Purpose                                                                                                                                           |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
|                |                                                                                                                                                   | variable-length subnet mask, the host address, and <b>any</b> to designate any address.                                                           |
| <b>Step 5</b>  | <b>interface vlan</b> <i>vlan-id</i><br><b>Example:</b><br><pre>switch(config-acl)# interface vlan vlan20</pre>                                   | Enters interface configuration mode, where <i>vlan-id</i> is the ID of the VLAN that you want to configure with a DHCP server IP address.         |
| <b>Step 6</b>  | <b>no shutdown</b><br><b>Example:</b><br><pre>switch(config-if)# no shutdown</pre>                                                                | Negate the shutdown command.                                                                                                                      |
| <b>Step 7</b>  | <b>ip access-group</b> <i>access-list out</i><br><b>Example:</b><br><pre>switch(config-if)# ip access-group Racl_On_Detination_Vlan_SVI out</pre> | Applies an IPv4 or IPv6 ACL to the Layer 3 interfaces for traffic flowing in the direction specified. You can apply one router ACL per direction. |
| <b>Step 8</b>  | <b>vrf member</b> <i>vxlان-number</i><br><b>Example:</b><br><pre>switch(config-if)# vrf member Cust-A</pre>                                       | Configure SVI for host.                                                                                                                           |
| <b>Step 9</b>  | <b>no ip redirects</b><br><b>Example:</b><br><pre>switch(config-if)# no ip redirects</pre>                                                        | Prevents the device from sending redirects.                                                                                                       |
| <b>Step 10</b> | <b>ip address</b> <i>ip-address/length</i><br><b>Example:</b><br><pre>switch(config-if)# ip address 20.1.1.10/24</pre>                            | Configures an IP address for this interface.                                                                                                      |
| <b>Step 11</b> | <b>no ipv6 redirects</b><br><b>Example:</b><br><pre>switch(config-if)# no ipv6 redirects</pre>                                                    | Disables the ICMP redirect messages on BFD-enabled interfaces.                                                                                    |
| <b>Step 12</b> | <b>fabric forwarding mode anycast-gateway</b><br><b>Example:</b><br><pre>switch(config-if)# fabric forwarding mode anycast-gateway</pre>          | Configure Anycast gateway forwarding mode.                                                                                                        |



## CHAPTER 17

# Configuring VXLAN QoS

---

This chapter contains the following sections:

- [Information About VXLAN QoS, on page 243](#)
- [Guidelines and Limitations for VXLAN QoS, on page 251](#)
- [Default Settings for VXLAN QoS, on page 252](#)
- [Configuring VXLAN QoS, on page 253](#)
- [Verifying the VXLAN QoS Configuration, on page 255](#)
- [VXLAN QoS Configuration Examples, on page 255](#)

## Information About VXLAN QoS

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

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

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



---

**Note**

QoS allows you to classify the network traffic, police and prioritize the traffic flow, and provide congestion avoidance. For more information about QoS, see the [Cisco Nexus 9000 Series NX-OS Quality of Service Configuration Guide, Release 9.2\(x\)](#).

---

This section contains the following topics:

## VXLAN QoS Terminology

This section defines VXLAN QoS terminology.

Table 8: VXLAN QoS Terminology

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



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

## VXLAN QoS Features

The following topics describe the VXLAN QoS features that are supported in a VXLAN network:

### Trust Boundaries

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

### Classification

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

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

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

### Marking

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

### Policing

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

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

## Queuing and Scheduling

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

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

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

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

## Traffic Shaping

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

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

By using traffic shaping, you can control the following:

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

For example, you can control access to the bandwidth when the policy dictates that the rate of a given interface must not, on average, exceed a certain rate. Despite the access rate exceeding the speed.

## Network QoS

The network QoS policy defines the characteristics of each CoS value, which are applicable network wide across switches. With a network QoS policy, you can configure the following:

- **Pause behavior**—You can decide whether a CoS requires the lossless behavior which is provided by using a priority flow control (PFC) mechanism that prevents packet loss during congestion) or not. You can configure drop (frames with this CoS value can be dropped) and no drop (frames with this CoS value cannot be dropped). For the drop and no drop configuration, you must also enable PFC per port. For more information about PFC, see “Configuring Priority Flow Control”.

Pause behavior can be achieved in the VXLAN tunnel for a specific queue-group.

## VXLAN Priority Tunneling

In the VXLAN tunnel, DSCP values in the outer header are used to provide QoS transparency in end-to-end of the tunnel. The outer header DSCP value is derived from the DSCP value with Layer 3 packets or the CoS value for Layer 2 frames. At the VXLAN tunnel egress point, the priority of the decapsulated traffic is chosen based on the mode. For more information, see [Decapsulated Packet Priority Selection](#), on page 250.

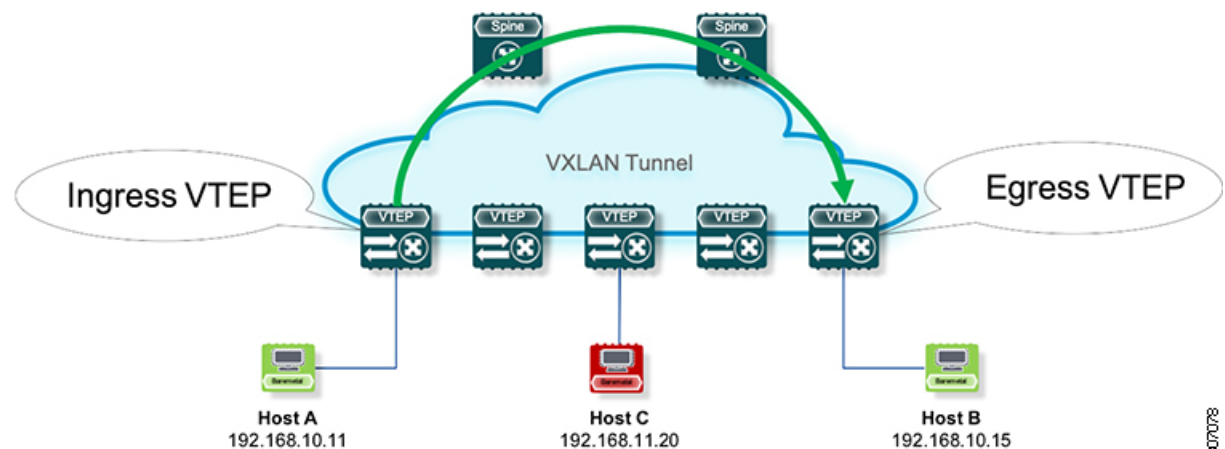
## MQC CLI

All available QoS features for VXLAN QoS are managed from the modular QoS command-line interface (CLI). The Modular QoS CLI (MQC) allows you to define traffic classes (class maps), create and configure traffic policies (policy maps), and perform actions that are defined in the policy maps to interface (service policy).

## VXLAN QoS Topology and Roles

This section describes the roles of network devices in implementing VXLAN QoS.

**Figure 26: VXLAN Network**



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

In the VXLAN network, points of interest are ingress VTEPs where the original traffic is encapsulated in a VXLAN header. Spines are transporting hops that connect ingress and egress VTEPs. An egress VTEP is the point where VXLAN encapsulated traffic is decapsulated and egresses the VTEP as classical Ethernet traffic.



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

This section contains the following topics:

### Ingress VTEP and Encapsulation in the VXLAN Tunnel

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

**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 processes packets as follows:

**Procedure**

- 
- Step 1** Packets encapsulated in VXLAN are received at the NVE interface of an egress VTEP, where the switch uses the inner header DSCP value to perform classification, marking, and policing.
  - Step 2** The switch removes the VXLAN header from the packet, and does a lookup that is based on the decapsulated packet's headers.
  - Step 3** The switch forwards the decapsulated packets to the appropriate output interface for processing.
  - Step 4** Before the packet is sent out, a DSCP value is assigned to a Layer 3 packet based on the decapsulation priority or based on marking Layer 2 frames.
  - Step 5** The decapsulated packets are sent through the outgoing interface to the IP network.
-

## Classification at the Ingress VTEP, Spine, and Egress VTEP

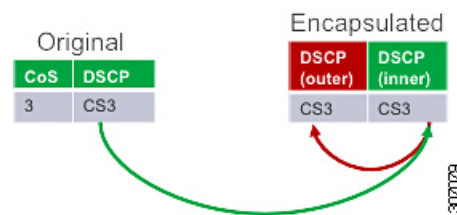
This section includes the following topics:

### IP to VXLAN

At the ingress VTEP, the ingress point of the VXLAN tunnel, traffic is encapsulated in the VXLAN header. Traffic on an ingress VTEP is classified based on the priority in the original header. Classification can be performed by matching the CoS, DSCP, and IP precedence values or by matching traffic with the ACL based on the original frame data.

When traffic is encapsulated in the VXLAN, the Layer 3 packet's DSCP value is copied from the original header to the outer header of the VXLAN encapsulated packet. This behavior is illustrated in the following figure:

**Figure 27: 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 252](#). In this way, the original QoS attributes are preserved in the VXLAN tunnel. This behavior is illustrated in the following figure:

**Figure 28: Copy of Priority from Layer-2 Frame to VXLAN Outer Header**



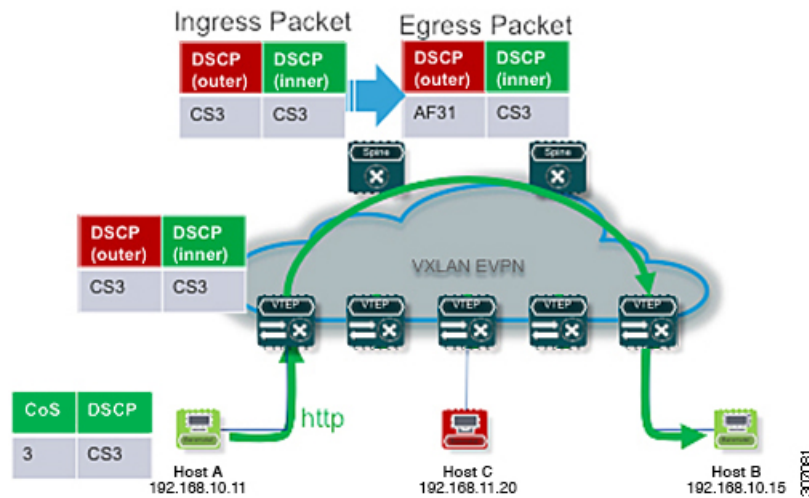
A Layer 2 frame, does not have a DSCP value present because the IP header is not present in the frame. After a Layer 2 frame is encapsulated, the original CoS value is not preserved in the VXLAN tunnel.

### 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 29: Marking Inside of the VXLAN Tunnel



## VXLAN to IP

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

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

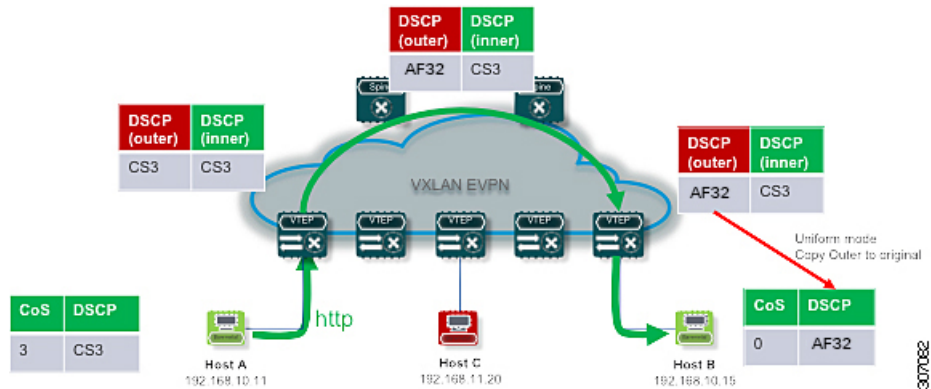
Marking and policing can be performed on the NVE interface for tunneled traffic. If marking is configured, newly marked values are present in the decapsulated packet. Because the original CoS value is not preserved in the encapsulated packet, marking can be performed for decapsulated packets for any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

## Decapsulated Packet Priority Selection

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

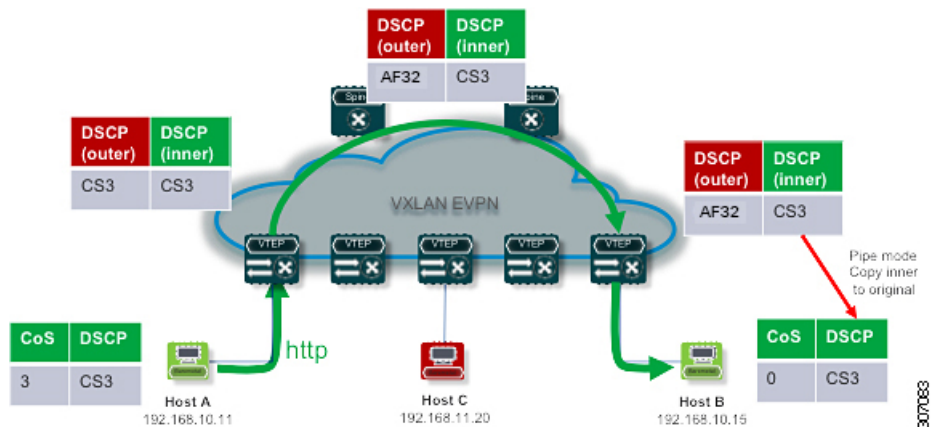
- Uniform mode – the DSCP value from the outer header of the VXLAN packet is copied to the decapsulated packet. Any change of the DSCP value in the VXLAN tunnel is preserved and present in the decapsulated packet. Uniform mode is the default mode of decapsulated packet priority selection.

**Figure 30: Uniform Mode Outer DSCP Value is Copied to Decapsulated Packet DSCP Value for a Layer-3 Packet**



- Pipe mode – the original DSCP value is preserved at the VXLAN tunnel end. At the egress VTEP, the system copies the inner DSCP value to the decapsulated packet DSCP value. In this way, the original DSCP value is preserved at the end of the VXLAN tunnel.

**Figure 31: 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 9.2(3), 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.
- In RX series line cards, the default mode is pipe for VXLAN decapsulation (inner packet DSCP not modified based on outer IP header DSCP value). This is a difference in behavior from other line cards types. If RX series line cards and other line cards are used in the same network, the **qos-mode pipe** command can be used in switches where non-RX line cards are present in order to have the same behavior. For details of the configuration command, see [Configuring Type QoS on the Egress VTEP, on page 253](#).
- 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                          |



| CoS of Original Layer 2 Frame | DSCP of Outer VXLAN Header |
|-------------------------------|----------------------------|
| 1                             | 8                          |
| 2                             | 16                         |
| 3                             | 26                         |
| 4                             | 32                         |
| 5                             | 46                         |
| 6                             | 48                         |
| 7                             | 56                         |

## Configuring VXLAN QoS

Configuration of VXLAN QoS is done using the MQC model. The same configuration that is used for the QoS configuration applies to VXLAN QoS. For more information about configuring QoS, see the [Cisco Nexus 9000 Series NX-OS Quality of Service Configuration Guide, Release 9.2\(x\)](#).

VXLAN QoS introduces a new service-policy attachment point which is NVE – Network Virtual Interface. At the egress VTEP, the NVE interface is the point where traffic is decapsulated. To account for all VXLAN traffic, the service policy must be attached to an NVE interface.

The next section describes the configuration of the classification at the egress VTEP, and **service-policy type qos** attachment to an NVE interface.

## Configuring Type QoS on the Egress VTEP

Configuration of VXLAN QoS is done by using the MQC model. The same configuration is used for QoS configuration for VXLAN QoS. For more information about configuring QoS, see the [Cisco Nexus 9000 Series NX-OS Quality of Service Configuration Guide, Release 9.2\(x\)](#).

VXLAN QoS introduces a new service-policy attachment point which is the Network Virtual Interface (NVE). At the egress VTEP, the NVE interface points where traffic is decapsulated. To account for all VXLAN traffic, the service policy must be attached to an NVE interface.

This procedure describes the configuration of classification at the egress VTEP, and **service-policy type qos** attachment to an NVE interface.

### Procedure

|               | Command or Action                                                                     | Purpose                                                                                    |
|---------------|---------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------|
| <b>Step 1</b> | <b>configure terminal</b><br><br><b>Example:</b><br>switch# <b>configure terminal</b> | Enters global configuration mode.                                                          |
| <b>Step 2</b> | <b>[no] class-map [type [qos]]   [match-all]   [match-any] class-map-name</b>         | Creates or accesses the class map <i>class--map-name</i> and enters <b>class-map</b> mode. |

|                | Command or Action                                                                                                                                            | Purpose                                                                                                                                                                                                                                                                                                               |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | <b>Example:</b><br><pre>switch(config)# class-map type qos class1</pre>                                                                                      | The <i>class-map-name</i> argument can contain alphabetic, hyphen, or underscore characters, and can be up to 40 characters. (match-any is the default when the <b>no</b> option is selected and multiple match statements are entered.)                                                                              |
| <b>Step 3</b>  | <b>[no] match [access-group   cos   dscp   precedence] {name / 0-7 / 0-63 / 0-7}</b><br><b>Example:</b><br><pre>switch(config-cmap-qos)# match dscp 26</pre> | Configures the traffic class by matching packets based on access-list, <b>cos</b> value, <b>dscp</b> values, or IP <b>precedence</b> value                                                                                                                                                                            |
| <b>Step 4</b>  | <b>[no] policy-map type qos policy-map-name</b><br><b>Example:</b><br><pre>switch(config-cmap-qos)# policy-map type qos policy</pre>                         | Creates or accesses the policy map that is 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.                                                                             |
| <b>Step 5</b>  | <b>[no] class class-name</b><br><b>Example:</b><br><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. |
| <b>Step 6</b>  | <b>[no] set qos-group qos-group-value</b><br><b>Example:</b><br><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 <b>qos-group</b> is referenced in type queuing and type network-qos as matching criteria.                                                                                                                            |
| <b>Step 7</b>  | <b>exit</b><br><b>Example:</b><br><pre>switch(config-pmap-c-qos)# exit</pre>                                                                                 | Exits class-map mode.                                                                                                                                                                                                                                                                                                 |
| <b>Step 8</b>  | <b>[no] interface nve nve-interface-number</b><br><b>Example:</b><br><pre>switch(config)# interface nve 1</pre>                                              | Enters interface mode to configure the NVE interface.                                                                                                                                                                                                                                                                 |
| <b>Step 9</b>  | <b>[no] service-policy type qos input policy-map-name</b><br><b>Example:</b><br><pre>switch(config-if-nve)# service-policy type qos input policy</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.                                                                                                                                                                       |
| <b>Step 10</b> | <b>(Optional) [no] qos-mode [pipe]</b><br><b>Example:</b>                                                                                                    | Selecting decapsulated packet priority selection and using pipe mode. Entering the <b>no</b> form of                                                                                                                                                                                                                  |

|  | Command or Action                                 | Purpose                                                      |
|--|---------------------------------------------------|--------------------------------------------------------------|
|  | <code>switch(config-if-nve)# qos-mode pipe</code> | this command negates pipe mode and defaults to uniform mode. |

## Verifying the VXLAN QoS Configuration

Table 10: VXLAN QoS Verification Commands

| Command                         | Purpose                                                |
|---------------------------------|--------------------------------------------------------|
| <code>show class map</code>     | Displays information about all configured class maps.  |
| <code>show policy-map</code>    | Displays information about all configured policy maps. |
| <code>show running ipqos</code> | 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 queuing** command for traffic in qos-group 1. Assigning 50% of the available bandwidth to q1 mapped to qos-group 1 and attaching policy in the output direction to all ports using the **system qos** command.

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

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



## CHAPTER 18

# Configuring vPC Fabric Peering

This chapter contains the following sections:

- [Information About vPC Fabric Peering, on page 257](#)
- [Guidelines and Limitations for vPC Fabric Peering, on page 258](#)
- [Configuring vPC Fabric Peering, on page 259](#)
- [Migrating from vPC to vPC Fabric Peering, on page 262](#)
- [Verifying vPC Fabric Peering Configuration, on page 264](#)

## Information About vPC Fabric Peering

vPC Fabric Peering provides an enhanced dual-homing access solution without the overhead of wasting physical ports for vPC Peer Link. This feature preserves all the characteristics of a traditional vPC.

The following lists the vPC Fabric Peering solution:

- vPC Fabric Peering port-channel with virtual members (tunnels).
- vPC Fabric Peering (tunnel) with removal of the physical peer link requirement.
- vPC Fabric Peering up/down events are triggered based on route updates and fabric up/down.
- Uplink tracking for extended failure coverage.
- vPC Fabric Peering reachability via the routed network, such as the spine.
- Increased resiliency of the vPC control plane over TCP-IP (CFSolP).
- Data plane traffic over the VXLAN tunnel.
- Communication between vPC member switches uses VXLAN encapsulation.
- Failure of all uplinks on a node result in vPC ports going down on that switch. In that scenario, vPC peer takes up the primary role and forwards the traffic.
- Uplink tracking with state dependency and up/down signaling for vPCs.
- Positive uplink state tracking drives vPC primary role election.
- For border leafs and spines, there is no need for per-VRF peering since network communication uses the fabric.
- Enhance forwarding to orphans hosts by extending the VIP/PIP feature to Type-2 routes.

- Infra-VLAN is not required for vPC fabric peering.



**Note** The vPC Fabric Peering counts as three VTEPs unlike a normal vPC which counts as one VTEP.

## Guidelines and Limitations for vPC Fabric Peering

The following are the vPC Fabric Peering guidelines and limitations:

- Cisco Nexus 9332C, 9364C, and 9300-FX/FXP/FX2 platform switches support vPC Fabric Peering. Cisco Nexus 9200, 9300-EX, and 9500 platform switches do not support vPC Fabric Peering.
- vPC Fabric Peering requires the application of TCAM carving of region "ing-flow-redirect." TCAM carving requires saving the configuration and reloading the switch prior to using the feature.
- Prior to reconfiguring the vPC Fabric Peering source and destination IP, the vPC domain must be shut down. Once the vPC Fabric Peering source and destination IP have been adjusted, the vPC domain can be enabled (**no shutdown**).
- The vPC Fabric Peering peer-link is established over the transport network (the spine layer of the fabric). As communication between vPC peers occurs in this manner, control plane information CFS messages used to synchronize port state information, VLAN information, VLAN-to-VNI mapping, host MAC addresses, and IGMP snooping groups are transmitted over the fabric. CFS messages are marked with the appropriate DSCP value, which should be protected in the transport network. The following example shows a sample QoS configuration on the spine layer of Cisco Nexus 9000 Series switches.

Classify traffic by matching the DSCP value (DSCP 56 is the default value):

```
class-map type qos match-all CFS
 match dscp 56
```

Set traffic to the qos-group that corresponds with the strict priority queue for the appropriate spine switch. In this example, the switch sends traffic to qos-group 7, which corresponds to the strict priority queue (Queue 7). Note that different Cisco Nexus platforms might have a different queueing structure.

```
policy-map type qos CFS
 class CFS
 Set qos-group 7
```

Assign a classification service policy to all interfaces toward the VTEP (the leaf layer of the network):

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

- The vPC Fabric Peering domain does not support attaching FEX to it.
- The vPC Fabric Peering domain is not supported in the role of a Multi-Site vPC BGW.
- Enhance forwarding to orphan hosts by extending the VIP/PIP feature to Type-2 routes.
- Layer 3 Tenant Routed Multicast (TRM) is supported. Layer 2/Layer 3 TRM (Mixed Mode) is not supported.

- If Type-5 routes are used with this feature, the **advertise-pip** command is a mandatory configuration.
- VTEPs behind vPC ports are not supported. This means that virtual peer-link peers cannot act as a transit node for the VTEPs behind the vPC ports.
- SVI and sub-interface uplinks are not supported.
- An orphan Type-2 host is advertised using PIP. A vPC Type-2 host is advertised using VIP. This is the default behavior for a Type-2 host.

To advertise an orphan Type-5 route using PIP, you need to advertise PIP under BGP.

- Traffic from remote VTEP to orphan hosts would land on the actual node which has the orphans. Bouncing of the traffic is avoided.



---

**Note** When the vPC leg is down, vPC hosts are still advertised with the VIP IP.

---

## Configuring vPC Fabric Peering

Ensure the vPC Fabric Peering DSCP value is consistent on both vPC member switches. Ensure that the corresponding QoS policy matches the vPC Fabric Peering DSCP marking.

All VLANs that require communication traversing the vPC Fabric Peering must have a VXLAN enabled (vn-segment); this includes the native VLAN.



---

**Note** For MSTP, VLAN 1 must be extended across vPC Fabric Peering if the peer-link and vPC legs have the default native VLAN configuration. This behavior can be achieved by extending VLAN 1 over VXLAN (vn-segment). If the peer-link and vPC legs have non-default native VLANs, those VLANs must be extended across vPC Fabric Peering by associating the VLANs with VXLAN (vn-segment).

---

Use the **show vpc virtual-peerlink vlan consistency** command for verification of the existing VLAN-to-VXLAN mapping used for vPC Fabric Peering.

**peer-keepalive** command for vPC Fabric Peering is supported with one of the following configurations:

- Management interface
- Dedicated Layer 3 link in default or non-default VRF
- Loopback interface reachable using the spine.

### Configuring Features

Example uses OSPF as the underlay routing protocol.

```
configure terminal
nv overlay evpn
feature ospf
feature bgp
feature pim
feature interface-vlan
```

```
feature vn-segment-vlan-based
feature vpc

feature nv overlay
```

### vPC Configuration



**Note** To change the vPC Fabric Peering source or destination IP, the vPC domain must be shutdown prior to modification. The vPC domain can be returned to operation after the modifying by using the **no shutdown** command.

### Configuring TCAM Carving

```
hardware access-list tcam region ing-racl 0
hardware access-list tcam region ing-sup 768
hardware access-list tcam region ing-flow-redirect 512
```

### Configuring the vPC Domain

```
vpc domain 100
peer-keepalive destination 192.0.2.1
virtual peer-link destination 192.0.2.100 source 192.0.2.20/32 [dscp <dscp-value>]
Warning: Appropriate TCAM carving must be configured for virtual peer-link vPC
peer-switch
peer-gateway
ip arp synchronize
ipv6 nd synchronize
exit
```



**Note** The **dscp** keyword is optional. Range is 1 to 63. The default value is 56.

### Configuring vPC Fabric Peering Port Channel

No need to configure members for the following port channel.

```
interface port-channel 10
switchport
switchport mode trunk
vpc peer-link

interface loopback0
```



**Note** This loopback is not the NVE source-interface loopback (interface used for the VTEP IP address).

```
interface loopback 0
ip address 192.0.2.20/32
ip router ospf 1 area 0.0.0.0
```



**Note** You can use the loopback for BGP peering or a dedicated loopback. This loopback must be different than the loopback for peer keep alive.



### Configuring the Underlay Interfaces

Both L3 physical and L3 port channels are supported. SVI and sub-interfaces are not supported.

```
router ospf 1
interface Ethernet1/16
ip address 192.0.2.2/24
ip router ospf 1 area 0.0.0.0
no shutdown
interface Ethernet1/17
port-type fabric
ip address 192.0.2.3/24
ip router ospf 1 area 0.0.0.0
no shutdown
interface Ethernet1/40
port-type fabric
ip address 192.0.2.4/24
ip router ospf 1 area 0.0.0.0
no shutdown
interface Ethernet1/41
port-type fabric
ip address 192.0.2.5/24
ip router ospf 1 area 0.0.0.0
no shutdown
```




---

**Note** All ports connected to spines must be port-type fabric.

---

### VXLAN Configuration




---

**Note** Configuring **advertise virtual-rmac** (NVE) and **advertise-pip** (BGP) are required steps. For more information, see the [Configuring vPC Multi-Homing, on page 147](#) chapter.

---

### Configuring VLANs and SVI

```
vlan 10
vn-segment 10010
vlan 101
vn-segment 10101
interface Vlan101
no shutdown
mtu 9216
vrf member vxlan-10101
no ip redirects
ip forward
ipv6 address use-link-local-only
no ipv6 redirects
interface vlan10
no shutdown
mtu 9216
vrf member vxlan-10101
no ip redirects
ip address 192.0.2.102/24
ipv6 address 2001:DB8:0:1::1/64
no ipv6 redirects
fabric forwarding mode anycast-gateway
```

### Configuring Virtual Port Channel

```
interface Ethernet1/3
switchport
switchport mode trunk
channel-group 100
no shutdown
exit
interface Ethernet1/39
switchport
switchport mode trunk
channel-group 101
no shutdown
interface Ethernet1/46
switchport
switchport mode trunk
channel-group 102
no shutdown
interface port-channel100
vpc 100
interface port-channel101
vpc 101
interface port-channel102
vpc 102
exit
```

## Migrating from vPC to vPC Fabric Peering

This procedure contains the steps to migration from a regular vPC to vPC Fabric Peering.

Any direct Layer 3 link between vPC peers should be used only for peer-keep alive. This link should not be used to advertise paths for vPC Fabric Peering loopbacks.



**Note** This migration is disruptive.

### Before you begin

We recommend that you shut all physical Layer 2 links between the vPC peers before migration. We also recommend that you map VLANs with vn-segment before or after migration.

### Procedure

|               | Command or Action                                                                     | Purpose                                              |
|---------------|---------------------------------------------------------------------------------------|------------------------------------------------------|
| <b>Step 1</b> | <b>configure terminal</b><br><br><b>Example:</b><br>switch# <b>configure terminal</b> | Enters global configuration mode.                    |
| <b>Step 2</b> | <b>show vpc</b><br><br><b>Example:</b><br>switch(config)# <b>show vpc</b>             | Determine the number of members in the port channel. |

|                | Command or Action                                                                                                                                                                                        | Purpose                                                                                                                        |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 3</b>  | <b>show port-channel summary</b><br><br><b>Example:</b><br><code>switch(config)# show port-channel summary</code>                                                                                        | Determine the number of members.                                                                                               |
| <b>Step 4</b>  | <b>interface ethernet <i>slot/port</i></b><br><br><b>Example:</b><br><code>switch(config)# interface ethernet 1/4</code>                                                                                 | Specifies the interface you are configuring.<br><br><b>Note</b> This is the peer link port channel.                            |
| <b>Step 5</b>  | <b>no channel-group</b><br><br><b>Example:</b><br><code>switch(config-if)# no channel-group</code>                                                                                                       | Remove vPC peer-link port-channel members.<br><br><b>Note</b> Disruption occurs following this step.                           |
| <b>Step 6</b>  | Repeat steps 4 and 5 for each interface.<br><br><b>Example:</b>                                                                                                                                          |                                                                                                                                |
| <b>Step 7</b>  | <b>show running-config vpc</b><br><br><b>Example:</b><br><code>switch(config-if)# show running-config vpc</code>                                                                                         | Determine the vPC domain.                                                                                                      |
| <b>Step 8</b>  | <b>vpc domain <i>domain-id</i></b><br><br><b>Example:</b><br><code>switch(config-if)# vpc domain 100</code>                                                                                              | Enter vPC domain configuration mode.                                                                                           |
| <b>Step 9</b>  | <b>virtual peer-link destination <i>dest-ip</i> source <i>source-ip</i></b><br><br><b>Example:</b><br><code>switch(config-vpc-domain)# virtual peer-link destination 192.0.2.1 source 192.0.2.100</code> | Specify the destination and source IP addresses for vPC fabric peering.                                                        |
| <b>Step 10</b> | <b>interface {ethernet   port-channel} <i>value</i></b><br><br><b>Example:</b><br><code>switch(config-if)# interface Ethernet1/17</code>                                                                 | Specifies the L3 underlay interface you are configuring.                                                                       |
| <b>Step 11</b> | <b>port-type fabric</b><br><br><b>Example:</b><br><code>switch(config-if)# port-type fabric</code>                                                                                                       | Configures port-type fabric for underlay interface.<br><br><b>Note</b> All ports connected to spines must be port-type fabric. |
| <b>Step 12</b> | (Optional) <b>show vpc fabric-ports</b><br><br><b>Example:</b><br><code>switch# show vpc fabric-ports</code>                                                                                             | Displays the fabric ports connected to spine.                                                                                  |

|                | Command or Action                                                                                                                                                                          | Purpose                                                                                                                                            |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 13</b> | <b>hardware access-list tcam region ing-flow-redirect tcam-size</b><br><br><b>Example:</b><br><pre>switch(config-vpc-domain)# hardware access-list tcam region ing-flow-redirect 512</pre> | Perform TCAM carving.<br><br>The minimum size for Ingress-Flow-redirect TCAM region size is 512. Also ensure it is configured in multiples of 512. |
| <b>Step 14</b> | <b>copy running-config startup-config</b><br><br><b>Example:</b><br><pre>switch(config-vpc-domain)# copy running-config startup-config</pre>                                               | Copies the running configuration to the startup configuration.                                                                                     |
| <b>Step 15</b> | <b>reload</b><br><br><b>Example:</b><br><pre>switch(config-vpc-domain)# reload</pre>                                                                                                       | Reboots the switch.                                                                                                                                |

## Verifying vPC Fabric Peering Configuration

To display the status for the vPC Fabric Peering configuration, enter one of the following commands:

**Table 11: vPC Fabric Peering Verification Commands**

| Command                                           | Purpose                                                      |
|---------------------------------------------------|--------------------------------------------------------------|
| <b>show vpc fabric-ports</b>                      | Displays the fabric ports state.                             |
| <b>show vpc</b>                                   | Displays information about vPC Fabric Peering mode.          |
| <b>show vpc virtual-peerlink vlan consistency</b> | Displays the VLANs which are not associated with vn-segment. |

### Example of the show vpc fabric-ports Command

```
switch# show vpc fabric-ports
Number of Fabric port : 9
Number of Fabric port active : 9

Fabric Ports State

Ethernet1/9 UP
Ethernet1/19/1 (port-channel151) UP
Ethernet1/19/2 (port-channel151) UP
Ethernet1/19/3 UP
Ethernet1/19/4 UP
Ethernet1/20/1 UP
Ethernet1/20/2 (port-channel152) UP
Ethernet1/20/3 (port-channel152) UP
Ethernet1/20/4 (port-channel152) UP
```

**Example of the show vpc Command**

```
switch# show vpc
```

Legend:

(\*) - local vPC is down, forwarding via vPC peer-link

```
vPC domain id : 3
Peer status : peer adjacency formed ok
vPC keep-alive status : peer is alive
Configuration consistency status : success
Per-vlan consistency status : success
Type-2 consistency status : success
vPC role : primary
Number of vPCs configured : 1
Peer Gateway : Enabled
Dual-active excluded VLANs : -
Graceful Consistency Check : Enabled
Auto-recovery status : Enabled, timer is off.(timeout = 240s)
Delay-restore status : Timer is off.(timeout = 30s)
Delay-restore SVI status : Timer is off.(timeout = 10s)
Operational Layer3 Peer-router : Disabled
Virtual-peerlink mode : Enabled
```

vPC Peer-link status

| id | Port   | Status | Active vlans                    |
|----|--------|--------|---------------------------------|
| 1  | Pol100 | up     | 1,56,98-600,1001-3401,3500-3525 |

vPC status

| Id  | Port   | Status | Consistency | Reason  | Active vlans    |
|-----|--------|--------|-------------|---------|-----------------|
| 101 | Pol101 | up     | success     | success | 98-99,1001-2800 |

Please check "show vpc consistency-parameters vpc <vpc-num>" for the consistency reason of down vpc and for type-2 consistency reasons for any vpc.

ToR\_B1#

**Example of the show vpc virtual-peerlink vlan consistency Command**

```
switch# show vpc virtual-peerlink vlan consistency
Following vlans are inconsistent
23
switch#
```





## APPENDIX **A**

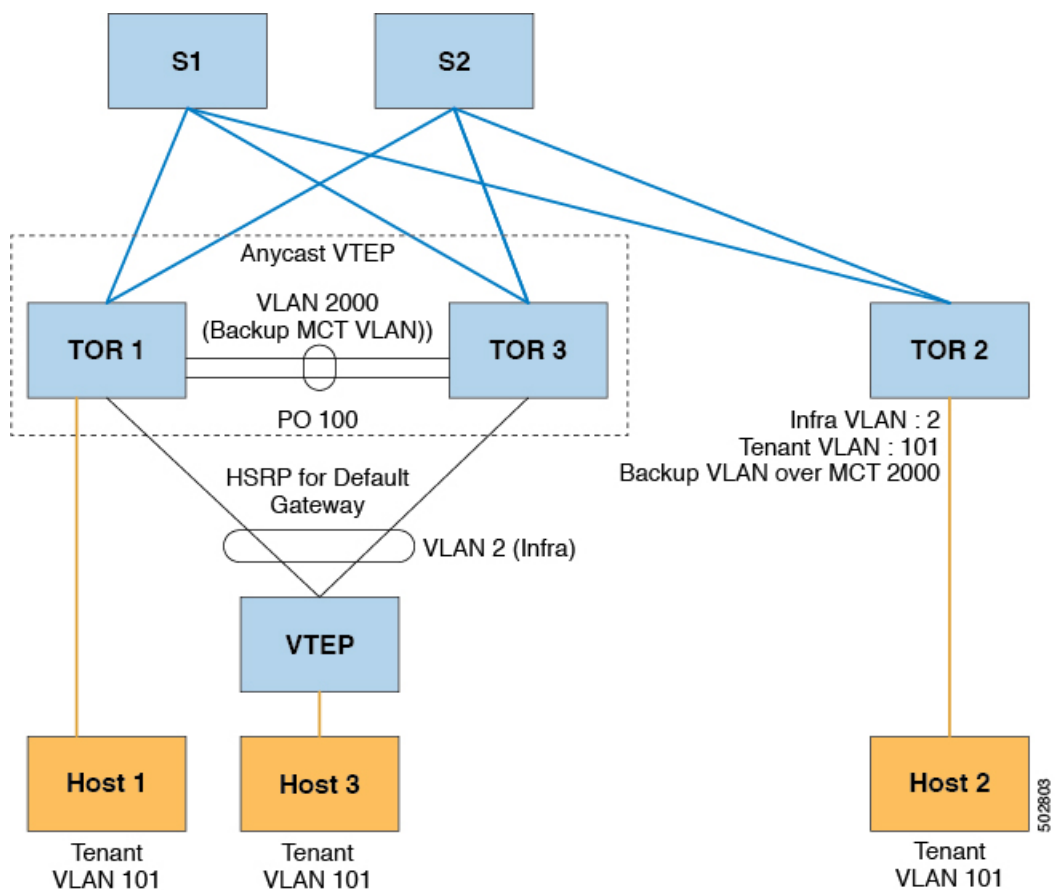
# Configuring Bud Node

This chapter contains the following sections:

- [VXLAN Bud Node Over vPC Overview, on page 267](#)
- [VXLAN Bud Node Over vPC Topology Example, on page 268](#)

## VXLAN Bud Node Over vPC Overview

*Figure 32: Underlay Network Based on PIM-SM and OSPF*





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

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



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

For Cisco Nexus 9200, 9300-EX, and 9300-FX/FX2 platform switches, use the **system nve infra-vlans** command to configure any VLANs that are used as infra-VLANs.

## VXLAN Bud Node Over vPC Topology Example

- Enable the required features:

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

- Configuration for PIM anycast RP.

In this example, 1.1.1.1 is the anycast RP address.

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

- VLAN configuration

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

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

- vPC configuration



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

- Infra VLAN SVI configuration

```
interface Vlan2
 no shutdown
 no ip redirects
 ip proxy-arp
 ip address 10.200.1.252/24
 no ipv6 redirects
 ip router ospf 1 area 0.0.0.0
 ip pim sparse-mode
 ip igmp static-oif route-map match-mcast-groups
 hsrp version 2
 hsrp 1
 ip 10.200.1.254
```

- Route-maps for matching multicast groups

Each VXLAN multicast group needs to have a static OIF on the backup SVI MCT.

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

- Backup SVI over MCT configuration

- Configuration Option 1:

```
interface Vlan2000
 no shutdown
 ip address 20.20.20.1/24
 ip router ospf 1 area 0.0.0.0
 ip pim sparse-mode
```

- Configuration Option 2:

```
interface Vlan2000
 no shutdown
 ip address 20.20.20.1/24
 ip router ospf 1 area 0.0.0.0
 ip pim sparse-mode
```

- vPC interface configuration that carries the infra VLAN

```
interface port-channel1
 switchport mode trunk
 switchport trunk allowed vlan 2
 vpc 1
```

- MCT configuration

```
interface port-channel100
 switchport mode trunk
 spanning-tree port type network
 vpc peer-link
```



---

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

---

#### NVE configuration

##### Option 1

```
interface nve1
 no shutdown
 source-interface loopback0
 member vni 10001 mcast-group 225.1.1.1
 member vni 10002 mcast-group 225.1.1.1
 member vni 10003 mcast-group 225.1.1.1
```

##### Option 2

```
interface nve1
 no shutdown
 source-interface loopback0
 global mcast-group 225.1.1.1
 member vni 10001
 member vni 10002
 member vni 10003
```

- Loopback interface configuration

```
interface loopback0
 ip address 101.101.101.101/32
 ip address 99.99.99.99/32 secondary
 ip router ospf 1 area 0.0.0.0
 ip pim sparse-mode
```

- Show commands

```

tor1# sh nve vni
Codes: CP - Control Plane DP - Data Plane
 UC - Unconfigured SA - Suppress ARP

Interface VNI Multicast-group State Mode Type [BD/VRF] Flags

nve1 10001 225.1.1.1 Up DP L2 [101]
nve1 10002 225.1.1.1 Up DP L2 [102]
nve1 10003 225.1.1.1 Up DP L2 [103]

tor1# sh nve peers
Interface Peer-IP State LearnType Uptime Router-Mac

nve1 10.200.1.1 Up DP 00:07:23 n/a
nve1 10.200.1.2 Up DP 00:07:18 n/a
nve1 102.102.102.102 Up DP 00:07:23 n/a

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

(*, 225.1.1.1/32), uptime: 00:07:41, ip pim nve static igmp
 Incoming interface: Ethernet2/1, RPF nbr: 10.1.5.2
 Outgoing interface list: (count: 3)
 Vlan2, uptime: 00:07:23, igmp
 Vlan2000, uptime: 00:07:31, static
 nve1, uptime: 00:07:41, nve

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

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

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

(102.102.102.102/32, 225.1.1.1/32), uptime: 00:07:40, ip mrib pim nve
 Incoming interface: Ethernet2/1, RPF nbr: 10.1.5.2
 Outgoing interface list: (count: 1)
 nve1, uptime: 00:07:40, nve

tor1# sh vpc
Legend:
 - local vPC is down, forwarding via vPC peer-link

vPC domain id : 1
Peer status : peer adjacency formed ok
vPC keep-alive status : peer is alive
Configuration consistency status : success
Per-vlan consistency status : success
Type-2 consistency status : success

```

```

vPC role : secondary, operational primary
Number of vPCs configured : 4
Peer Gateway : Enabled
Dual-active excluded VLANs : -
Graceful Consistency Check : Enabled
Auto-recovery status : Disabled
Delay-restore status : Timer is off.(timeout = 180s)
Delay-restore SVI status : Timer is off.(timeout = 10s)

```

#### vPC Peer-link status

```

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

```

#### vPC status

```

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

```

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

#### Legend:

Type 1 : vPC will be suspended in case of mismatch

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



## APPENDIX **B**

# DHCP Relay in VXLAN BGP EVPN

This chapter contains the following sections:

- [DHCP Relay in VXLAN BGP EVPN Overview, on page 273](#)
- [DHCP Relay in VXLAN BGP EVPN Example, on page 274](#)
- [DHCP Relay on VTEPs, on page 275](#)
- [Client on Tenant VRF and Server on Layer 3 Default VRF, on page 275](#)
- [Client on Tenant VRF \(SVI X\) and Server on the Same Tenant VRF \(SVI Y\), on page 278](#)
- [Client on Tenant VRF \(VRF X\) and Server on Different Tenant VRF \(VRF Y\), on page 282](#)
- [Client on Tenant VRF and Server on Non-Default Non-VXLAN VRF, on page 285](#)
- [Configuring vPC Peers Example, on page 287](#)
- [vPC VTEP DHCP Relay Configuration Example, on page 289](#)

## DHCP Relay in VXLAN BGP EVPN Overview

DHCP relay is utilized to forward DHCP packets between the hosts and DHCP server. The VXLAN VTEP can act as a relay agent, providing DHCP relay services in a multi-tenant VXLAN environment.

With DHCP Relay, DHCP messages require to be sent through the same Switch in both directions. GiAddr (Gateway IP Address) for DHCP Relay is commonly used for Scope Selection and DHCP response messages. In any VXLAN fabric with Distributed IP Anycast Gateway, DHCP messages can be returned to ANY Switch hosting the respective Gateway IP Address (GiAddr).

Solution requires a different way of Scope Selection and Unique IP Address for each Switch. Unique Loopback Interface per Switch will become GiAddr for responding to correct Switch. Option 82 (dhcp option vpn) will be used for Scope Selection based on L2VNI.

In a multi-tenant EVPN environment, DHCP relay uses the following sub-options of Option 82:

- Sub-option 151(0x97) - Virtual Subnet Selection (Defined in RFC#6607)

Used to convey VRF related information to the DHCP server in an MPLS-VPN and VXLAN EVPN multi-tenant environment.

- Sub-option 11(0xb) - Server ID Override (Defined in RFC#5107)

The server identifier (server ID) override sub-option allows the DHCP relay agent to specify a new value for the server ID option, which is inserted by the DHCP server in the reply packet. This sub-option allows the DHCP relay agent to act as the actual DHCP server such that the renew requests will come to the relay agent rather than the DHCP server directly. The server ID override sub-option contains the incoming

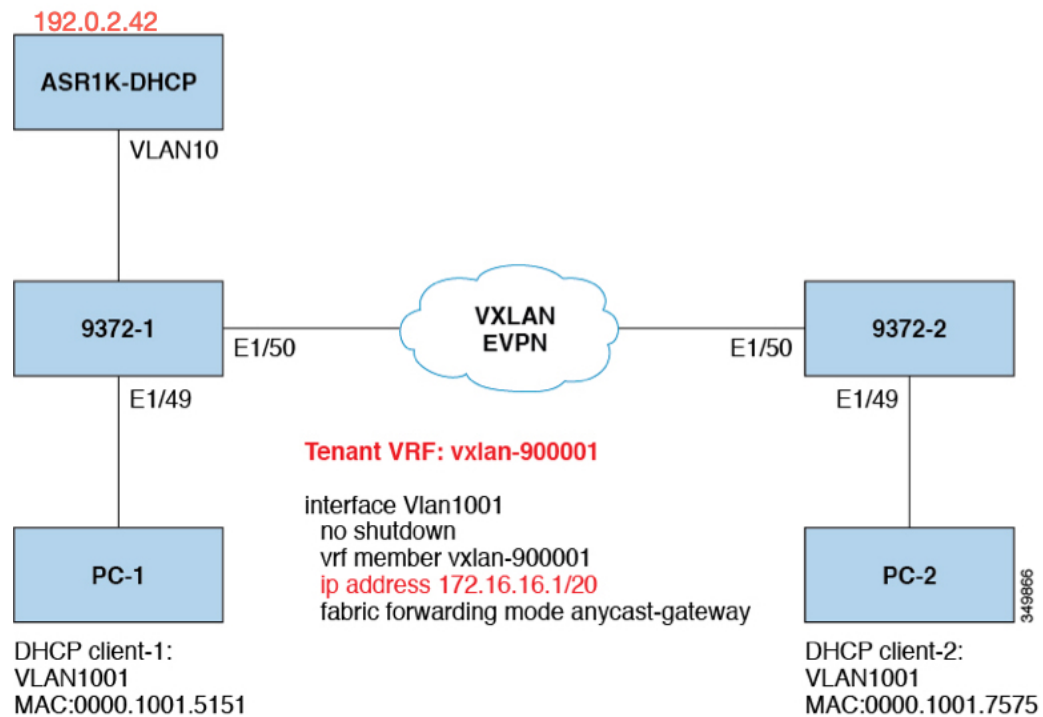
interface IP address, which is the IP address on the relay agent that is accessible from the client. Using this information, the DHCP client sends all renew and release request packets to the relay agent. The relay agent adds all of the appropriate sub-options and then forwards the renew and release request packets to the original DHCP server. For this function, Cisco's proprietary implementation is sub-option 152(0x98). You can use the **ip dhcp relay sub-option type cisco** command to manage the function.

- Sub-option 5(0x5) - Link Selection (Defined in RFC#3527)

The link selection sub-option provides a mechanism to separate the subnet/link on which the DHCP client resides from the gateway address (giaddr), which can be used to communicate with the relay agent by the DHCP server. The relay agent will set the sub-option to the correct subscriber subnet and the DHCP server will use that value to assign an IP address rather than the giaddr value. The relay agent will set the giaddr to its own IP address so that DHCP messages are able to be forwarded over the network. For this function, Cisco's proprietary implementation is sub-option 150(0x96). You can use the **ip dhcp relay sub-option type cisco** command to manage the function.

## DHCP Relay in VXLAN BGP EVPN Example

Figure 33: Example Topology



Topology characteristics:

- Switches 9372-1 and 9372-2 are VTEPs connected to the VXLAN fabric.
- Client1 and client2 are DHCP clients in vlan1001. They belong to tenant VRF vxlan-900001.
- The DHCP server is ASR1K, a router that sits in vlan10.
- DHCP server configuration

```
ip vrf vxlan900001
ip dhcp excluded-address vrf vxlan900001 172.16.16.1 172.16.16.9
ip dhcp pool one
vrf vxlan900001
network 172.16.16.0 255.240.0.0
defaultrouter 172.16.16.1
```

## DHCP Relay on VTEPs

The following are common deployment scenarios:

- Client on tenant VRF and server on Layer 3 default VRF.
- Client on tenant VRF (SVI X) and server on the same tenant VRF (SVI Y).
- Client on tenant VRF (VRF X) and server on different tenant VRF (VRF Y).
- Client on tenant VRF and server on non-default non-VXLAN VRF.

The following sections below move vlan10 to different VRFs to depict different scenarios.

## Client on Tenant VRF and Server on Layer 3 Default VRF

Put DHCP server (192.0.2.42) into the default VRF and make sure it is reachable from both 9372-1 and 9372-2 through the default VRF.

```
9372-1# sh run int vl 10

!Command: show running-config interface Vlan10
!Time: Mon Aug 24 07:51:16 2018

version 7.0(3)I1(3)

interface Vlan10
 no shutdown
 ip address 192.0.2.25/24
 ip router ospf 1 area 0.0.0.0

9372-1# ping 192.0.2.42 cou 1

PING 192.0.2.42 (192.0.2.42): 56 data bytes
64 bytes from 192.0.2.42: icmp_seq=0 ttl=254 time=0.593 ms
- 192.0.2.42 ping statistics -
1 packets transmitted, 1 packets received, 0.00% packet loss
roundtrip min/avg/max = 0.593/0.592/0.593 ms

9372-2# ping 192.0.2.42 cou 1
PING 192.0.2.42 (192.0.2.42): 56 data bytes
64 bytes from 192.0.2.42: icmp_seq=0 ttl=252 time=0.609 ms
- 192.0.2.42 ping statistics -
1 packets transmitted, 1 packets received, 0.00% packet loss
round-trip min/avg/max = 0.609/0.608/0.609 ms
```

## DHCP Relay Configuration

### • 9372-1

```
9372-1# sh run dhcp

!Command: show running-config dhcp
!Time: Mon Aug 24 08:26:00 2018

version 7.0(3) I1(3)
feature dhcp

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

interface Vlan1001
 ip dhcp relay address 192.0.2.42 use-vrf default
```

### • 9372-2

```
9372-2# sh run dhcp

!Command: show running-config dhcp
!Time: Mon Aug 24 08:26:16 2018

version 7.0(3)11(3)
feature dhcp

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

interfaoe Vlan1001
 ip dhcp relay address 192.0.2.42 use-vrf default
```

## Debug Output

- The following is a packet dump for DHCP interact sequences.

```
9372-1# ethanalyzer local interface inband display-filter
"udp.srcport==67 or udp.dstport==67" limit-captured frames 0

Capturing on inband
20180824 08:35:25.066530 0.0.0.0 -> 255.255.255.0 DHCP DHCP Discover - Transaction ID
0x636a38fd
20180824 08:35:25.068141 192.0.2.25 -> 192.0.2.42 DHCP DHCP Discover - Transaction ID
0x636a38fd
20180824 08:35:27.069494 192.0.2.42 -> 192.0.2.25 DHCP DHCP Offer Transaction - ID
0x636a38fd
20180824 08:35:27.071029 172.16.16.1 -> 172.16.16.11 DHCP DHCP Offer Transaction - ID
0x636a38fd
20180824 08:35:27.071488 0.0.0.0 -> 255.255.255.0 DHCP DHCP Request Transaction - ID
```



```

0x636a38fd
20180824 08:35:27.072447 192.0.2.25 -> 192.0.2.42 DHCP DHCP Request Transaction - ID
0x636a38fd
20180824 08:35:27.073008 192.0.2.42 -> 192.0.2.25 DHCP DHCP ACK Transaction - ID
0x636a38fd
20180824 08:35:27.073692 172.16.16.1 -> 172.16.16.11 DHCP DHCP ACK Transaction - ID
0x636a38fd

```



**Note** 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 192.0.2.25 (ip address of vlan10) and suboptions 5/11/151 are set accordingly.

```

Bootp flags: 0x0000 (unicast)
client IP address: 0.0.0.0 (0.0.0.0)
Your (client) IP address: 0.0.0.0 (0.0.0.0)
Next server IP address: 0.0.0.0 (0.0.0.0)
Relay agent IP address: 192.0.2.25 (192.0.2.25)
client MAC address Hughes_01:51:51 (00:00:10:01:51:51)
client hardware address padding: 00000000000000000000
Server host name not given
Boot file name not given
Magic cookie: DHCP
Option: (53) DHCP Message Type
 Length: 1
 DHCP: Discover (1)
Option: (55) Parameter Request List
 Length: 4
 Parameter Request List Item: (1) Subnet Mask
 Parameter Request List Item: (3) Router
 Parameter Request List Item: (58) Renewal Time Value
 Parameter Request List Item: (59) Rebinding Time Value
Option: (61) client identifier
 Length: 7
 Hardware type: Ethernet (0x01)
 Client MAC address: Hughes_01:51:51 (00:00:10:01:51:51)
Option: (82) Agent Information Option
 Length: 47
Option 82 Suboption: (1) Agent Circuit ID
 Length: 10
 Agent Circuit ID: 01080006001e88690030
Option 82 Suboption: (2) Agent Remote ID
 Length: 6
 Agent Remote ID: f8c2882333a5
Option 82 Suboption: (151) VRF name/VPN ID
Option 82 Suboption: (11) Server ID Override
 Length: 4
 Server ID Override: 172.16.16.1 (172.16.16.1)
Option 82 Suboption: (5) Link selection
 Length: 4
 Link selection: 172.16.16.0 (172.16.16.0)

```

```
ASR1K-DHCP# sh ip dhcp bin
```

```

Bindings from all pools not associated with VRF:
IP address ClientID/ Lease expiration Type State Interface
 Hardware address/
 User name

Bindings from VRF pool vxlan900001:
IP address ClientID/ Lease expiration Type State Interface
 Hardware address/
 User name
172.16.16.10 0100.0010.0175.75 Aug 25 2018 09:21 AM Automatic Active GigabitEthernet2/1/0
172.16.16.11 0100.0010.0151.51 Aug 25 2018 08:54 AM Automatic Active GigabitEthernet2/1/0

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

10.11.11.11/8, ubest/mbest: 2/0, attached
 *via 10.11.11.11, Lo1, [0/0], 18:31:57, local
 *via 10.11.11.11, Lo1, [0/0], 18:31:57, direct
10.22.22.22/8, ubest/mbest: 1/0
 *via 1.2.2.2%default, [200/0], 18:31:57, bgp65535,internal, tag 65535 (evpn)segid:
900001 tunnelid: 0x2020202
encap: VXLAN

172.16.16.0/20, ubest/mbest: 1/0, attached
 *via 172.16.16.1, Vlan1001, [0/0], 18:31:57, direct
172.16.16.1/32, ubest/mbest: 1/0, attached
 *via 172.16.16.1, Vlan1001, [0/0], 18:31:57, local
172.16.16.10/32, ubest/mbest: 1/0
 *via 1.2.2.2%default, [200/0], 00:00:47, bgp65535,internal, tag 65535 (evpn)segid:
900001 tunnelid: 0x2020202
encap: VXLAN

172.16.16.11/32, ubest/mbest: 1/0, attached
 *via 172.16.16.11, Vlan1001, [190/0], 00:28:10, hmm

9372-1# ping 172.16.16.11 vrf vxlan900001 count 1
PING 172.16.16.11 (172.16.16.11): 56 data bytes
64 bytes from 172.16.16.11: icmp_seq=0 ttl=63 time=0.846 ms
- 172.16.16.11 ping statistics -
1 packets transmitted, 1 packets received, 0.00% packet loss
round-trip min/avg/max = 0.846/0.845/0.846 ms

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

```

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

Put DHCP server (192.0.2.42) into VRF of vxlan-900001 and make sure it is reachable from both 9372-1 and 9372-2 through VRF of vxlan-900001.

```

9372-1# sh run int vl 10

!Command: show running-config interface Vlan10
!Time: Mon Aug 24 09:10:26 2018

version 7.0(3)I1(3)

interface Vlan10
 no shutdown
 vrf member vxlan-900001
 ip address 192.0.2.25/24

```

Because 172.16.16.1 is an anycast address for vlan1001 configured on all the VTEPs, we need to pick up a unique address as the DHCP relay packet's source address to make sure the DHCP server can deliver a response to the original DHCP Relay agent. In this scenario, we use loopback1 and we need to make sure loopback1 is reachable from everywhere of VRF vxlan-900001.

```

9372-1# sh run int lo1

!Command: show running-config interface loopback1
!Time: Mon Aug 24 09:18:53 2018

version 7.0(3)I1(3)

interface loopback1
 vrf member vxlan-900001
 ip address 10.11.11.11/8

9372-1# ping 192.0.2.42 vrf vxlan900001 source 10.11.11.11 cou 1
PING 192.0.2.42 (192.0.2.42) from 10.11.11.11: 56 data bytes
64 bytes from 192.0.2.42: icmp_seq=0 ttl=254 time=0.575 ms
- 192.0.2.42 ping statistics -
1 packets transmitted, 1 packets received, 0.00% packet loss
round-trip min/avg/max = 0.575/0.574/0.575 ms

9372-2# sh run int lo1

!Command: show running-config interface loopback1
!Time: Mon Aug 24 09:19:30 2018

version 7.0(3)I1(3)

interface loopback1
 vrf member vxlan900001
 ip address 10.22.22.22/8

9372-2# ping 192.0.2.42 vrf vxlan-900001 source 10.22.22.22 cou 1
PING 192.0.2.42 (192.0.2.42) from 10.22.22.22: 56 data bytes
64 bytes from 192.0.2.42: icmp_seq=0 ttl=253 time=0.662 ms
- 192.0.2.42 ping statistics -
1 packets transmitted, 1 packets received, 0.00% packet loss
round-trip min/avg/max = 0.662/0.662/0.662 ms

```

## DHCP Relay Configuration

- 9372-1

```

9372-1# sh run dhcp

!Command: show running-config dhcp
!Time: Mon Aug 24 08:26:00 2018

version 7.0(3)11(3)
feature dhcp

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

interface Vlan1001
 ip dhcp relay address 192.0.2.42
 ip dhcp relay source-interface loopback1

```

#### • 9372-2

```

9372-2# sh run dhcp

!Command: show running-config dhcp
!Time: Mon Aug 24 08:26:16 2018

version 7.0(3) 11(3)
feature dhcp

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

interface Vlan1001
 ip dhcp relay address 192.0.2.42
 ip dhcp relay source-interface loopback1

```

### Debug Output

- The following is a packet dump for DHCP interact sequences.

```

9372-1# ethanalyzer local interface inband display-filter
"udp.srcport==67 or udp.dstport==67" limit-captured frames 0

Capturing on inband
20180824 09:31:38.129393 0.0.0.0 -> 255.255.255.0 DHCP DHCP Discover - Transaction ID
0x860cd13
20180824 09:31:38.129952 10.11.11.11 -> 192.0.2.42 DHCP DHCP Discover - Transaction ID
0x860cd13
20180824 09:31:40.130134 192.0.2.42 -> 10.11.11.11 DHCP DHCP Offer - Transaction ID
0x860cd13
20180824 09:31:40.130552 172.16.16.1 -> 172.16.16.11 DHCP DHCP Offer - Transaction ID
0x860cd13
20180824 09:31:40.130990 0.0.0.0 -> 255.255.255.0 DHCP DHCP Request - Transaction ID
0x860cd13
20180824 09:31:40.131457 10.11.11.11 -> 192.0.2.42 DHCP DHCP Request - Transaction ID
0x860cd13
20180824 09:31:40.132009 192.0.2.42 -> 10.11.11.11 DHCP DHCP ACK - Transaction ID

```

```
0x860cd13
20180824 09:31:40.132268 172.16.16.1 -> 172.16.16.11 DHCP DHCP ACK - TransactionID
0x860cd13
```



**Note** 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 10.11.11.11(loopback1) and suboptions 5/11/151 are set accordingly.

```
Bootstrap Protocol
Message type: Boot Request (1)
Hardware type: Ethernet (0x01)
Hardware address length: 6
Hops: 1
Transaction ID: 0x0860cd13
Seconds elapsed: 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: 10.11.11.11 (10.11.11.11)
Client MAC address: Hughes_01:51:51 (00:00:10:01:51:51)
Client hardware address padding: 00000000000000000000
Server host name not given
Boot file name not given
Magic cookie: DHCP
Option: (53) DHCP Message Type
Length: 1
DHCP: Discover (1)
Option: (55) Parameter Request List
Option: (61) Client Identifier
Option: (82) Agent Information Option
Length: 47
Option 82 suboption: (1) 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 2018 10:02 AM Automatic Active GigabitEthernet2/1/0
172.16.16.11 0100.0010.0151.51 Aug 25 2018 09:50 AM Automatic Active GigabitEthernet2/1/0

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

10.11.11.11/8, ubest/mbest: 2/0, attached
 *via 10.11.11.11, Lo1, [0/0], 19:13:56, local
 *via 10.11.11.11, Lo1, [0/0], 19:13:56, direct
10.22.22.22/8, ubest/mbest: 1/0
 *via 2.2.2.2%default, [200/0], 19:13:56, bgp65535,internal, tag 65535 (evpn)segid:
900001 tunnelid: 0x2020202
encap: VXLAN
172.16.16.0/20, ubest/mbest: 1/0, attached
 *via 172.16.16.1, Vlan1001, [0/0], 19:13:56, direct
172.16.16.1/32, ubest/mbest: 1/0, attached
 *via 172.16.16.1, Vlan1001, [0/0], 19:13:56, local
172.16.16.10/32, ubest/mbest: 1/0
 *via 2.2.2.2%default, [200/0], 00:01:27, bgp65535,
internal, tag 65535 (evpn)segid: 900001 tunnelid: 0x2020202
encap: VXLAN
172.16.16.11/32, ubest/mbest: 1/0, attached
 *via 172.16.16.11, Vlan1001, [190/0], 00:13:56, hmm
192.0.2.20/24, ubest/mbest: 1/0, attached
 *via 192.0.2.25, Vlan10, [0/0], 00:36:08, direct
192.0.2.25/24, ubest/mbest: 1/0, attached
 *via 192.0.2.25, Vlan10, [0/0], 00:36:08, local
9372-1# ping 172.16.16.10 vrf vxlan-900001 cou 1
PING 172.16.16.10 (172.16.16.10): 56 data bytes
64 bytes from 172.16.16.10: icmp_seq=0 ttl=62 time=0.808 ms
- 172.16.16.10 ping statistics -
1 packets transmitted, 1 packets received, 0.00% packet loss
round-trip min/avg/max = 0.808/0.808/0.808 ms

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

```

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

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

```

9372-1# sh run int vl 10
!Command: show runningconfig interface Vlan10
!Time: Tue Aug 25 08:48:22 2018

```

```

version 7.0(3)I1(3)
interface Vlan10
 no shutdown
 vrf member vxlan900002
 ip address 192.0.2.40/24

9372-1# sh run int lo2
!Command: show runningconfig interface loopback2
!Time: Tue Aug 25 08:48:57 2018
version 7.0(3)I1(3)
interface loopback2
 vrf member vxlan900002
 ip address 10.33.33.33/8

9372-2# sh run int lo2
!Command: show runningconfig interface loopback2
!Time: Tue Aug 25 08:48:44 2018
version 7.0(3)I1(3)
interface loopback2
 vrf member vxlan900002
 ip address 10.44.44.44/8

9372-1# ping 192.0.2.42 vrf vxlan-900002 source 10.33.33.33 cou 1
PING 192.0.2.42 (192.0.2.42) from 10.33.33.33: 56 data bytes
64 bytes from 192.0.2.42: icmp_seq=0 ttl=254 time=0.544 ms
- 192.0.2.42 ping statistics -
1 packets transmitted, 1 packets received, 0.00% packet loss
round-trip min/avg/max = 0.544/0.544/0.544 ms

9372-2# ping 192.0.2.42 vrf vxlan-900002 source 10.44.44.44 count 1
PING 192.0.2.42 (192.0.2.42) from 10.44.44.44: 56 data bytes
64 bytes from 192.0.2.42: icmp_seq=0 ttl=253 time=0.678 ms
- 192.0.2.42 ping statistics -
1 packets transmitted, 1 packets received, 0.00% packet loss
round-trip min/avg/max = 0.678/0.678/0.678 ms

```

## DHCP Relay Configuration

### • 9372-1

```

9372-1# sh run dhcp

!Command: show running-config dhcp
!Time: Mon Aug 24 08:26:00 2018

version 7.0(3) Ii (3)
feature dhcp

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

interface Vlan1001
 ip dhcp relay address 192.0.2.42 use-vrf vxlan-900002
 ip dhcp relay source-interface loopback2

```

### • 9372-2

```

!Command: show running-config dhcp
!Time: Mon Aug 24 08:26:16 2018

version 7.0(3)11(3)
feature dhcp

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

interface Vlan1001
 ip dhcp relay address 192.0.2.42 use-vrf vxlan-900002
 ip dhcp relay source-interface loopback2

```

### Debug Output

- The following is a packet dump for DHCP interact sequences.

```

9372-1# ethanalyzer local interface inband display-filter "udp.srcport==67 or
udp.dstport==67" limit-captured-frames 0
Capturing on inband
20180825 08:59:35.758314 0.0.0.0 -> 255.255.255.0 DHCP DHCP Discover - Transaction ID
0x3eebccae
20180825 08:59:35.758878 10.33.33.33 -> 192.0.2.42 DHCP DHCP Discover - Transaction ID
0x3eebccae
20180825 08:59:37.759560 192.0.2.42 -> 10.33.33.33 DHCP DHCP Offer - Transaction ID
0x3eebccae
20180825 08:59:37.759905 172.16.16.1 -> 172.16.16.11 DHCP DHCP Offer - Transaction ID
0x3eebccae
20180825 08:59:37.760313 0.0.0.0 -> 255.255.255.0 DHCP DHCP Request - Transaction ID
0x3eebccae
20180825 08:59:37.760733 10.33.33.33 -> 192.0.2.42 DHCP DHCP Request - Transaction ID
0x3eebccae
20180825 08:59:37.761297 192.0.2.42 -> 10.33.33.33 DHCP DHCP ACK - Transaction ID
0x3eebccae
20180825 08:59:37.761554 172.16.16.1 -> 172.16.16.11 DHCP DHCP ACK - Transaction ID
0x3eebccae

```

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

giaddr is set to 10.33.33.33 (loopback2) and suboptions 5/11/151 are set accordingly.

```

Bootstrap Protocol
Message type: Boot Request (1)
Hardware type: Ethernet (0x01)
Hardware address length: 6
Hops: 1
Transaction ID: 0x3eebccae
Seconds elapsed: 0
Bootp flags: 0x0000 (unicast)
Client IP address: 0.0.0.0 (0.0.0.0)
Your (client) IP address: 0.0.0.0 (0.0.0.0)
Next server IP address: 0.0.0.0 (0.0.0.0)
Relay agent IP address: 10.33.33.33 (10.33.33.33)
Client MAC address: i-ughes_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 M0 interface. The IP address changes to 10.122.164.147 accordingly.

```

9372-1# sh run int m0
!Command: show running-config interface mgmt0
!Time: Tue Aug 25 09:17:04 2018
version 7.0(3)I1(3)
interface mgmt0
 vrf member management
 ip address 10.122.165.134/8

9372-1# ping 10.122.164.147 vrf management cou 1
PING 10.122.164.147 (10.122.164.147): 56 data bytes
64 bytes from 10.122.164.147: icmp_seq=0 ttl=251 time=1.024 ms
- 10.122.164.147 ping statistics -
1 packets transmitted, 1 packets received, 0.00% packet loss
round-trip min/avg/max = 1.024/1.024/1.024 ms

9372-2# sh run int m0
!Command: show running-config interface mgmt0
!Time: Tue Aug 25 09:17:47 2018
version 7.0(3)I1(3)
interface mgmt0
 vrf member management
 ip address 10.122.165.148/8

9372-2# ping 10.122.164.147 vrf management cou 1
PING 10.122.164.147 (10.122.164.147): 56 data bytes
64 bytes from 10.122.164.147: icmp_seq=0 ttl=251 time=1.03 ms
- 10.122.164.147 ping statistics -
1 packets transmitted, 1 packets received, 0.00% packet loss

```

```
round-trip min/avg/max = 1.03/1.03/1.03 ms
```

## DHCP Relay Configuration

### • 9372-1

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

!Command: show running-config dhcp
!Time: Mon Aug 24 08:26:00 2018

version 7.0(3)11(3)
feature dhcp

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

interface 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 2018

version 7.0(3)11(3)
feature dhcp

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

interface 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
20180825 09:30:54.214998 0.0.0.0 -> 255.255.255.0 DHCP DHCP Discover - Transaction ID
0x28a8606d
20180825 09:30:56.216491 172.16.16.1 -> 172.16.16.11 DHCP DHCP Offer - Transaction ID
0x28a8606d
20180825 09:30:56.216931 0.0.0.0 -> 255.255.255.0 DHCP DHCP Request - Transaction ID
0x28a8606d
20180825 09:30:56.218426 172.16.16.1 -> 172.16.16.11 DHCP DHCP ACK - Transaction ID
0x28a8606d
```

```

9372-1# ethanalyzer local interface mgmt display-filter "ip.src==10.122.164.147 or
ip.dst==10.122.164.147" limit-captured-frames 0
Capturing on mgmt0
20180825 09:30:54.215499 10.122.165.134 -> 10.122.164.147 DHCP DHCP Discover - Transaction
ID 0x28a8606d
20180825 09:30:56.216137 10.122.164.147 -> 10.122.165.134 DHCP DHCP Offer - Transaction
ID 0x28a8606d
20180825 09:30:56.217444 10.122.165.134 -> 10.122.164.147 DHCP DHCP Request - Transaction
ID 0x28a8606d
20180825 09:30:56.218207 10.122.164.147 -> 10.122.165.134 DHCP DHCP ACK - Transaction
ID 0x28a8606d

```

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

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

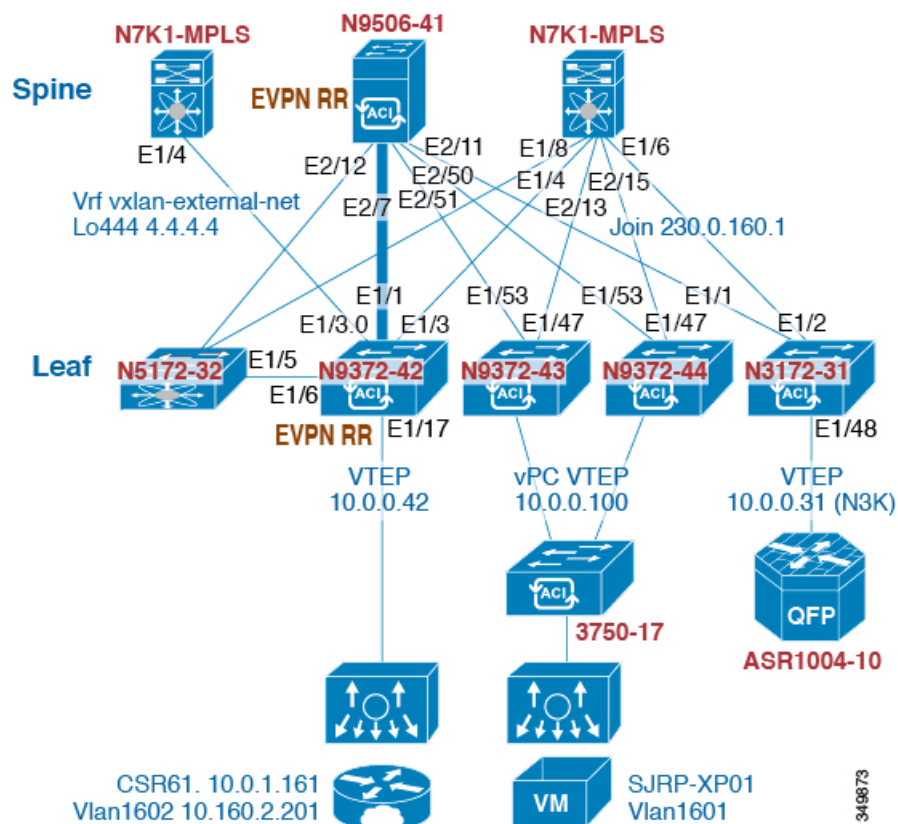
```

Bootstrap Protocol
Message type: Boot Request (1)
Hardware type: Ethernet (0x01)
Hardware address length: 6
Hops: 1
Transaction ID: 0x28a8606d
Seconds elapsed: 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 192.0.2.36/24
 ip router ospf 1 area 0 /* Only required for vPC VTEP. */
```

- Advertise LoX into the Layer 3 VRF BGP.

```
Router bgp 2
 vrf X
 network 10.1.1.42/8
```

- Configure DHCP relay on the SVI under the VRF.

```
interface Vlan1601
 vrf member evpn-tenant-kk1
 ip address 10.160.1.254/8
 fabric forwarding mode anycast-gateway
 ip dhcp relay address 10.160.2.201
 ip dhcp relay source-interface loopback601
```

- Configure Layer 3 VNI SVI with **ip forward**.

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

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




---

**Note** Only required for vPC VTEP

---

```
Vlan 1605
interface Vlan1605
 vrf member evpn-tenant-kk1
 ip address 10.160.5.43/8
 ip router ospf 1 area 10.10.10.41
```

- Create the VRF routing.




---

**Note** Only required for vPC VTEP.

---

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

## vPC VTEP DHCP Relay Configuration Example

To address a need to configure a VLAN that is allowed across the MCT/peer-link, such as a vPC VLAN, an SVI can be associated to the VLAN and is created within the tenant VRF. This becomes an underlay peering, with the underlay protocol, such as OSPF, that needs the tenant VRF instantiated under the routing process.

Alternatively, instead of placing the SVI within the routing protocol and instantiate the Tenant-VRF under the routing process, you can use the static routes between the vPC peers across the MCT. This approach ensures that the reply from the server returns to the correct place and each VTEP uses a different loopback interface for the GiAddr.

The following are examples of these configurations:

- Configuration of SVI within underlay routing:

```
/* vPC Peer-1 */

router ospf UNDERLAY
vrf tenant-vrf

interface Vlan2000
 no shutdown
 mtu 9216
 vrf member tenant-vrf
 ip address 192.168.1.1/16
 ip router ospf UNDERLAY area 0.0.0.0

/* vPC Peer-2 */

router ospf UNDERLAY
vrf tenant-vrf

interface Vlan2000
 no shutdown
 mtu 9216
 vrf member tenant-vrf
 ip address 192.168.1.2/16
 ip router ospf UNDERLAY area 0.0.0.0
```

- Configuration of SVI using static routes between vPC peers across the MCT:

```
/* vPC Peer-1 */

interface Vlan2000
 no shutdown
 mtu 9216
 vrf member tenant-vrf
 ip address 192.168.1.1/16

vrf context tenant-vrf
ip route 192.168.1.2/16 192.168.1.1

/* vPC Peer-2 */

interface Vlan2000
 no shutdown
 mtu 9216
 vrf member tenant-vrf
 ip address 192.168.1.2/16

vrf context tenant-vrf
ip route 192.168.1.1/16 192.168.1.2
```



## APPENDIX C

# Configuring Layer 4 - Layer 7 Network Services Integration

---

This chapter contains the following sections:

- [About VXLAN Layer 4 - Layer 7 Services, on page 291](#)
- [Integrating Layer 3 Firewalls in VXLAN Fabrics, on page 291](#)
- [Firewall as Default Gateway, on page 305](#)
- [Transparent Firewall Insertion, on page 306](#)

## About VXLAN Layer 4 - Layer 7 Services

This chapter covers insertion of Layer 4 – Layer 7 network services (firewall, load balancer, and so on) in a VXLAN fabric.

As opposed to traditional 3-tier network topologies, in which L4-L7 services are connected to the switches hosting the default gateway (aggregation/distribution), L4-L7 services in VXLAN fabrics are typically connected to the leaf or border switches, often referred to as *services leafs*.

You can attach a L4-L7 services device to a VXLAN fabric in various ways. This chapter addresses the considerations you must take depending on how the L4-L7 services device is attached and the requirements of the device and the network.

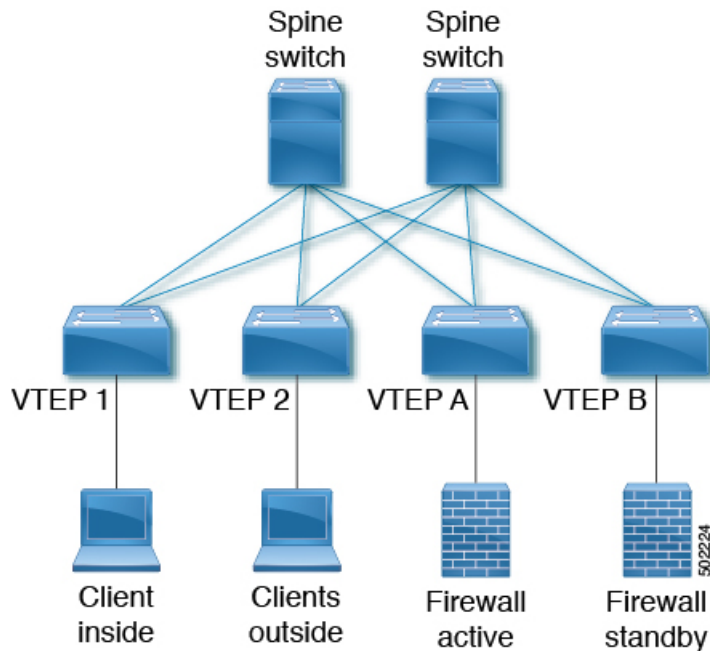
## Integrating Layer 3 Firewalls in VXLAN Fabrics

This section provides details on how to integrate a firewall within a VXLAN EVPN fabric. A Layer-3 firewall involves separating different security zones.

When integrating a Layer-3 firewall in a VXLAN EVPN fabric with a distributed Anycast Gateway, each of these zones must correspond to a VRF/tenant on the fabric. The traffic within a tenant is routed by the fabric. Traffic between the tenants is routed by the firewall. This scenario often refers to an inter-tenant or tenant edge firewall.

Consider two zones: an inside zone and an outside zone. This scenario requires a VRF definition on the fabric. You can call the VRFs the inside VRF and the outside VRF. Traffic between subnets within the same VRF is routed on the VXLAN fabric using the distributed gateway. Traffic between VRFs is routed by the firewall where the rules are applied.

Figure 34: Topology Overview with Firewall Attachment



## Single-Attached Firewall with Static Routing

If the firewall does not support running a routing protocol, you must have static routes on each VTEP pointing to the firewall as the next hop. The firewall also has static routes pointing to the Anycast Gateway IP as the next hop. The challenge with a static route is that the VTEP with an active firewall must be the one advertising the routes to the fabric. One way to accomplish this is to track the active firewall reachability via HMM and use this tracking to advertise routes into the fabric. When the active firewall is connected to VTEP A, VTEP A has a static route that tracks where the route is advertised if the firewall IP is learned as the HMM route. When the firewall fails and the standby firewall takes over, VTEP A learns the firewall IP using BGP, and VTEP B learns the firewall IP using HMM. VTEP A withdraws the route, and VTEP B advertises the route into the fabric. See the following example.

### VTEP A and VTEP B:

```
Vlan 10
 Name inside
 Vn-segment 10010

Vlan 20
 Name outside
 Vn-segment 10020

Interface VLAN 10
 Description inside_vlan
 VRF member INSIDE
 IP address 10.1.1.254/24
 fabric forwarding mode anycast-gateway

Interface VLAN 20
 Description outside_vlan
 VRF member OUTSIDE
```



```

IP address 20.1.1.254/24
fabric forwarding mode anycast-gateway

interface nve1
 no shutdown
 host-reachability protocol bgp
 source-interface loopback1
 member vni 10010
 mcastgroup 239.1.1.1
 member vni 10020
 mcastgroup 239.1.1.1
 member vni 1001000 associate-vrf
 member vni 1002000 associate-vrf

track 10 ip route 10.1.1.1/32 reachability hmm
 vrf member INSIDE
!
VRF context INSIDE
 Vni 1001000
 IP route 20.1.1.0/24 10.1.1.1 track 10

track 20 ip route 20.1.1.1/32 reachability hmm
 vrf member OUTSIDE
!
VRF context OUTSIDE
 Vni 1001000
 IP route 10.1.1.0/24 20.1.1.1 track 20

VTEPA# show track 10 Track 10
IP Route 20.1.1.1/32 Reachability Reachability is UP

VTEPA# show ip route 20.1.1.0/24 vrf INSIDE
IP Route Table for VRF "INSIDE"
'*' denotes best ucast next-hop
***' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

20.1.1.0/24, ubest/mbest: 1/0
 *via 10.1.1.1 [1/0], 00:00:08, static

Firewall Failure on VTEP A caused the track to go down causing VTEP A to withdraw the static
route.

VTEPA# show track 20 Track 20
IP Route 20.1.1.1/32 Reachability Reachability is DOWN

VTEPA# show ip route 20.1.1.0/24 vrf INSIDE
IP Route Table for VRF "RED"
'*' denotes best ucast next-hop
***' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

Route not found

```

## Recursive Static Routes Distributed to the Rest of the Fabric

With this approach, the static routes are configured wherever the inside or outside VRF exists. As the next-hop is reachable through host routes (EVPN Route-Type2), the change of the active firewall to standby and vice versa is only seen locally and doesn't introduce any churn to the other VXLAN fabric. This approach can help to better scale and improve convergence.

### Any VTEP:

```
VRF context OUTSIDE
Vni 1002000
IP route 10.1.1.0/24 20.1.1.1
! static route on VTEP pointing to Firewall next hop
! firewall VIP 20.1.1.1

VRF context INSIDE
Vni 1001000
IP route 20.1.1.0/24 10.1.1.1
! static route on VTEP pointing to Firewall next hop
! firewall VIP 10.1.1.1
```

## Redistribute Static Routes into BGP and Advertise to the Rest of the Fabric

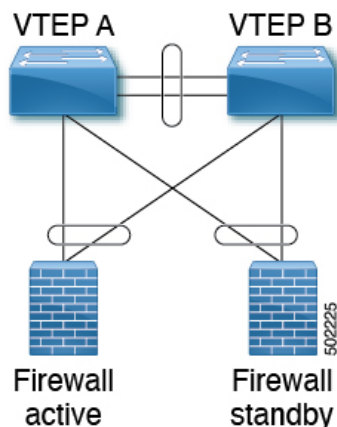
Through redistribution, we make the route toward the active firewall shown to the VTEP where it resides. The route is seen as a prefix route (EVPN Route-Type5), and as such, only the route toward the VTEP with the active firewall is seen. In the case of a firewall active/standby change, the tracking needs to detect the change and inform all of the remote VTEPs of this change. This behavior is equal to a route "delete" followed by an "add." This approach needs to notify all VTEPs with the VRF, and hence a wider churn can be seen.

### VTEP A and VTEP B:

```
router bgp 65000
vrf OUTSIDE
address-family ipv4 unicast
redistribute static route-map Static-to-BGP
```

## Dual-Attached Firewall with Static Routing

Figure 35: Dual-Attached Firewall with Static Routing



**VTEP A and VTEP B:**

```

Vlan 10
 Name inside
 Vn-segment 10010

Vlan 20
 Name outside
 Vn-segment 10020

interface nve1
 no shutdown
 host-reachability protocol bgp
 source-interface loopback1
 member vni 10010
 mcastgroup 239.1.1.1
 member vni 10020
 mcastgroup 239.1.1.1
 member vni 1001000 associate-vrf
 member vni 1002000 associate-vrf

Interface VLAN 10
 Description inside_vlan
 VRF member INSIDE
 IP address 10.1.1.254/24
 fabric forwarding mode anycast-gateway

Interface VLAN 20
 Description outside_vlan
 VRF member OUTSIDE
 IP address 20.1.1.254/24
 fabric forwarding mode anycast-gateway

VRF context INSIDE
 Vni 1001000
 IP route 20.1.1.0/24 10.1.1.1
 ! static route on VTEP pointing to Firewall next hop
 ! firewall VIP 10.1.1.1
VRF context OUTSIDE
 Vni 1002000
 IP route 10.1.1.0/24 20.1.1.1
 ! static route on VTEP pointing to Firewall next hop
 ! firewall VIP 20.1.1.1

router bgp 65000
 vrf INSIDE
 address-family ipv4 unicast
 redistribute static route-map INSIDE-to-BGP
 vrf OUTSIDE
 address-family ipv4 unicast
 redistribute static route-map OUTSIDE-to-BGP

```

## Single-Attached Firewall with eBGP Routing

If the firewall supports BGP, one option is to use BGP as a protocol between the firewall and the service VTEP. Peering using the anycast IP is not supported. The recommended design is to use dedicated loopback IPs on each VTEP and peer using the loopback. As long as the loopback interfaces are not advertised via

EVPN, the same IP address could be used on all of the belonging VTEPs. We recommend using individual IP addresses on a per-VTEP basis.

Reachability to the loopback from the firewall can be configured using a static route on the firewall, pointing to the Anycast Gateway IP on the VTEPs.

In the following example, an eBGP peering is established from the VTEPs, which are in AS 65000, and the firewall in AS 65002. The BGP peering with iBGP is not supported.



**Note** When having eBGP peering to active/standby firewalls connected to different VTEPs, **export-gateway-ip** must be enabled.

Do not use Anycast Gateway for BGP peerings.

#### VTEP A:

```
Vlan 10
 Name inside
 Vn-segment 10010

Vlan 20
 Name outside
 Vn-segment 10020

Interface VLAN 10
 Description inside_vlan
 VRF member INSIDE
 IP address 10.1.1.254/24
 fabric forwarding mode anycast-gateway

Interface loopback100
 Vrf member INSIDE
 Ip address 172.16.1.253/32

Interface VLAN 20
 Description outside_vlan
 VRF member OUTSIDE
 IP address 20.1.1.254/24
 fabric forwarding mode anycast-gateway

Interface loopback101
 Vrf member OUTSIDE
 Ip address 172.18.1.253/32

router bgp 65000
 vrf INSIDE
 ! peer with Firewall Inside
 neighbor 10.1.1.0/24 remote-as 65123
 update-source loopback100
 ebgp-multihop 5
 address-family ipv4 unicast
 local-as 65051 no-prepend replace-as

 vrf OUTSIDE
 ! peer with Firewall Outside
 neighbor 20.1.1.0/24 remote-as 65123
 update-source loopback101
 ebgp-multihop 5
```

```
address-family ipv4 unicast
 local-as 65052 no-prepend replace-as
```

### VTEP B:

```
Vlan 10
 Name inside
 Vn-segment 10010

Vlan 20
 Name outside
 Vn-segment 10020
Interface VLAN 10
 Description inside_vlan
 VRF member INSIDE
 IP address 10.1.1.254/24
 fabric forwarding mode anycast-gateway

Interface loopback100
 Vrf member INSIDE
 Ip address 172.16.1.254/32

Interface VLAN 20
 Description outside_vlan
 VRF member OUTSIDE
 IP address 20.1.1.254/24
 fabric forwarding mode anycast-gateway

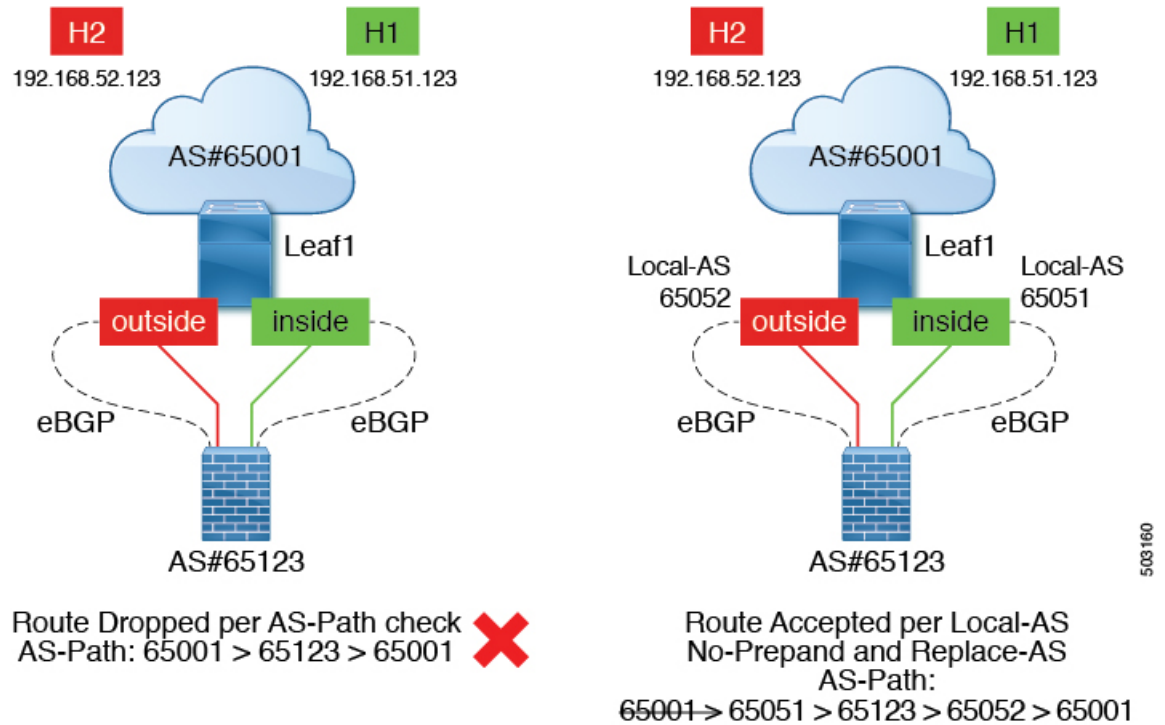
Interface loopback101
 Vrf member OUTSIDE
 Ip address 172.18.1.254/32

router bgp 65000
 vrf INSIDE
 ! peer with Firewall Inside
 neighbor 10.1.1.0/24 remote-as 65123
 update-source loopback100
 ebgp-multihop 5
 address-family ipv4 unicast
 local-as 65051 no-prepend replace-as

 vrf OUTSIDE
 ! peer with Firewall Outside
 neighbor 20.1.1.0/24 remote-as 65123
 update-source loopback101
 ebgp-multihop 5
 address-family ipv4 unicast
 local-as 65052 no-prepend replace-as
```

With the VXLAN fabric generally being in a single BGP Autonomous System (AS), the AS of the inside VRF and the outside VRF is the same. BGP does not install routes that are received from its own AS. Therefore, we need to adjust the AS-path to override this rule. Various approaches exist, including disabling the rule that BGP drops routes from its own AS, which has further implications to the network. To keep all of the BGP protection mechanics in place, the “local-as” approach allows you to mimic routes being originated from a different AS. We recommend inserting the “local-as #ASN# no-prepend replace-as” on each firewall peering with different “local-as” per VRF.

Figure 36: eBGP AS-Path Check



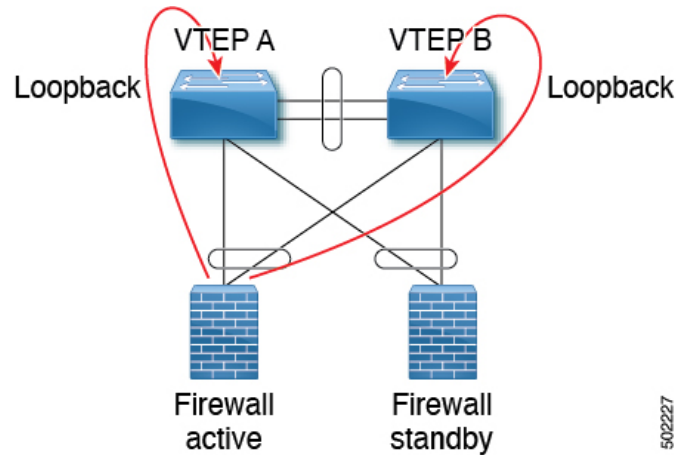
## Dual-Attached Firewall with eBGP Routing

If the firewall supports BGP, one option is to use BGP as a protocol between the firewall and the service VTEP. Peering using the anycast IP is not supported. The recommended design is to use dedicated loopback IPs on each VTEP and peer using the loopback. As long as the loopback interfaces are not advertised via EVPN, the same IP address could be used on all of the belonging VTEPs. We recommend using individual IP addresses on a per-VTEP basis. For vPC environments, it is required.

Reachability to the loopback from the firewall can be configured using a static route on the firewall, pointing to the Anycast Gateway IP on the VTEPs.

In vPC deployments, you must have a per-VRF peering via a vPC peer-link. In addition to the per-VRF peering, you can enable the advertisement of prefix routes (EVPN Route-Type 5) using the **advertise-pip** command. For vPC with fabric peering, the per-VRF peering is not necessary, and the advertisement of prefix routes (EVPN Route-Type5) is required.

In the following example, an eBGP peering is established from the VTEPs, which are in AS 65000, and the firewall in AS 65002. The BGP peering with iBGP is not supported.

**Figure 37: Dual-Attached Firewall with eBGP**

**Note** When having eBGP peering to active/standby firewalls connected to different VTEPs, **export-gateway-ip** must be enabled.

Do not use Anycast Gateway for BGP peerings.

#### VTEP A:

```
Vlan 10
 Name inside
 Vn-segment 10010

Vlan 20
 Name outside
 Vn-segment 10020

Interface VLAN 10
 Description inside_vlan
 VRF member INSIDE
 IP address 10.1.1.254/24
 fabric forwarding mode anycast-gateway

Interface loopback100
 Vrf member INSIDE
 Ip address 172.16.1.253/32

Interface VLAN 20
 Description outside_vlan
 VRF member OUTSIDE
 IP address 20.1.1.254/24
 fabric forwarding mode anycast-gateway

Interface loopback101
 Vrf member OUTSIDE
 Ip address 172.18.1.253/32

router bgp 65000
vrf INSIDE
 ! peer with Firewall Inside
```

```

neighbor 10.1.1.0/24 remote-as 65123
update-source loopback100
ebgp-multihop 5
address-family ipv4 unicast
 local-as 65051 no-prepend replace-as

vrf OUTSIDE
 ! peer with Firewall Outside
neighbor 20.1.1.0/24 remote-as 65123
update-source loopback101
ebgp-multihop 5
address-family ipv4 unicast
 local-as 65052 no-prepend replace-as

```

**VTEP B:**

```

Vlan 10
 Name inside
 Vn-segment 10010

Vlan 20
 Name outside
 Vn-segment 10020

Interface VLAN 10
 Description inside_vlan
 VRF member INSIDE
IP address 10.1.1.254/24
fabric forwarding mode anycast-gateway

Interface loopback100
 Vrf member INSIDE
 Ip address 172.16.1.254/32

Interface VLAN 20
 Description outside_vlan
 VRF member OUTSIDE
IP address 20.1.1.254/24
fabric forwarding mode anycast-gateway

Interface loopback101
 Vrf member OUTSIDE
 Ip address 172.18.1.254/32

router bgp 65000
 vrf INSIDE
 ! peer with Firewall Inside
 neighbor 10.1.1.0/24 remote-as 65123
 update-source loopback100
 ebgp-multihop 5
 address-family ipv4 unicast
 local-as 65051 no-prepend replace-as

 vrf OUTSIDE
 ! peer with Firewall Outside
 neighbor 20.1.1.0/24 remote-as 65123
 update-source loopback101
 ebgp-multihop 5
 address-family ipv4 unicast
 local-as 65052 no-prepend replace-as

```



## Per-VRF Peering via vPC Peer-Link

### VTEP A and VTEP B:

```

vlan 3966
! vlan use for peering between the vPC VTEPS

vlan 3967
! vlan use for peering between the vPC VTEPS

system nve infra-vlans 3966,3967

interface vlan 3966
vrf member INSIDE
ip address 100.1.1.1/31

interface vlan 3967
vrf member OUTSIDE
ip address 100.1.2.1/31

router bgp 65000
vrf INSIDE
neighbor 100.1.1.0 remote-as 65000
update-source vlan 3966
next-hop self
address-family ipv4 unicast

vrf OUTSIDE
neighbor 100.1.2.0 remote-as 65000
update-source vlan 3967
next-hop self
address-family ipv4 unicast

```

The routes learned in each VRF are advertised to the rest of the fabric via BGP EVPN updates.

## Single-Attached Firewall with OSPF

The following example shows a configuration snippet from VTEP A running OSPF peering with the firewall.

SVIs are defined on the VTEP for both inside and outside VRFs. The VTEP peers with the firewall on each of these VRFs dynamically learn routing information to go from one VRF to the other.

### VTEP A and VTEP B:

```

vlan 10
name inside
vn-segment 10010

vlan 20
name outside
vn-segment 10020

interface VLAN 10
Description inside_vlan
VRF member INSIDE
IP address 10.1.1.254/24
IP router ospf 1 area 0
fabric forwarding mode anycast-gateway

Interface VLAN 20
Description outside_vlan
VRF member OUTSIDE

```

```

IP address 20.1.1.254/24
IP router ospf 1 area 0
fabric forwarding mode anycast-gateway

interface nve1
no shutdown
host-reachability protocol bgp
source-interface loopback1
member vni 10010
 mcastgroup 239.1.1.1
member vni 10020
 mcastgroup 239.1.1.1
member vni 1001000 associate-vrf
member vni 1002000 associate-vrf

router ospf 1
router-id 192.168.1.1
vrf INSIDE
VRF OUTSIDE

VTEPA# show ip route ospf-1 vrf OUTSIDE
IP Route Table for VRF "OUTSIDE"
'*' denotes best ucast next-hop
***' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

10.1.1.0/24, ubest/mbest: 1/0
 *via 20.1.1.1 Vlan20, [110/41], 1w5d, ospf-1, intra

VTEPA# show ip route ospf-1 vrf INSIDE
IP Route Table for VRF "INSIDE"
'*' denotes best ucast next-hop
***' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

20.1.1.0/24, ubest/mbest: 1/0
 *via 10.1.1.1 Vlan10, [110/41], 1w5d, ospf-1, intra

```

This route is then redistributed into BGP and advertised through the EVPN fabric so that all other VTEPs have all routes in each VRF pointing to VTEP A as the next hop.

## Redistribute OSPF Routes into BGP and Advertise to the Rest of the Fabric

### VTEP A and VTEP B:

```

router bgp 65000
vrf OUTSIDE
 address-family ipv4 unicast
 redistribute ospf 1 route-map OUTSIDEOSPF-to-BGP
vrf INSIDE
 address-family ipv4 unicast
 redistribute ospf 1 route-map INSIDEOSPF-to-BGP

VTEPA# show ip route 10.1.1.0/24 vrf OUTSIDE

10.1.1.0/24 ubest/mbest: 1/0
 *via 10.1.1.18%default, [200/41], 1w1d, bgp-65000, internal, tag 65000 (evpn) segid:
200100 tunnelid: 0xa010112 encap: VXLAN

```

Traffic is VXLAN encapsulated from VTEP to services VTEP and decapsulated and sent to the firewall. The firewall enforces the rules and sends the traffic to the services VTEP on the inside VRF. This traffic is then VXLAN encapsulated and sent to the destination VTEP where traffic is decapsulated and sent to the end client.

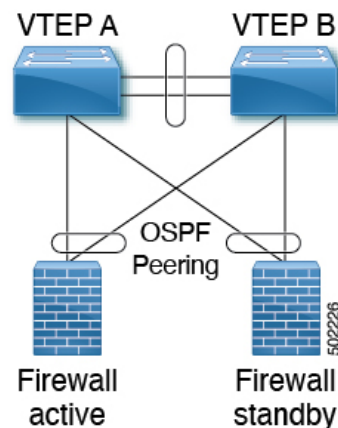
### Firewall Failover

When the active firewall fails and the standby firewall takes over, routes are withdrawn from service VTEP A and advertised to the fabric by service VTEP B.

## Dual-Attached Firewall with OSPF

Cisco NX-OS supports dynamic OSPF peering over vPC using Layer 3, which enables firewall connectivity using vPC and establishes OSPF peering over this link. The VLAN used to establish peering between the Cisco Nexus 9000 switches and the firewall must be a non-VXLAN-enabled VLAN.

**Figure 38: Dual-Attached Firewall with OSPF**



**Note** Do not use Anycast Gateway for OSPF adjacencies.

### VTEP A:

```
Vlan 10
 Name inside

Vlan 20
 Name outside

Interface VLAN 10
 Description inside_vlan
 VRF member INSIDE
 IP address 10.1.1.253/24
 Ip router ospf 1 area 0

Interface VLAN 20
 Description outside_vlan
 VRF member OUTSIDE
 IP address 20.1.1.253/24
 Ip router ospf 1 area 0
```

```

vpc domain 100
 layer3 peer-router
 peer-gateway
 peer-switch
 peer-keepalive destination x.x.x.x source x.x.x.x peer-gateway
 ipv6 nd synchronize
 ip arp synchronize

router ospf 1
 vrf INSIDE VRF OUTSIDE

```

### VTEP B:

```

Vlan 10
 Name inside

Vlan 20
 Name outside

Interface VLAN 10
 Description inside_vlan
 VRF member INSIDE
 IP address 10.1.1.254/24
 Ip router ospf 1 area 0

Interface VLAN 20
 Description outside_vlan
 VRF member OUTSIDE
 IP address 20.1.1.254/24
 Ip router ospf 1 area 0

vpc domain 100
 layer3 peer-router
 peer-gateway
 peer-switch
 peer-keepalive destination x.x.x.x source x.x.x.x peer-gateway
 ipv6 nd synchronize
 ip arp synchronize

router ospf 1
 vrf INSIDE VRF OUTSIDE

VTEPA# show ip route ospf-1 vrf OUTSIDE
IP Route Table for VRF "OUTSIDE"
 '*' denotes best ucast next-hop
 '**' denotes best mcast next-hop
 '[x/y]' denotes [preference/metric]
 '%<string>' in via output denotes VRF <string>

10.1.1.0/24, ubest/mbest: 1/0
 *via 20.1.1.1 Vlan20, [110/41], 1w5d, ospf-1, intra

VTEPA# show ip route ospf-1 vrf INSIDE
IP Route Table for VRF "INSIDE"
 '*' denotes best ucast next-hop
 '**' denotes best mcast next-hop
 '[x/y]' denotes [preference/metric]
 '%<string>' in via output denotes VRF <string>

20.1.1.0/24, ubest/mbest: 1/0
 *via 10.1.1.1 Vlan10, [110/41], 1w5d, ospf-1, intra

```

## Redistribute OSPF Routes into BGP and Advertise to the Rest of the Fabric

### VTEP A and VTEP B:

```
router bgp 65000
vrf OUTSIDE
 address-family ipv4 unicast
 redistribute ospf 1 route-map OUTSIDEOSPF-to-BGP
vrf INSIDE
 address-family ipv4 unicast
 redistribute ospf 1 route-map INSIDEOSPF-to-BGP
```

## Firewall as Default Gateway

In this deployment model, the VXLAN fabric is a Layer 2 fabric, and the default gateway resides on the firewall.

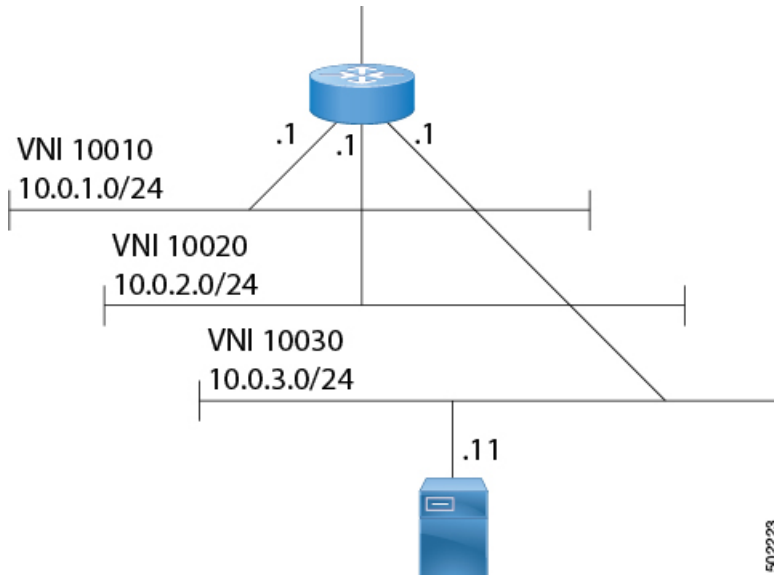
For example:

```
vlan 10
 name WEB
 vn-segment 10010
vlan 20
 name APPLICATION
 vn-segment 10020
vlan 30
 name DATABASE
 vn-segment 10030

interface nve1
 no shutdown
 host-reachability protocol bgp
 source-interface loopback1
 member vni 10010
 mcastgroup 239.1.1.1
 member vni 10020
 mcastgroup 239.1.1.1
 member vni 10030
 mcastgroup 239.1.1.1
```

The firewall has a logical interface in each VNI and is the default gateway for all endpoints. Every inter-VNI communication flows through the firewall. Take special care with the sizing of the firewall so that it does not become a bottleneck. Therefore, use this design in environments with low-bandwidth requirements.

Figure 39: Firewall as Default Gateway with a Layer-2 VXLAN Fabric



## Transparent Firewall Insertion

Transparent firewalls or Layer 2 firewalls (including IPS/IDS) typically bridge between an inside VLAN and outside VLAN and inspect traffic as it traverses through them. VLAN stitching is done by placing the default gateway for the service on the inside VLAN. The Layer 2 reachability to this gateway is done on the outside VLAN.

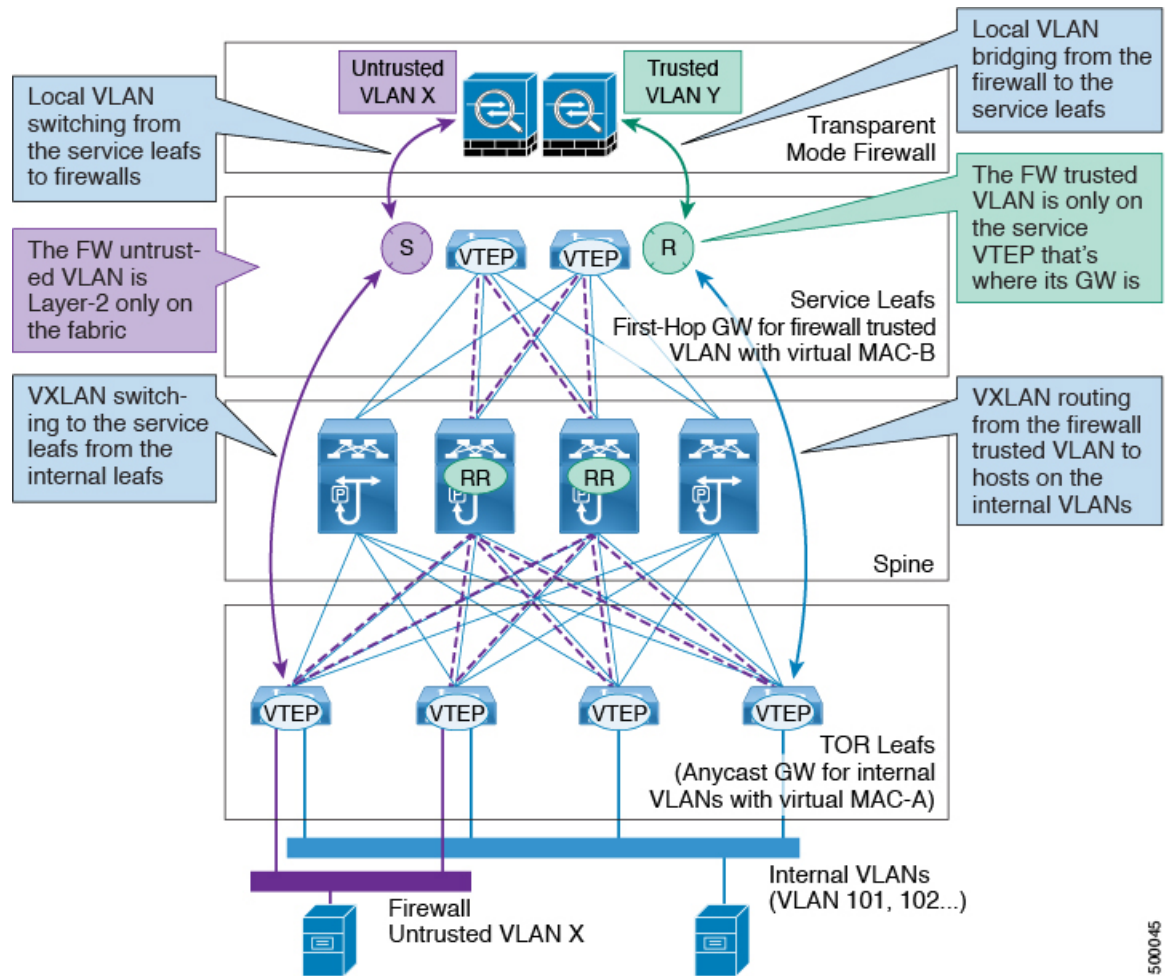
## Overview of EVPN with Transparent Firewall Insertion

The topology contains the following types of VLANs:

- Internal VLAN (a regular VXLAN on ToR leafs with Anycast Gateway)
- Firewall untrusted VLAN X
- Firewall trusted VLAN Y

In this topology, the traffic that goes from VLAN X to other VLANs must go through a transparent Layer 2 firewall that is attached to the service leafs. This topology utilizes an approach of an untrusted VLAN X and a trusted VLAN Y. All ToR leafs have a Layer 2 VNI VLAN X. There is no SVI for VLAN X. The service leafs that are connected to the firewall have Layer 2 VNI VLAN X, non-VXLAN VLAN Y, and SVI Y with an HSRP gateway.

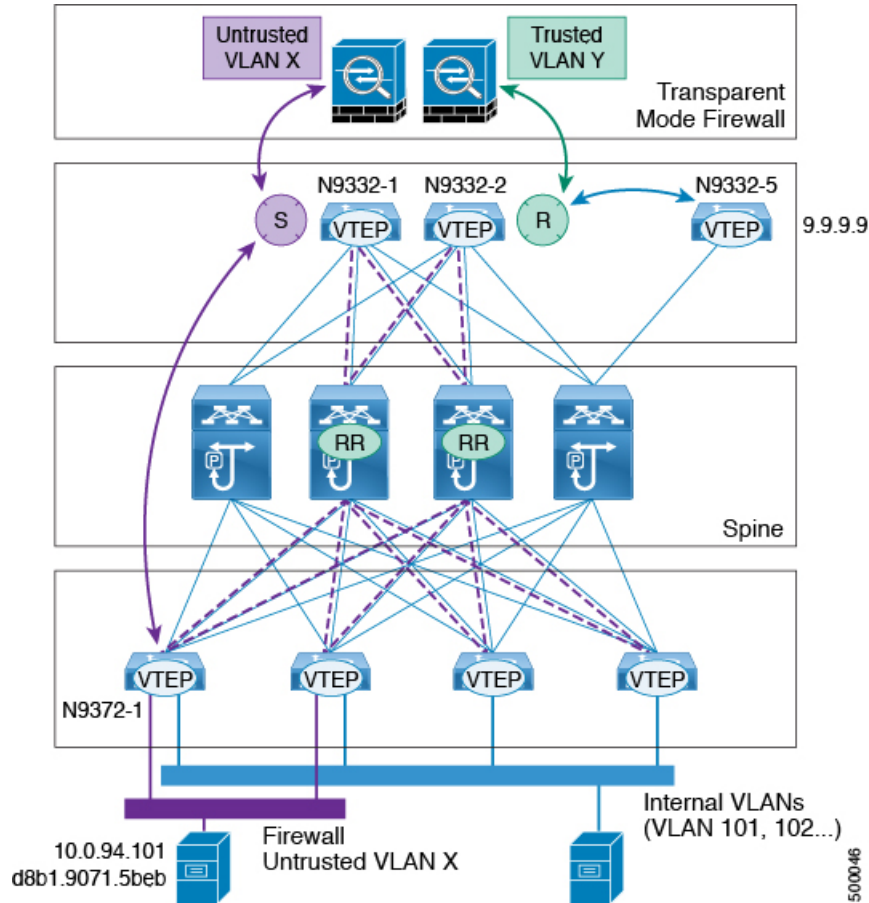
## Overview of EVPN with Transparent Firewall Insertion

**Note**

For VXLAN EVPN, we recommend using the distributed Anycast Gateway with transparent firewall insertion. Doing so allows all VLANs to be VXLAN enabled. When using an HSRP/VRRP-based First-Hop Gateway, the VLAN for the SVI can't be VXLAN enabled and should reside on a vPC pair for redundancy.

## EVPN with Transparent Firewall Insertion Example

### Example of EVPN with Transparent Firewall Insertion



- Host in VLAN X: 10.1.94.101
- ToR leaf: N9372-1
- Service leaf in vPC: N9332-1 and N9332-2
- Border leaf: N9332-5

### ToR Leaf Configuration

```

vlan 94
vn-segment 100094

interface nve1
 member vni 100094
 mcastgroup 239.1.1.1

router bgp 64500
 routerid 1.1.2.1
 neighbor 1.1.1.1 remote-as 64500
 address-family l2vpn evpn
 send-community extended

```



```
neighbor 1.1.1.2 remote-as 64500
address-family l2vpn evpn
 send-community extended
vrf Ten1
 address-family ipv4 unicast
 advertise l2vpn evpn

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

### Service Leaf 1 Configuration Using HSRP

```
vlan 94
description untrusted_vlan
vn-segment 100094

vlan 95
description trusted_vlan

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

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

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

interface nve1
member vni 100094
mcast-group 239.1.1.1

router bgp 64500
routerid 1.1.2.1
neighbor 1.1.1.1 remote-as 64500
address-family l2vpn evpn
 send-community extended
neighbor 1.1.1.2 remote-as 64500
address-family l2vpn evpn
 send-community extended
vrf Ten-1
 address-family ipv4 unicast
```

```

 network 10.0.94.0/24 /*advertise /24 for SVI 95 subnet; it is not VXLAN anymore*/
 advertise l2vpn evpn

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

```

### Service Leaf 2 Configuration Using HSRP

```

vlan 94
 description untrusted_vlan
 vnsegment 100094

vlan 95
 description trusted_vlan

vpc domain 10
 peer-switch
 peer-keepalive destination 10.1.59.159
 peer-gateway
 auto-recovery
 ip arp synchronize

interface Vlan2
 description vpc_backup_svi_for_overlay
 no shutdown
 no ip redirects
 ip address 10.10.60.18/30
 no ipv6 redirects
 ip router ospf 100 area 0.0.0.0
 ip pim sparsemode

interface Vlan95
 description SVI_for_trusted_vlan
 no shutdown
 mtu 9216
 vrf member Ten-1
 no ip redirects
 ip address 10.0.94.3/24
 hsrp 0
 preempt priority 255
 ip 10.0.94.1

interface nve1
 member vni 100094
 mcastgroup 239.1.1.1

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

evpn
vni 100094 l2

```

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

## Show Command Examples

Display information about the ingress leaf learned local MAC from host:

```
switch# sh mac add vl 94 | i 5b|MAC
* primary entry, G - Gateway MAC, (R) Routed - MAC, O - Overlay MAC
VLAN MAC Address Type age Secure NTFY Ports
* 94 d8b1.9071.5beb dynamic 0 F F Eth1/1
```

Display information about the service leaf found MAC of host:



**Note** In VLAN 94, the service leaf learned the host MAC from the remote peer by BGP.

```
switch# sh mac add vl 94 | i VLAN|eb

VLAN MAC Address Type age Secure NTFY Ports
* 94 d8b1.9071.5beb dynamic 0 F F nvel(1.1.2.1)

switch# sh mac add vl 94 | i VLAN|eb

VLAN MAC Address Type age Secure NTFY Ports
* 94 d8b1.9071.5beb dynamic 0 F F nvel(1.1.2.1)

switch# sh mac add vl 95 | i VLAN|eb

VLAN MAC Address Type age Secure NTFY Ports
+ 95 d8b1.9071.5beb dynamic 0 F F Po300

switch# sh mac add vl 95 | i VLAN|eb

VLAN MAC Address Type age Secure NTFY Ports
+ 95 d8b1.9071.5beb dynamic 0 F F Po300
```

Display information about service leaf learned ARP for host on VLAN 95:

```
switch# sh ip arp vrf ten-1
Address Age MAC Address Interface
10.0.94.101 00:00:26 d8b1.9071.5beb Vlan95
```

Service Leaf learns 9.9.9.9 from EVPN.

```
switch# sh ip route vrf ten-1 9.9.9.9
IP Route Table for VRF "Ten-1"
'*' denotes best ucast nexthop
'**' denotes best mcast nexthop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

9.9.9.9/32, ubest/mbest: 1/0
 *via 1.1.2.7%default, [200/0], 02:57:27, bgp64500,internal, tag 65000 (evpn) segid: 10011
tunnelid: 0x1
010207 encap: VXLAN
```

Display information about the border leaf learned host routes by BGP:

```
switch# sh ip route 10.0.94.101

IP Route Table for VRF "default"
'*' denotes best ucast nexthop
'**' denotes best mcast nexthop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

10.0.94.0/24, ubest/mbest: 1/0
 *via 10.100.5.0, [20/0], 03:14:27, bgp65000,external, tag 6450
```



## APPENDIX **D**

# Configuring Multi-Homing

This chapter contains the following sections:

- [VXLAN EVPN Multi-Homing Overview, on page 313](#)
- [Configuring VXLAN EVPN Multi-Homing, on page 317](#)
- [Configuring Layer 2 Gateway STP, on page 319](#)
- [Configuring VXLAN EVPN Multi-Homing Traffic Flows, on page 324](#)
- [Configuring ESI ARP Suppression, on page 335](#)
- [Configuring VLAN Consistency Checking, on page 338](#)

## VXLAN EVPN Multi-Homing Overview

### Introduction to Multi-Homing

Cisco Nexus platforms support vPC-based multi-homing, where a pair of switches act as a single device for redundancy and both switches function in an active mode. With first generation Nexus 9000 Series switches (i.e. any hardware model before -EX) in VXLAN BGP EVPN environment, there are two solutions to support Layer 2 multi-homing; 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 can use LACP to detect the misconfigurations. It also eliminates the MCT link that is traditionally used by vPC and it offers more flexibility as each VTEP can be a part of one or more redundancy groups. It can potentially support many VTEPs in a given group.

### BGP EVPN Multi-Homing

Cisco Nexus 9000 platforms can only interoperate with other VTEPs fully supporting BGP EVPN multi-homing (as it will be clarified in a later section of this document) for the specific endpoints/legacy switches southbound attachment use case. To better understand how this interoperability works, this section provides a quick refresh of the basic functionalities offered by BGP EVPN multi-homing

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 multi-homing further provides fast

convergence during certain failure scenarios, that otherwise cannot be achieved without a control protocol (data plane flood and learn).

## BGP EVPN Multi-Homing Terminology

See this section for the terminology used in BGP EVPN multi-homing:

- 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 Multi-Homing 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 - Type-1
- MAC advertisement Route - Type-2
- Inclusive Multicast Route - Type-3
- Ethernet Segment Route - Type-4
- IP Prefix Route - Type-5

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 multi-homing, 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 multi-homing, 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 multi-homing:

- 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 Multi-Homing Redundancy Group

Consider a dual-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 same considerations below apply when the host H2 is directly dual-homed to the switches L1 and L2.

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

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

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

## Ethernet Segment Identifier

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

## LACP Bundling

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

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

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

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

## Guidelines and Limitations for VXLAN EVPN Multi-Homing

See the following limitations for configuring VXLAN EVPN multi-homing:

- EVPN multi-homing is only supported on first generation Cisco Nexus 9300 platform switches. It is not supported on Cisco Nexus 9200 switches nor on Cisco Nexus 9300-EX switches (and newer models).
- Beginning with Cisco NX-OS Release 9.2(3), a FEX member port on a VXLAN VLAN with peer-link less vPC/vPC<sup>2</sup> is not supported.
- VXLAN EVPN multi-homing works with the iBGP or eBGP control plane. iBGP is preferred.
- If iBGP is used with VXLAN EVPN multi-homing, 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 multi-homing, 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 multi-homing requires that all switches in a given network must be EVPN multi-homing capable. Mixing platforms with and without EVPN multi-homing is not supported.
- EVPN multi-homing is not supported on FEX.
- ARP suppression is supported with EVPN multi-homing.
- EVPN multi-homing is supported with multi-homing to two switches only.
- To enable EVPN multi-homing, 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 Multi-Homing

### Enabling EVPN Multi-Homing

Cisco NX-OS allows either vPC based EVPN multi-homing or ESI based EVPN multi-homing. Both features should not be enabled together. ESI based multi-homing is enabled using **evpn esi multihoming** CLI command. It is important to note that the command for ESI multi-homing 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 multi-homing:

#### Before you begin

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

#### Procedure

|               | Command or Action                                                                                                                                                                                      | Purpose                                                                                                                                                                         |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 1</b> | <b>evpn esi multihoming</b>                                                                                                                                                                            | Enables EVPN multi-homing globally.                                                                                                                                             |
| <b>Step 2</b> | <b>address-family l2vpn evpn maximum-paths</b><br><b>&lt;&gt;maximum-paths ibgp &lt;&gt;</b><br><br><b>Example:</b><br><br><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. |

|               | Command or Action                                                                                                                                                         | Purpose                                                                                                                                                                                                                                                                                                                      |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 3</b> | <b>evpn multihoming core-tracking</b>                                                                                                                                     | Enables EVPN multi-homing 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.                                                          |
| <b>Step 4</b> | <b>interface port-channel Ethernet-segment</b><br><b>&lt;&gt;System-mac &lt;&gt;</b><br><br><b>Example:</b><br><pre> ethernet-segment 11 system-mac 0000.0000.0011 </pre> | <p>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.</p> <p>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.</p> |
| <b>Step 5</b> | <b>hardware access-list tcam region</b><br><b>vpc-convergence 256</b><br><br><b>Example:</b><br><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 Multi-Homing Configuration Examples

See the sample VXLAN EVPN multi-homing 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

EVPN multi-homing 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 multi-homing VLANs. Use the **spanning-tree domain enable** CLI command to enable L2G-STP on all VTEPs. With L2G-STP enabled, the VXLAN fabric (all VTEPs) emulates a single pseudo root switch for the customer access switches. The L2G-STP is initiated to run on all VXLAN VLANs by default on boot up and the root

is fixed on the overlay. With L2G-STP, the root-guard gets enabled by default on all the access ports. Use **spanning-tree domain** *<id>* to additionally enable Spanning Tree Topology Change Notification(STP-TCN), to be tunneled across the fabric.

All the access ports from VTEPs connecting to the customer access switches are in a *desg* forwarding state by default. All ports on the customer access switches connecting to VTEPs are either in root-port forwarding or alt-port blocking state. The root-guard kicks in if better or superior STP information is received from the customer access switches and it puts the ports in the *blk l2g\_inc* state to secure the root on the overlay-fabric and to prevent a loop.

## Guidelines for Moving to Layer 2 Gateway STP

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

- With Layer 2 Gateway STP, root guard is enabled by default on all the access ports.
- With Layer 2 Gateway STP enabled, the VXLAN fabric (all VTEPs) emulates a single pseudo-root switch for the customer access switches.
- All access ports from VTEPs connecting to the customer access switches are in the **Desg FWD** state by default.
- All ports on customer access switches connecting to VTEPs are either in the root-port FWD or Altn BLK state.
- Root guard is activated if superior spanning-tree information is received from the customer access switches. This process puts the ports in **BLK L2GW\_Inc** state to secure the root on the VXLAN fabric and prevent a loop.
- Explicit domain ID configuration is needed to enable spanning-tree BPDU tunneling across the fabric.
- As a best practice, you should configure all VTEPs with the lowest spanning-tree priority of all switches in the spanning-tree domain to which they are attached. By setting all the VTEPs as the root bridge, the entire VXLAN fabric appears to be one virtual bridge.
- ESI interfaces should not be enabled in spanning-tree edge mode to allow Layer 2 Gateway STP to run across the VTEP and access layer.
- You can continue to use ESIs or orphans (single-homed hosts) in spanning-tree edge mode if they directly connect to hosts or servers that do not run Spanning Tree Protocol and are end hosts.
- Configure all VTEPs that are connected by a common customer access layer in the same Layer 2 Gateway STP domain. Ideally, all VTEPs on the fabric on which the hosts reside and to which the hosts can move.
- The Layer 2 Gateway STP domain scope is global, and all ESIs on a given VTEP can participate in only one domain.
- Mappings between Multiple Spanning Tree (MST) instances and VLANs must be consistent across the VTEPs in a given Layer 2 Gateway STP domain.
- Non-Layer 2 Gateway STP enabled VTEPs cannot be directly connected to Layer 2 Gateway STP-enabled VTEPs. Performing this action results in conflicts and disputes because the non-Layer 2 Gateway STP VTEP keeps sending BPDUs and it can steer the root outside.
- Ensure that the root of an STP domain local to the VXLAN fabric is a VTEP or placed within the fabric.

- 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                                                                                                      |
|---------------|----------------------------------------------------|--------------------------------------------------------------------------------------------------------------|
| <b>Step 1</b> | <b>spanning-tree mode &lt;rapid-pvst, mst&gt;</b>  | Enables Spanning Tree Protocol mode.                                                                         |
| <b>Step 2</b> | <b>spanning-tree domain enable</b>                 | Enables Layer 2 Gateway STP on a switch. It disables Layer 2 Gateway STP on all EVPN ESI multi-homing VLANs. |
| <b>Step 3</b> | <b>spanning-tree domain 1</b>                      | Explicit domain ID is needed to tunnel encoded BPDUs to the core and processes received from the core.       |
| <b>Step 4</b> | <b>spanning-tree mst &lt;id&gt; priority 8192</b>  | Configures Spanning Tree Protocol priority.                                                                  |
| <b>Step 5</b> | <b>spanning-tree vlan &lt;id&gt; priority 8192</b> | Configures Spanning Tree Protocol priority.                                                                  |
| <b>Step 6</b> | <b>spanning-tree domain disable</b>                | 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.bdlb.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 Multi-Homing Traffic Flows

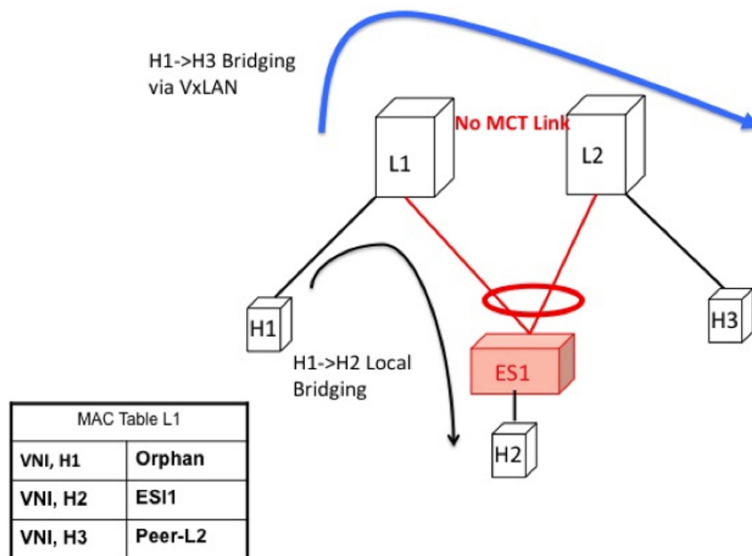
## EVPN Multi-Homing Local Traffic Flows

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

### Locally Bridged Traffic

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

**Figure 40: Local Bridging at L1. H1->H3 bridging via VXLAN. In vPC, H1->H3 will be via MCT link.**



### Access Failure for Locally Bridged Traffic

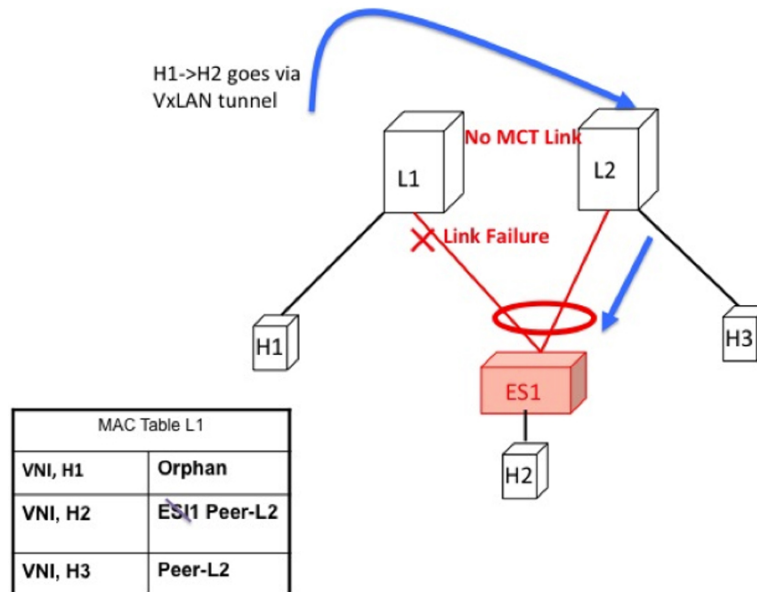
If the ESI link at L1 fails, there is no path for the bridged traffic to reach from H1 to H2 except via the overlay. Therefore, the local bridged traffic takes the sub-optimal path, similar to the H1 to H3 orphan flow.



**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.



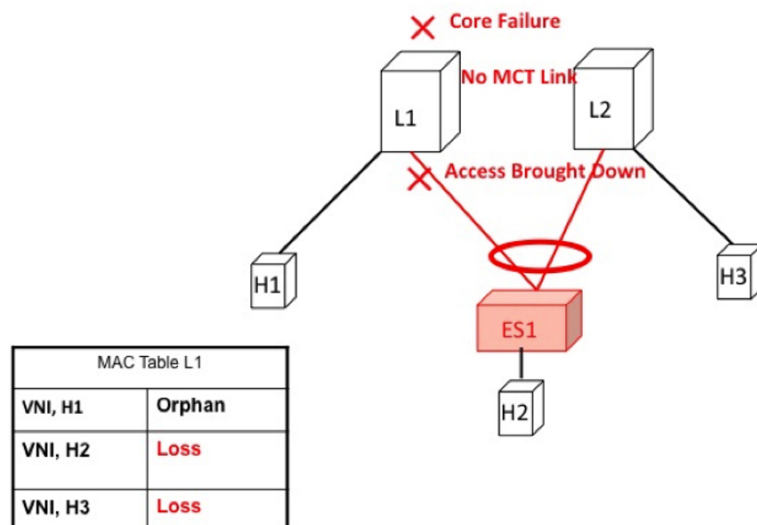
Figure 41: ES1 failure on L1. H1->H2 is now bridged over VXLAN tunnel.



### 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 MCT link.

Figure 42: Core failure on L1. H1->H2 loses all connectivity as there is no MCT.



### Locally Routed Traffic

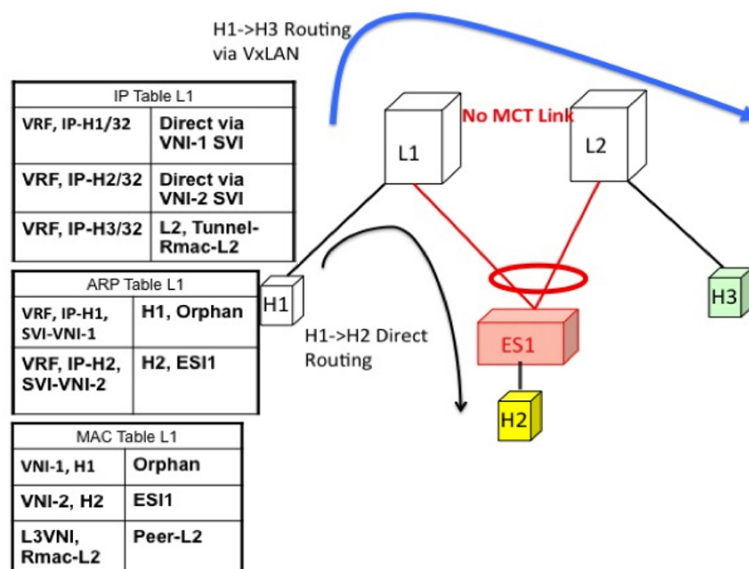
Consider H1, H2, and H3 being in different subnets and L1/L2 being distributed anycast gateways.

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.

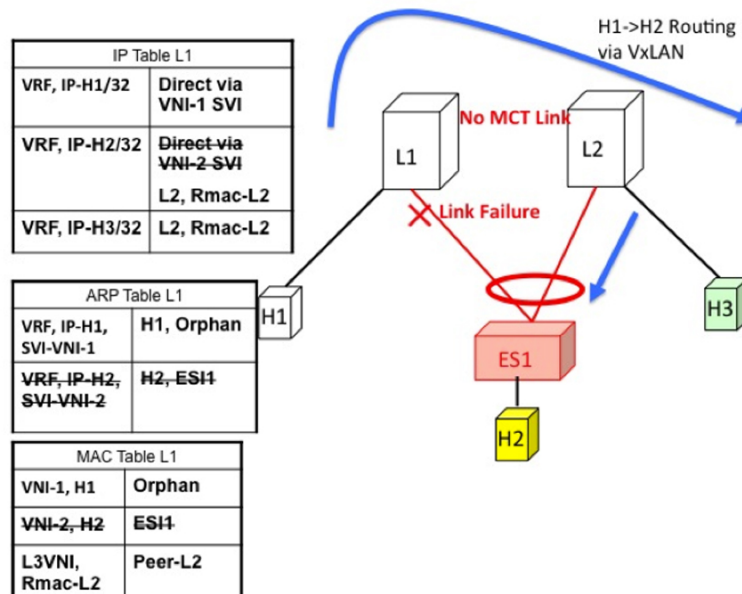
**Figure 43: L1 is Distributed Anycast Gateway. H1, H2, and H3 are in different VLANs. H1->H3 routing happens via VXLAN tunnel encapsulation. In vPC, H3 ARP would have been synced via MCT and direct routing.**



### Access Failure for Locally Routed Traffic

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

Figure 44: H1, H2, and H3 are in different VLANs. ES1 fails on L1. H1->H2 routing happens via VXLAN tunnel encapsulation.

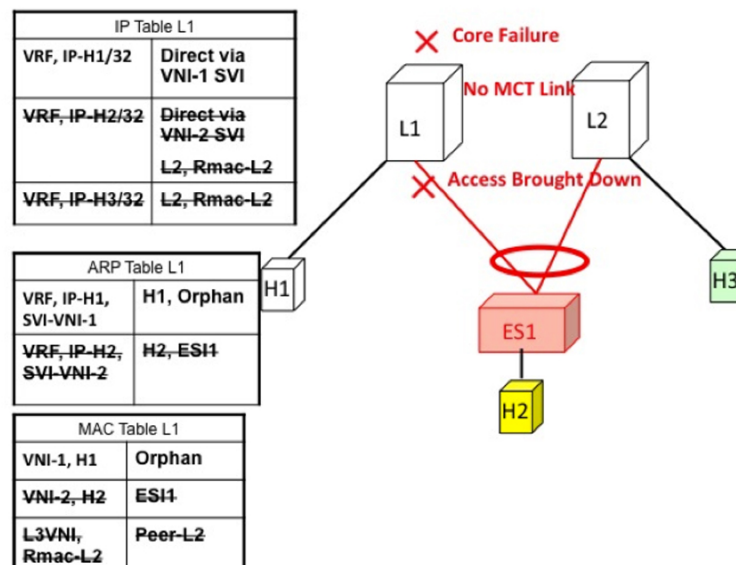


### Core Failure for Locally Routed Traffic

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 MCT link.

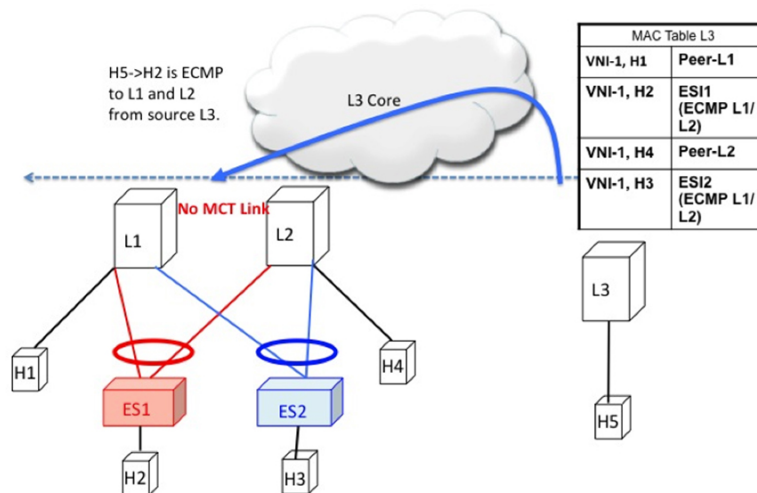
Figure 45: H1, H2, and H3 are in different VLANs. Core fails on L1. Access is brought down. H1 loses all connectivity.



## EVPN Multi-Homing 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 46: 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<br>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)

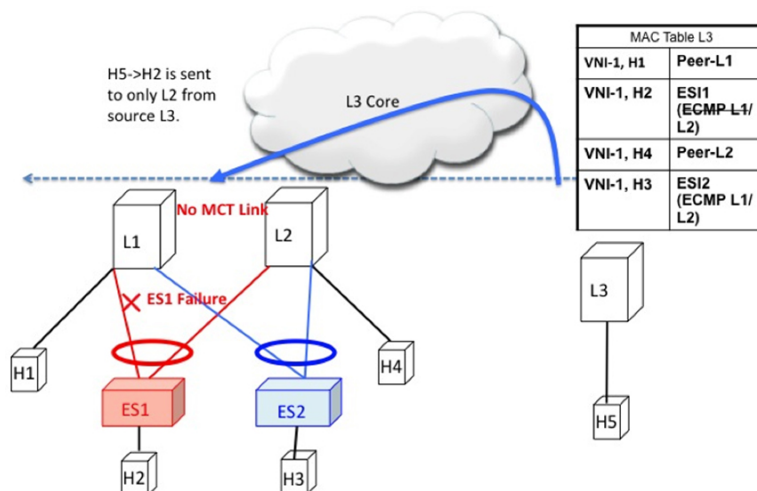
MAC-IP Route remains the same as used in the current vPC multi-homing and NX-OS single-homing solutions. However, now it has a non-zero ESI field that indicates that this is a multihomed host and it is a candidate for ECMP Path Resolution.

| 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<br>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.

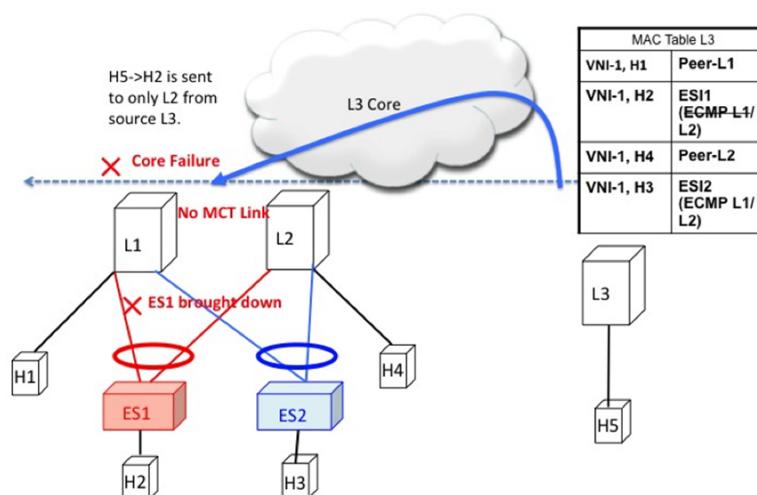
**Figure 47: Layer 2 VXLAN Gateway. ESI failure on L1. L3 withdraws L1 from MAC ECMP list. This will happen due to EAD/ES mass withdrawal from L1.**



### 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 48: 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.**

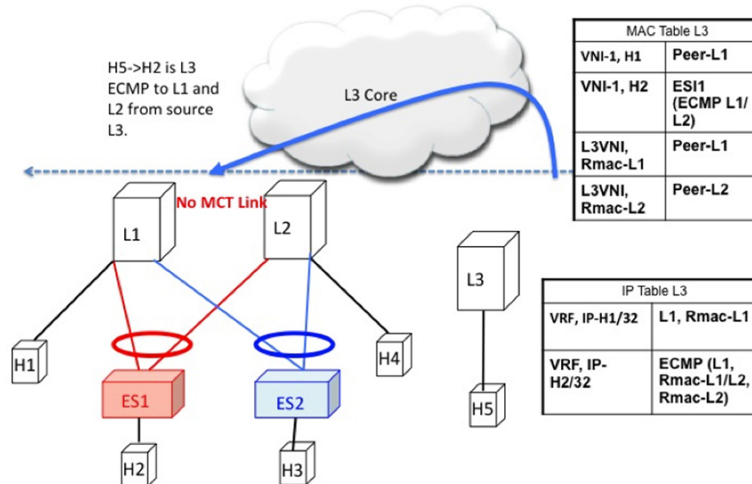


### Remote Routed Traffic

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 49: 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.

The diagram illustrates a network configuration involving a central L3 Core (cloud) and two IP Tables (L1 and L3). The L3 Core is connected to two IP Tables, L1 and L3, via a blue arrow labeled "L3 Core". The L3 Core is also connected to two IP Tables, L1 and L3, via a blue arrow labeled "L3 Core".

**IP Table L1**

| VRF, IP-H2/32 | L1, Rmac-L2 |
|---------------|-------------|
| VRF, IP-H2/32 | L1, Rmac-L2 |

**IP Table L3**

| VRF, IP-H1/32 | L1, Rmac-L2                    |
|---------------|--------------------------------|
| VRF, IP-H1/32 | L1, Rmac-L2                    |
| VRF, IP-H2/32 | ECMP (L1, Rmac-L1/L2, Rmac-L2) |

**Network Topology and Connections:**

- L1 and L2:** Two routers (L1 and L2) are connected via a blue arrow labeled "No MCT Link".
- ES1 and ES2:** Two edge switches (ES1 and ES2) are connected via a blue arrow labeled "ES1 Failure".
- H1, H2, H3, H4, H5:** Five hosts (H1, H2, H3, H4, H5) are connected to the network. H1 and H2 are connected to L1. H3 and H4 are connected to L2. H5 is connected to L3.
- L3 Core:** A central cloud representing the L3 Core is connected to L1 and L2 via blue arrows.

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.

H5->H2 inter subnet traffic sent only to L2

L3 Core

Core Failure

No MCT Link

ES1 brought down

ES2 brought up

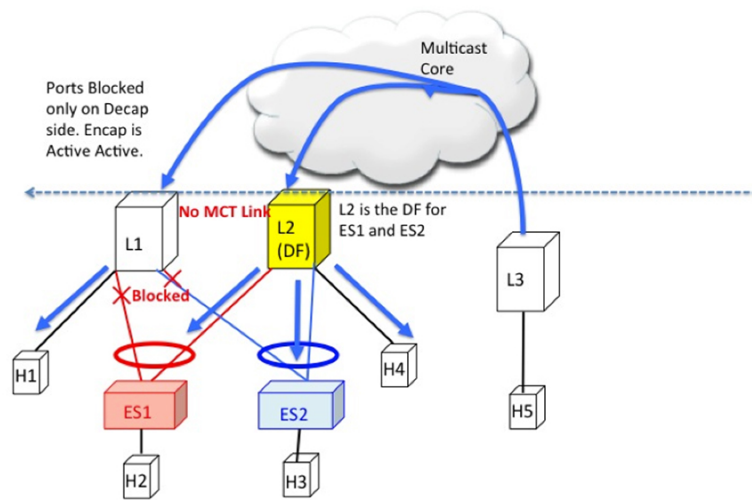
|                |                  |
|----------------|------------------|
| VNI-1, H1      | Peer-L1          |
| VNI-1, H2      | ES1 (ECMP-L1/L2) |
| L3VNI, Rmac-L1 | Peer-L1          |
| L3VNI, Rmac-L2 | Peer-L2          |

|               |                                  |
|---------------|----------------------------------|
| VRF, IP-H1/32 | L1, Rmac-L1                      |
| VRF, IP-H2/32 | ECMP (L1-1, Rmac-L1/L2, Rmac-L2) |

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 52: 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.

Ports Blocked only on Decap side. Encap is Active Active.

Multicast Core

No MCT Link

L2 is the DF for ES1 and ES2

Local Bias

Split Horizon

L1

L2 (DF)

L3

H1

H2

H3

H4

H5

ES1

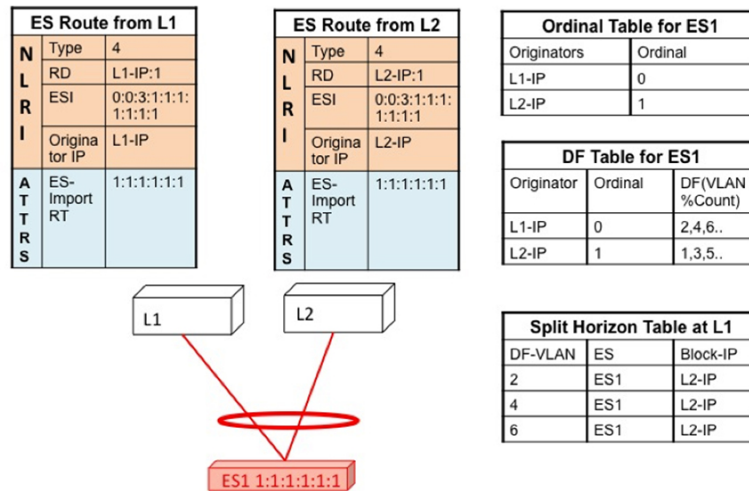
ES2

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.

## DF Election and VLAN Carving

334

Figure 54: 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 multi-homing 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 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 multi-homing 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 only in L3 mode.

## Limitations for ESI ARP Suppression

See the following limitations for ESI ARP suppression:

- ESI multi-homing 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<br>Vtep Addr | 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
 Ip Address Age Mac Address Vlan Physical-ifindex Flags
Remote Vtep Addr
61.1.1.40 00:48:37 0000.0610.0040 610 (null) R
VTEP1, VTEP2.. VTEPn

```

```

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

```

```

 Ip Address Age Mac Address Vlan Physical-ifindex Flags
Remote Vtep Addr
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#
```

# Configuring VLAN Consistency Checking

## Overview of VLAN Consistency Checking

In a typical multi-homing 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 multi-homing 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 multi-homing peers, a broadcast request with a snapshot of configured and administratively up VLANs for the ethernet-segment (ES) is sent to all the CFS peers.

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

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

VLAN consistency checking runs for the following events:

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

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

## VLAN Consistency Checking Guidelines and Limitations

See the following guidelines and limitations for VLAN consistency checking:

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

## Configuring VLAN Consistency Checking

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

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

```
switch (config)# sh running-config | in cfs
cfs ipv4 mcast-address 239.255.200.200
cfs ipv4 distribute

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 Vlan: 3001-3002
 DF Vlan: 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.
```





## APPENDIX **E**

# Configuring Proportional Multipath for VNF

---

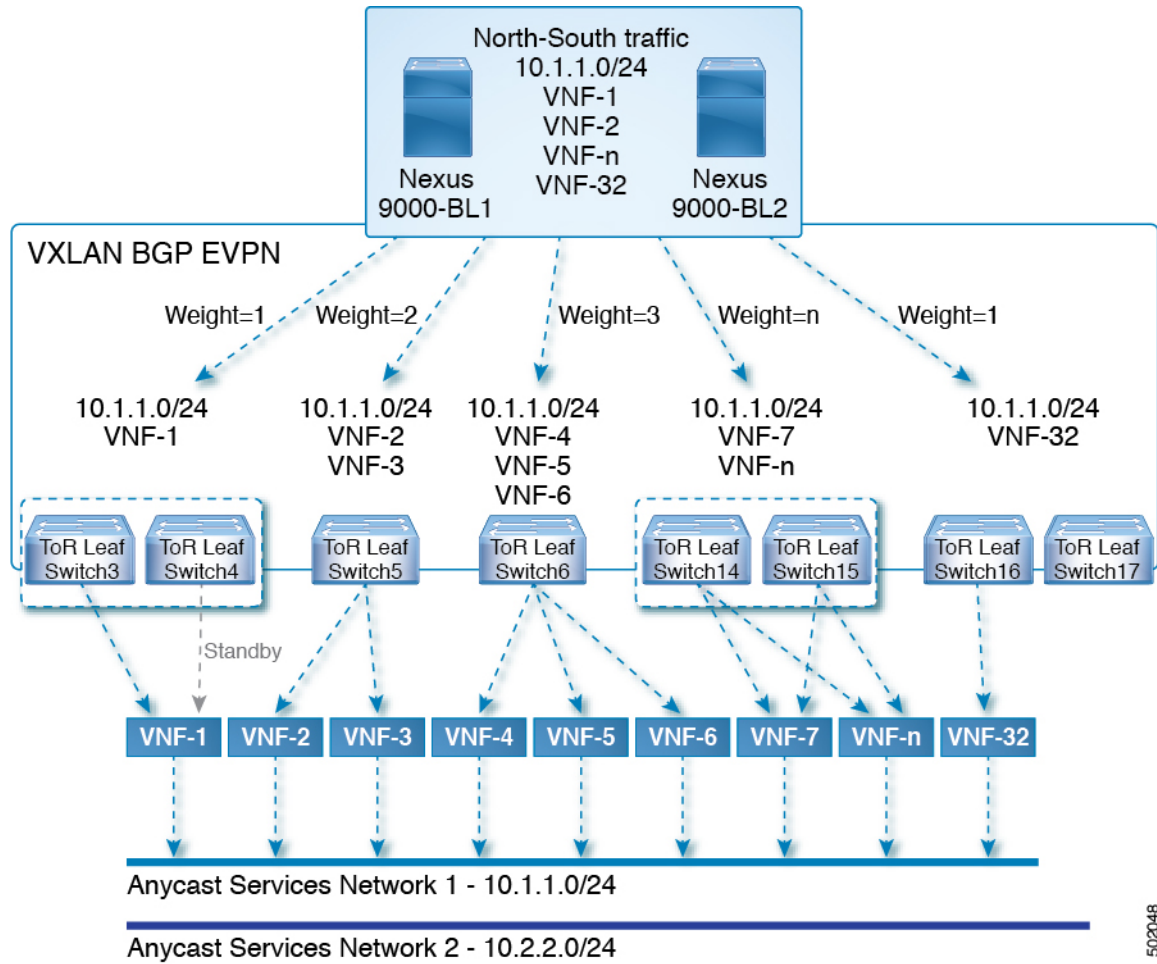
This chapter contains the following sections:

- [About Proportional Multipath for VNF, on page 341](#)
- [Guidelines and Limitations for Proportional Multipath for VNF, on page 345](#)
- [Configuring the Route Reflector, on page 346](#)
- [Configuring the ToR, on page 347](#)
- [Configuring the Border Leaf, on page 350](#)
- [Configuring the BGP Legacy Peer, on page 354](#)
- [Configuring a User-Defined Profile for Maintenance Mode, on page 355](#)
- [Configuring a User-Defined Profile for Normal Mode, on page 356](#)
- [Configuring a Default Route Map, on page 356](#)
- [Applying a Route Map to a Route Reflector, on page 356](#)
- [Verifying Proportional Multipath for VNF, on page 357](#)

## About Proportional Multipath for VNF

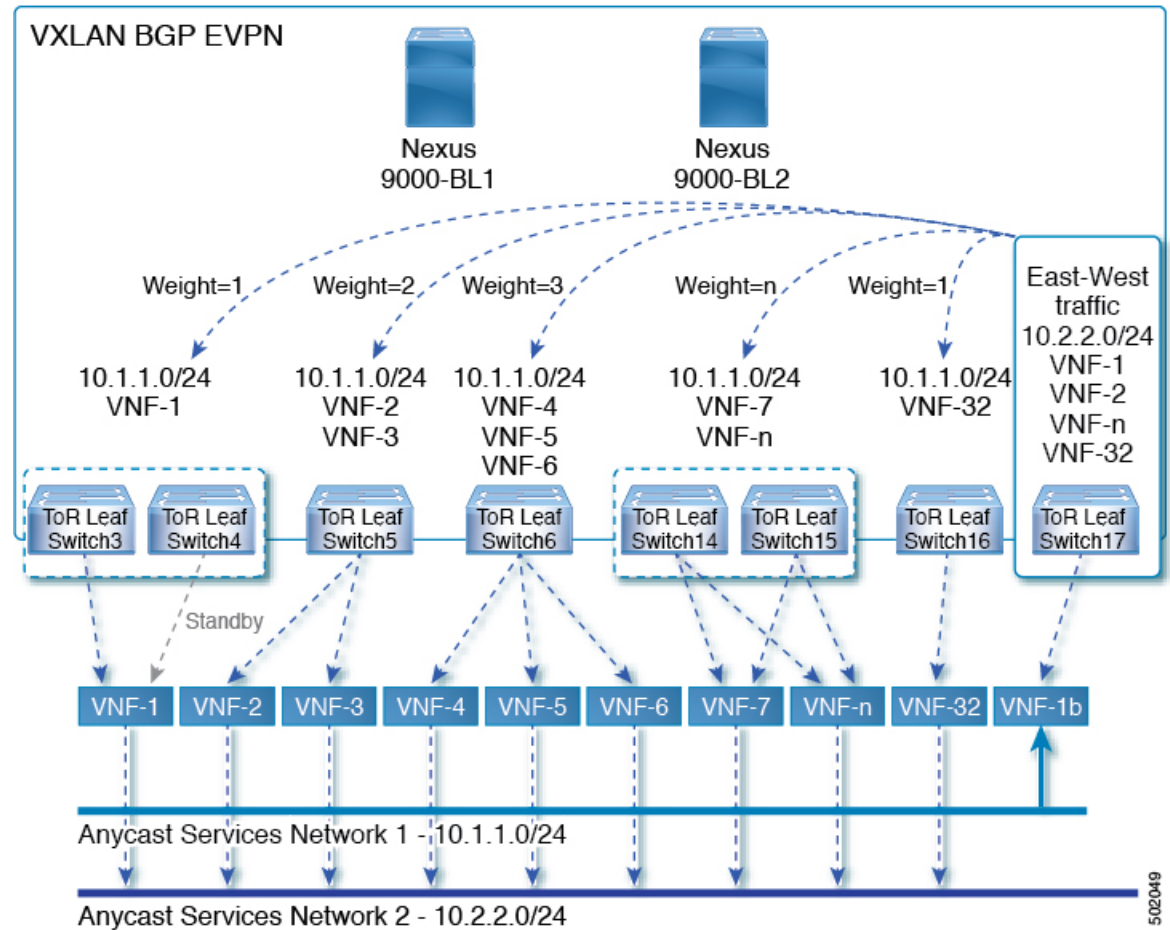
In Network Function Virtualization Infrastructures (NFVi), anycast services networks are advertised from multiple Virtual Network Functions (VNFs). The Proportional Multipath for VNF feature enables advertising of all the available next hops to a given destination network. This feature enables the switch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

Figure 55: Sample Topology (North-South Traffic)



In the preceding diagram, North-South traffic that enters the VXLAN fabric at a border leaf is sent across all egress endpoints with the traffic forwarded proportional to the number of links from the egress top of rack (ToR) to the destination network.

Figure 56: Sample Topology (East-West Traffic)



East-West traffic is forwarded between the VXLAN Tunnel Endpoints (VTEPs) proportional to the number of next hops advertised by each ToR switch to the destination network.

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

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

### Advertisement of Customer Networks

Customer networks are configured statically or learned locally by using an interior gateway protocol, (IGP) or external BGP (eBGP), over a Provider Edge(PE)-Customer Edge(CE) link. These networks are redistributed into BGP and advertised to the VXLAN fabric.

The networks advertised to the ToRs by the virtual machines (VMs) attached to them are advertised to the VXLAN fabric as EVPN Type-5 routes with the following:

- The route distinguisher (RD) will be the Layer 3 VNI's configured RD.
- The gateway IP field will be populated with the next hop.
- The next hop of the EVPN route will continue to be the VTEP IP.
- The export route targets of the routes will be derived from the configured export route targets of the associated Layer 3 VNI.

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

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

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

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

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

When the **`maximum-paths mixed`** command is enabled, BGP ignores the AS-path length, and URIB ignores the administrative distance when choosing ECMPs.

### Legacy Peer Support

Use the **`advertise-gw-ip`** command to advertise EVPN Type-5 routes with the gateway IP set. ToRs then advertise the gateway IP in the Type-5 NLRI. However, legacy peers running on NX-OS version older than Cisco NX-OS Release 9.2(1) can't process the gateway IP which might lead to unexpected behavior. To prevent this scenario from occurring, use the **`no advertise-gw-ip`** command to disable the Proportional Multipath for VNF feature for a legacy peer. BGP sets the gateway IP field of the Type-5 NLRI to zero even if the path being advertised has a valid gateway IP.

The **`no advertise-gw-ip`** command flaps the specified peer session as gracefully as possible. The remote peer triggers a graceful restart if the peer supports this capability. When the session is re-established, the local peer advertises EVPN Type-5 routes with the gateway IP set or with the gateway IP as zero depending on whether

the **advertise-gw-ip** command has been used. By default, this knob is enabled and the gateway IP field is populated with the appropriate next hop value.

## Guidelines and Limitations for Proportional Multipath for VNF

Proportional Multipath for VNF has the following guidelines and limitations:

- If the Proportional Multipath for VNF feature is enabled, maintenance mode isolation doesn't work because BGP installs all the paths in mixed multipath mode. Alternatively, a route-map is used to deny outbound BGP updates when a switch goes into maintenance mode by using user-defined profiles.
- This feature is supported for Cisco Nexus 9364C, 9300-EX, and 9300-FX/FX2 platform switches.
- Static and direct routes have to be redistributed into the BGP when the Proportional Multipath for VNF feature is enabled.
- If OSPF or EIGRP is being used as an IGP, routes can't be redistributed into BGP.
- If Proportional Multipath for VNF is enabled and routes aren't redistributed into BGP, asymmetric load balancing of traffic may occur as the local routes from URIB may not show up in BGP and on remote TORs as EVPN paths.
- Devices on which mixed-multipath is enabled must support the same load-balancing algorithm.
- If a VNF instance is multi-homed to multiple TORs, policies have to be configured or BGP routes have to be originated using a network command. As a result, each TOR connection to the VNF is displayed in the BGP routing table. Each TOR can now see the VNF's direct routes to the other TORs in which the VNF is multi-homed. Consequently, each TOR can advertise paths to the Gateway IPs through other TORs leading to a next hop resolution loop.

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

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

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

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

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

# Configuring the Route Reflector

## Procedure

|               | Command or Action                                                                                                                                                 | Purpose                                                                    |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------|
| <b>Step 1</b> | <b>configure terminal</b><br><br><b>Example:</b><br><code>switch# configure terminal</code>                                                                       | Enter global configuration mode.                                           |
| <b>Step 2</b> | <b>router bgp <i>number</i></b><br><br><b>Example:</b><br><code>switch(config)# router bgp 2</code>                                                               | Configure BGP.                                                             |
| <b>Step 3</b> | <b>address-family l2vpn evpn</b><br><br><b>Example:</b><br><code>switch(config-router)# address-family l2vpn evpn</code>                                          | Configure address family Layer 2 VPN EVPN under <b>router bgp</b> context. |
| <b>Step 4</b> | <b>additional-paths send</b><br><br><b>Example:</b><br><code>switch(config-router-af)# additional-paths send</code>                                               | The additional-paths configuration for sending..                           |
| <b>Step 5</b> | <b>additional-paths receive</b><br><br><b>Example:</b><br><code>switch(config-router-af)# additional-paths receive</code>                                         | The additional-paths configuration for receiving.                          |
| <b>Step 6</b> | <b>additional-paths selection route-map passall</b><br><br><b>Example:</b><br><code>switch(config-router-af)# additional-paths selection route-map passall</code> | The additional-paths configuration applied the route map.                  |
| <b>Step 7</b> | <b>route-map passall permit <i>seq-num</i></b><br><br><b>Example:</b><br><code>switch(config)# route-map passall permit 10</code>                                 | Configure the route map.                                                   |
| <b>Step 8</b> | <b>set path-selection all advertise</b><br><br><b>Example:</b><br><code>switch(config-route-map)# set path-selection all advertise</code>                         | Sets the route-map related to the additional-paths feature.                |

# Configuring the ToR

This procedure describes how to configure the ToR.

## Procedure

|               | Command or Action                                                                                                                                                                                                                                                                                                                                                                                                                                         | Purpose                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 1</b> | <b>configure terminal</b><br><b>Example:</b><br>switch# <b>configure terminal</b>                                                                                                                                                                                                                                                                                                                                                                         | Enter global configuration mode.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>Step 2</b> | <b>router bgp number</b><br><b>Example:</b><br>switch(config)# <b>router bgp 2</b>                                                                                                                                                                                                                                                                                                                                                                        | Configure BGP.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <b>Step 3</b> | <b>address-family l2vpn evpn</b><br><b>Example:</b><br>switch(config-router)# <b>address-family l2vpn evpn</b>                                                                                                                                                                                                                                                                                                                                            | Configure address family Layer 2 VPN EVPN under <b>router bgp</b> context.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>Step 4</b> | <b>maximum-paths eBGP max-paths  mixed mpath-count</b><br><b>Example:</b><br>switch(config-router-af)# maximum-paths ?<br><1-64> Number of parallel paths<br><br>*Default value is 1<br>eibgp Configure multipath for both EBGp and IBGP paths<br>ibgp Configure multipath for IBGP paths<br>local Configure multipath for local paths<br>mixed Configure multipath for local and remote paths<br>switch(config-router-af)# <b>maximum-paths mixed 32</b> | <ul style="list-style-type: none"> <li>• <i>eBGP max-path</i>—Enables the eBGP maximum paths. The range is from 1 to 64 parallel paths. The default value is 1.</li> <li>• Enables BGP and the Unicast Routing Information Base (URIB) to consider the following paths as Equal Cost Multi Path (ECMP): <ul style="list-style-type: none"> <li>• eBGP paths</li> <li>• eiBGP paths</li> <li>• iBGP paths</li> <li>• Paths from other protocols (such as static) that are redistributed or injected into BGP</li> </ul> </li> <li>•</li> <li>• <b>local</b>—Enables the multipath for local paths.</li> <li>•</li> </ul> |
| <b>Step 5</b> | <b>additional-paths send</b><br><b>Example:</b><br>switch(config-router-af)# <b>additional-paths send</b>                                                                                                                                                                                                                                                                                                                                                 | The additional-paths configuration for sending.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

|                | Command or Action                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Purpose                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 6</b>  | <b>additional-paths receive</b><br><b>Example:</b><br><pre>switch(config-router-af) # additional-paths receive</pre>                                                                                                                                                                                                                                                                                                                                                                                     | The additional-paths configuration for receiving.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>Step 7</b>  | <b>additional-paths selection route-map passall</b><br><b>Example:</b><br><pre>switch(config-router-af) # additional-paths selection route-map passall</pre>                                                                                                                                                                                                                                                                                                                                             | The additional-paths configuration applied the route map.                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <b>Step 8</b>  | <b>exit</b><br><b>Example:</b><br><pre>switch(config-router-af) # exit</pre>                                                                                                                                                                                                                                                                                                                                                                                                                             | Exits command mode.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>Step 9</b>  | <b>vrf evpn-tenant-1001</b><br><b>Example:</b><br><pre>switch(config-router) # vrf evpn-tenant-1001</pre>                                                                                                                                                                                                                                                                                                                                                                                                | Switch to the VRF configuration mode.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>Step 10</b> | <b>address-family ipv4 unicast</b><br><b>Example:</b><br><pre>switch(config-router) # address-family ipv4 unicast</pre>                                                                                                                                                                                                                                                                                                                                                                                  | Configure address family for IPv4.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>Step 11</b> | <b>export-gateway-ip</b><br><b>Example:</b><br><pre>switch(config-router-vrf-af) # export-gateway-ip</pre>                                                                                                                                                                                                                                                                                                                                                                                               | Enables BGP to advertise the gateway IP in the EVPN Type-5 routes.                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>Step 12</b> | <b>maximum-paths eBGP max-paths  mixed mpath-count</b><br><b>Example:</b><br><pre>switch(config-router-vrf-af) # maximum-paths ?     &lt;1-64&gt;  Number of parallel paths                  *Default value is 1     eibgp    Configure multipath for both EBGP and IBGP paths     ibgp     Configure multipath for IBGP paths     local    Configure multipath for local paths     mixed    Configure multipath for local and remote paths  switch(config-router-vrf-af) # maximum-paths mixed 32</pre> | <ul style="list-style-type: none"> <li>• <i>eBGP max-path</i>—Enables the eBGP maximum paths. The range is from 1 to 64 parallel paths. The default value is 1.</li> <li>• Enables BGP and the Unicast Routing Information Base (URIB) to consider the following paths as Equal Cost Multi Path (ECMP): <ul style="list-style-type: none"> <li>• eBGP paths</li> <li>• eiBGP paths</li> <li>• iBGP paths</li> <li>• Paths from other protocols (such as static) that are redistributed or injected into BGP</li> </ul> </li> </ul> |



|                | Command or Action                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Purpose                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | <ul style="list-style-type: none"> <li>• <b>local</b>—Enables the multipath for local paths.</li> <li>•</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>Step 13</b> | <b>redistribute static route-map redist-rtmap</b><br><b>Example:</b><br><pre>switch(config-router-vrf-af) # redistribute static route-map redist-rtmap</pre>                                                                                                                                                                                                                                                                                                                                             | Preserves the next-hop of the redistributed paths.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>Step 14</b> | <b>exit</b><br><b>Example:</b><br><pre>switch(config-router-vrf-af) # exit</pre>                                                                                                                                                                                                                                                                                                                                                                                                                         | Exits command mode.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>Step 15</b> | <b>address-family ipv6 unicast</b><br><b>Example:</b><br><pre>switch(config-router-vrf) # address-family ipv6 unicast</pre>                                                                                                                                                                                                                                                                                                                                                                              | Configure address family for IPv6.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>Step 16</b> | <b>export-gateway-ip</b><br><b>Example:</b><br><pre>switch(config-router-vrf-af) # export-gateway-ip</pre>                                                                                                                                                                                                                                                                                                                                                                                               | Enables BGP to advertise the gateway IP in the EVPN Type-5 routes.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>Step 17</b> | <b>maximum-paths eBGP max-paths  mixed mpath-count</b><br><b>Example:</b><br><pre>switch(config-router-vrf-af) # maximum-paths ?     &lt;1-64&gt;  Number of parallel paths                  *Default value is 1     eibgp    Configure multipath for both EBGP and IBGP paths     ibgp     Configure multipath for IBGP paths     local    Configure multipath for local paths     mixed    Configure multipath for local and remote paths  switch(config-router-vrf-af) # maximum-paths mixed 32</pre> | <ul style="list-style-type: none"> <li>• <i>eBGP max-path</i>—Enables the eBGP maximum paths. The range is from 1 to 64 parallel paths. The default value is 1.</li> <li>• Enables BGP and the Unicast Routing Information Base (URIB) to consider the following paths as Equal Cost Multi Path (ECMP): <ul style="list-style-type: none"> <li>• eBGP paths</li> <li>• eiBGP paths</li> <li>• iBGP paths</li> <li>• Paths from other protocols (such as static) that are redistributed or injected into BGP</li> </ul> </li> <li>•</li> <li>• <b>local</b>—Enables the multipath for local paths.</li> <li>•</li> </ul> |

|                | Command or Action                                                                                                                                                          | Purpose                                                     |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------|
| <b>Step 18</b> | <b>redistribute static route-map redist-rtmap</b><br><br><b>Example:</b><br><code>switch(config-router-vrf-af) #<br/>redistribute static route-map<br/>redist-rtmap</code> | Preserves the next-hop of the redistributed paths.          |
| <b>Step 19</b> | <b>exit</b><br><br><b>Example:</b><br><code>switch(config-router-vrf-af) # exit</code>                                                                                     | Exits command mode.                                         |
| <b>Step 20</b> | <b>route-map passall permit seq-num</b><br><br><b>Example:</b><br><code>switch(config) # route-map passall permit<br/>10</code>                                            | Configure the route map.                                    |
| <b>Step 21</b> | <b>set path-selection all advertise</b><br><br><b>Example:</b><br><code>switch(config-route-map) # set<br/>path-selection all advertise</code>                             | Sets the route-map related to the additional-paths feature. |

## Configuring the Border Leaf

This procedure describes how to configure the border leaf.

### Procedure

|               | Command or Action                                                                                                             | Purpose                                                                                                                                                                    |
|---------------|-------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 1</b> | <b>configure terminal</b><br><br><b>Example:</b><br><code>switch# configure terminal</code>                                   | Enter global configuration mode.                                                                                                                                           |
| <b>Step 2</b> | <b>router bgp number</b><br><br><b>Example:</b><br><code>switch(config) # router bgp 2</code>                                 | Configure BGP.                                                                                                                                                             |
| <b>Step 3</b> | <b>address-family l2vpn evpn</b><br><br><b>Example:</b><br><code>switch(config-router) # address-family<br/>l2vpn evpn</code> | Configure address family Layer 2 VPN EVPN under <b>router bgp</b> context.                                                                                                 |
| <b>Step 4</b> | <b>maximum-paths eBGP max-paths  mixed mpath-count</b><br><br><b>Example:</b>                                                 | <ul style="list-style-type: none"> <li>• <i>eBGP max-path</i>—Enables the eBGP maximum paths. The range is from 1 to 64 parallel paths. The default value is 1.</li> </ul> |

|                | Command or Action                                                                                                                                                                                                                                                                                                                                                                                                                          | Purpose                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | <pre> switch(config-router-af)# maximum-paths ?   &lt;1-64&gt;  Number of parallel paths                *Default value is 1   eibgp   Configure multipath for both           EBGP and IBGP paths   ibgp    Configure multipath for IBGP           paths   local   Configure multipath for local           paths   mixed   Configure multipath for local           and remote paths switch(config-router-af)# maximum-paths mixed 32 </pre> | <ul style="list-style-type: none"> <li>Enables BGP and the Unicast Routing Information Base (URIB) to consider the following paths as Equal Cost Multi Path (ECMP): <ul style="list-style-type: none"> <li>eBGP paths</li> <li>eiBGP paths</li> <li>iBGP paths</li> <li>Paths from other protocols (such as static) that are redistributed or injected into BGP</li> </ul> </li> <li>•</li> <li>• <b>local</b>—Enables the multipath for local paths.</li> <li>•</li> </ul> |
| <b>Step 5</b>  | <b>additional-paths send</b><br><b>Example:</b><br><pre> switch(config-router-af)# additional-paths send </pre>                                                                                                                                                                                                                                                                                                                            | The additional-paths configuration for sending.                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>Step 6</b>  | <b>additional-paths receive</b><br><b>Example:</b><br><pre> switch(config-router-af)# additional-paths receive </pre>                                                                                                                                                                                                                                                                                                                      | The additional-paths configuration for receiving.                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>Step 7</b>  | <b>additional-paths selection route-map passall</b><br><b>Example:</b><br><pre> switch(config-router-af)# additional-paths selection route-map passall </pre>                                                                                                                                                                                                                                                                              | The additional-paths configuration enables the additional-paths feature.                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>Step 8</b>  | <b>exit</b><br><b>Example:</b><br><pre> switch(config-router-af)# exit </pre>                                                                                                                                                                                                                                                                                                                                                              | Exits command mode.                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>Step 9</b>  | <b>vrf evpn-tenant-1001</b><br><b>Example:</b><br><pre> switch(config-router)# vrf evpn-tenant-1001 </pre>                                                                                                                                                                                                                                                                                                                                 | Switch to the VRF configuration mode.                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>Step 10</b> | <b>address-family ipv4 unicast</b><br><b>Example:</b><br><pre> switch(config-router)# address-family ipv4 unicast </pre>                                                                                                                                                                                                                                                                                                                   | Configure address family for IPv4.                                                                                                                                                                                                                                                                                                                                                                                                                                          |

|                | Command or Action                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Purpose                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Step 11</b> | <b>export-gateway-ip</b><br><b>Example:</b><br><pre>switch(config-router-vrf-af) # export-gateway-ip</pre>                                                                                                                                                                                                                                                                                                                                                                | Enables BGP to advertise the gateway IP in the EVPN Type-5 routes.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>Step 12</b> | <b>maximum-paths eBGP max-paths  mixed mpath-count</b><br><b>Example:</b><br><pre>switch(config-router-af) # maximum-paths ? &lt;1-64&gt; Number of parallel paths            *Default value is 1 eibgp    Configure multipath for both EBGP and IBGP paths ibgp     Configure multipath for IBGP paths local    Configure multipath for local paths mixed    Configure multipath for local and remote paths  switch(config-router-vrf-af) # maximum-paths mixed 32</pre> | <ul style="list-style-type: none"> <li>• <i>eBGP max-path</i>—Enables the eBGP maximum paths. The range is from 1 to 64 parallel paths. The default value is 1.</li> <li>• Enables BGP and the Unicast Routing Information Base (URIB) to consider the following paths as Equal Cost Multi Path (ECMP): <ul style="list-style-type: none"> <li>• eBGP paths</li> <li>• eiBGP paths</li> <li>• iBGP paths</li> <li>• Paths from other protocols (such as static) that are redistributed or injected into BGP</li> </ul> </li> <li>• <b>local</b>—Enables the multipath for local paths.</li> </ul> |
| <b>Step 13</b> | <b>redistribute static route-map redist-rtmap</b><br><b>Example:</b><br><pre>switch(config-router-vrf-af) # redistribute static route-map redist-rtmap</pre>                                                                                                                                                                                                                                                                                                              | Preserves the next-hop of the redistributed paths.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>Step 14</b> | <b>address-family ipv6 unicast</b><br><b>Example:</b><br><pre>switch(config-router-vrf) # address-family ipv6 unicast</pre>                                                                                                                                                                                                                                                                                                                                               | Configure address family for IPv6.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>Step 15</b> | <b>export-gateway-ip</b><br><b>Example:</b><br><pre>switch(config-router-vrf-af) # export-gateway-ip</pre>                                                                                                                                                                                                                                                                                                                                                                | Enables BGP to advertise the gateway IP in the EVPN Type-5 routes.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>Step 16</b> | <b>maximum-paths eBGP max-paths  mixed mpath-count</b><br><b>Example:</b>                                                                                                                                                                                                                                                                                                                                                                                                 | <ul style="list-style-type: none"> <li>• <i>eBGP max-path</i>—Enables the eBGP maximum paths. The range is from 1 to 64 parallel paths. The default value is 1.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                        |

|                | Command or Action                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Purpose                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | <pre>switch(config-router-vrf-af) # maximum-paths ?     &lt;1-64&gt;  Number of parallel paths                  *Default value is 1     eibgp    Configure multipath for both             EBGP and IBGP paths     ibgp     Configure multipath for IBGP             paths     local    Configure multipath for local             paths     mixed    Configure multipath for local             and remote paths  switch(config-router-vrf-af) # <b>maximum-paths mixed 32</b></pre> | <ul style="list-style-type: none"> <li>Enables BGP and the Unicast Routing Information Base (URIB) to consider the following paths as Equal Cost Multi Path (ECMP): <ul style="list-style-type: none"> <li>eBGP paths</li> <li>eiBGP paths</li> <li>iBGP paths</li> <li>Paths from other protocols (such as static) that are redistributed or injected into BGP</li> </ul> </li> <li> <ul style="list-style-type: none"> <li><b>local</b>—Enables the multipath for local paths.</li> </ul> </li> </ul> |
| <b>Step 17</b> | <b>redistribute static route-map redistrib-rtmap</b><br><b>Example:</b><br><pre>switch(config-router-vrf-af) # <b>redistribute static route-map</b> <b>redistrib-rtmap</b></pre>                                                                                                                                                                                                                                                                                                   | Preserves the next-hop of the redistributed paths.                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>Step 18</b> | <b>exit</b><br><b>Example:</b><br><pre>switch(config-router-vrf-af) # <b>exit</b></pre>                                                                                                                                                                                                                                                                                                                                                                                            | Exits command mode.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>Step 19</b> | <b>route-map passall permit seq-num</b><br><b>Example:</b><br><pre>switch(config) # <b>route-map passall permit</b> <b>10</b></pre>                                                                                                                                                                                                                                                                                                                                                | Configure the route map.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>Step 20</b> | <b>set path-selection all advertise</b><br><b>Example:</b><br><pre>switch(config-route-map) # <b>set</b> <b>path-selection all advertise</b></pre>                                                                                                                                                                                                                                                                                                                                 | Sets the route-map related to the additional-paths feature.                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>Step 21</b> | <b>ip load-sharing address source-destination rotate rotate universal-id seed</b><br><b>Example:</b><br><pre>ip load-sharing address source-destination rotate 32 universal-id 1</pre>                                                                                                                                                                                                                                                                                             | <p>Configures the unicast FIB load-sharing algorithm for data traffic.</p> <ul style="list-style-type: none"> <li>The <b>universal-id</b> option sets the random seed for the hash algorithm and shifts the flow from one link to another.</li> </ul> <p>You do not need to configure the universal ID. Cisco NX-OS chooses the Universal ID if you</p>                                                                                                                                                 |

|  | Command or Action | Purpose                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|--|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  |                   | <p>do not configure it. The <i>seed</i> range is from 1 to 4294967295.</p> <ul style="list-style-type: none"> <li>The <b>rotate</b> option causes the hash algorithm to rotate the link picking selection so that it does not continually choose the same link across all nodes in the network. It does so by influencing the bit pattern for the hash algorithm. This option shifts the flow from one link to another and load balances the already load-balanced (polarized) traffic from the first ECMP level across multiple links.</li> </ul> <p>If you specify a <b>rotate</b> value, the 64-bit stream is interpreted starting from that bit position in a cyclic rotation. The <b>rotate</b> range is from 1 to 63, and the default is 32.</p> <p><b>Note</b> With multi-tier Layer 3 topology, polarization is possible. To avoid polarization, use a different rotate bit at each tier of the topology.</p> <p><b>Note</b> To configure a rotation value for port channels, use the <b>port-channel load-balance src-dst ip-l4port rotate</b> <i>rotate</i> command. For more information on this command, see the <a href="#">Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide, Release 9.x</a>.</p> |

## Configuring the BGP Legacy Peer

If you are running a Cisco Nexus Release prior to 9.2(1), follow this procedure to disable sending the gateway IP address to that peer.

### Procedure

|               | Command or Action                                                                         | Purpose                          |
|---------------|-------------------------------------------------------------------------------------------|----------------------------------|
| <b>Step 1</b> | <b>configure terminal</b><br><br><b>Example:</b><br><pre>switch# configure terminal</pre> | Enter global configuration mode. |

|               | Command or Action                                                                                                                                               | Purpose                                                                                        |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|
| <b>Step 2</b> | <b>router bgp</b> <i>number</i><br><b>Example:</b><br>switch(config) # <b>router bgp 2000000</b>                                                                | Configure BGP.                                                                                 |
| <b>Step 3</b> | <b>neighbor</b> <i>address</i> <b>remote-as</b> <i>number</i><br><b>Example:</b><br>switch(config-router) # <b>neighbor 8.8.8.8</b><br><b>remote-as 2000000</b> | Define neighbor.                                                                               |
| <b>Step 4</b> | <b>address-family</b> <b>l2vpn evpn</b><br><b>Example:</b><br>switch(config-router-neighbor) #<br><b>address-family l2vpn evpn</b>                              | Configure address family Layer 2 VPN EVPN.                                                     |
| <b>Step 5</b> | <b>no advertise-gw-ip</b><br><b>Example:</b><br>switch(config-router-neighbor-af) # <b>no</b><br><b>advertise-gw-ip</b>                                         | Disables the BGP EVPN Mixed-path and Proportional Layer-3 Multipath feature for a legacy peer. |

## Configuring a User-Defined Profile for Maintenance Mode

### Procedure

|               | Command or Action                                                                                                                                                      | Purpose                                                                              |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| <b>Step 1</b> | <b>configure terminal</b><br><b>Example:</b><br>switch# <b>configure terminal</b>                                                                                      | Enter global configuration mode.                                                     |
| <b>Step 2</b> | <b>configure maintenance profile</b><br><b>maintenance-mode</b><br><b>Example:</b><br>switch(config) # <b>configure maintenance</b><br><b>profile maintenance-mode</b> | Configure maintenance mode profile.                                                  |
| <b>Step 3</b> | <b>route-map</b> <i>name</i> <b>deny</b> <i>sequence</i><br><b>Example:</b><br>switch(config-mm-profile) # <b>route-map GIR</b><br><b>deny 5</b>                       | Configure route map. The value of <i>sequence</i> is from 0 to 65535. Default is 10. |

## Configuring a User-Defined Profile for Normal Mode

### Procedure

|               | Command or Action                                                                                                                                 | Purpose                                                                              |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| <b>Step 1</b> | <b>configure terminal</b><br><br><b>Example:</b><br><code>switch# configure terminal</code>                                                       | Enter global configuration mode.                                                     |
| <b>Step 2</b> | <b>configure maintenance profile normal-mode</b><br><br><b>Example:</b><br><code>switch(config)# configure maintenance profile normal-mode</code> | Configure maintenance mode.                                                          |
| <b>Step 3</b> | <b>route-map name permit sequence</b><br><br><b>Example:</b><br><code>switch(config-mm-profile)# route-map GIR permit 5</code>                    | Configure route map. The value of <i>sequence</i> is from 0 to 65535. Default is 10. |

## Configuring a Default Route Map

### Procedure

|               | Command or Action                                                                                                              | Purpose                                                                              |
|---------------|--------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| <b>Step 1</b> | <b>configure terminal</b><br><br><b>Example:</b><br><code>switch# configure terminal</code>                                    | Enter global configuration mode.                                                     |
| <b>Step 2</b> | <b>route-map name permit sequence</b><br><br><b>Example:</b><br><code>switch(config-mm-profile)# route-map GIR permit 5</code> | Configure route map. The value of <i>sequence</i> is from 0 to 65535. Default is 10. |

## Applying a Route Map to a Route Reflector

### Procedure

|               | Command or Action                                | Purpose                          |
|---------------|--------------------------------------------------|----------------------------------|
| <b>Step 1</b> | <b>configure terminal</b><br><br><b>Example:</b> | Enter global configuration mode. |



|               | Command or Action                                                                                                                        | Purpose                                                                                                                              |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------|
|               | <code>switch# configure terminal</code>                                                                                                  |                                                                                                                                      |
| <b>Step 2</b> | <b>router <i>bgp number</i></b><br><br><b>Example:</b><br><code>switch(config)# router bgp 2</code>                                      | Configure BGP.                                                                                                                       |
| <b>Step 3</b> | <b>neighbor <i>ip-address</i></b><br><br><b>Example:</b><br><code>switch(config-router)# neighbor 10.1.1.1</code>                        | Configure the IP address of a BGP neighbor which is the route reflector. <i>ip-address</i> can be an IPv4 or IPv6 address or prefix. |
| <b>Step 4</b> | <b>address-family <i>l2vpn evpn</i></b><br><br><b>Example:</b><br><code>switch(config-router-neighbor)# address-family l2vpn evpn</code> | Configure a Layer 2 VPN EVPN address family.                                                                                         |
| <b>Step 5</b> | <b>route-map <i>name out</i></b><br><br><b>Example:</b><br><code>switch(config-router-neighbor-af)# route-map GIR out</code>             | Apply the route map to the neighbor route reflector.                                                                                 |

## Verifying Proportional Multipath for VNF

| Command                                                | Purpose                                                                                                                          |
|--------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------|
| <code>show bgp ipv4 unicast</code>                     | Displays Border Gateway Protocol (BGP) information for the IPv4 unicast address family.                                          |
| <code>show bgp l2vpn evpn</code>                       | Displays BGP information for the Layer-2 Virtual Private Network (L2VPN) Ethernet Virtual Private Network (EVPN) address family. |
| <code>show ip route</code>                             | Displays routes from the unicast RIB.                                                                                            |
| <code>show maintenance profile maintenance-mode</code> | Displays the GIR user-defined profile for the maintenance mode.                                                                  |
| <code>show maintenance profile normal-mode</code>      | Displays the GIR user-defined profile for the normal mode.                                                                       |

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

```
switch# show bgp l2vpn evpn 11.1.1.0
BGP routing table information for VRF default, address family L2VPN EVPN
Route Distinguisher: 13.13.13.13:3 // Remote route
BGP routing table entry for [5]:[0]:[0]:[24]:[11.1.1.0]/224, version 1341
Paths: (3 available, best #1)
```

Flags: (0x000002) on xmit-list, is not in l2rib/evpn, is not in HW  
 Multipath: eBGP

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

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

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

Path-id 1 not advertised to any peer

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

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

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

```

Received label 22001
Received path-id 2
Extcommunity: RT:23456:22001 Route-Import:33.33.33.33:2001 ENCAP:8
Router MAC:e00e.da4a.589d

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

Path-id 1 not advertised to any peer

```

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

```

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

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

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

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

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

```

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

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

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

The following example shows how to display the GIR user-defined profile for the maintenance mode:

```
switch# show maintenance profile maintenance-mode
[Maintenance Mode]
ip pim isolate
router bgp 2
 isolate
router isis 1
 isolate
route-map GIR deny 5
```

The following example shows how to display the GIR user-defined profile for the normal mode:

```
switch# show maintenance profile normal-mode
[Normal Mode]
no ip pim isolate
router bgp 2
 no isolate
router isis 1
 no isolate
route-map GIR permit 5
```



## INDEX

### A

action forward [236, 241](#)  
 address-family ipv4 unicast [71, 76, 128](#)  
 address-family ipv6 unicast [76](#)  
 address-family l2vpn evpn [76–78](#)  
 advertise [76](#)

### C

class [254](#)  
 class-map [253](#)  
 configure maintenance profile maintenance-mode [355](#)  
 configure maintenance profile normal-mode [356](#)

### F

fabric forwarding mode anycast-gateway [238, 242](#)  
 feature nv overlay [30, 70](#)  
 feature vn-segment [70](#)  
 feature vn-segment-vlan-based [29](#)

### H

hardware access-list team region arp-ether double-wide [17, 79](#)  
 hardware access-list team region egr-racl 256 [241](#)  
 hardware access-list team region ing-ifacl 256 [234, 237](#)  
 hardware access-list team region vacl 256 [239–240](#)  
 host-reachability protocol bgp [73, 75](#)

### I

ingress-replication protocol bgp [30, 75](#)  
 ingress-replication protocol static [31](#)  
 interface [73](#)  
 interface ethernet [234, 237](#)  
 interface nve [26, 30, 254](#)  
 interface nve 1 [79](#)  
 interface vlan [70, 242](#)  
 ip access-group [237, 242](#)  
 ip access-list [234–235, 237, 239–241](#)  
 ip address [73, 237, 242](#)  
 ip port access-group [235](#)  
 ip route 0.0.0.0/0 [128](#)

### M

mac address-table static [29](#)  
 match [254](#)  
 match ip address [236, 239](#)  
 mcast-group [26, 74](#)  
 member vni [26, 30–31, 74–75, 79](#)

### N

neighbor [76–78](#)  
 no feature nv overlay [80](#)  
 no feature vn-segment-vlan-based [80](#)  
 no ip redirects [237, 242](#)  
 no ipv6 redirects [238, 242](#)  
 no nv overlay evpn [80](#)  
 no shutdown [235, 237, 242](#)  
 nv overlay evpn [70](#)

### P

peer-ip [31](#)  
 permit [239–240](#)  
 permit ip [234–235, 237, 239–241](#)  
 policy-map type qos [254](#)

### Q

qos-mode [254](#)

### R

rd auto [71, 128](#)  
 retain route-target all [77–78](#)  
 route-map [356](#)  
 route-map permitall out [77](#)  
 route-target both [128](#)  
 route-target both auto [71, 128](#)  
 route-target both auto evpn [72](#)  
 router bgp [75, 77–78](#)  
 router-id [75](#)

**S**

- send-community extended [76–78](#)
- service-policy type qos input [254](#)
- set qos-group [254](#)
- show bgp l2vpn evpn [82](#)
- show interface [217](#)
- show ip arp suppression-cache [82](#)
- show l2route evpn fl all [82](#)
- show l2route evpn imet all [82](#)
- show l2route evpn mac [82](#)
- show l2route evpn mac-ip all [83](#)
- show l2route evpn mac-ip all detail [83](#)
- show l2route topology [83](#)
- show mac address-table static interface nve [29](#)
- show nve vrf [82](#)
- show vxlan interface [82](#)
- show vxlan interface | count [82](#)
- source-interface [26, 30](#)
- source-interface config [16](#)
- source-interface hold-down-time [16](#)
- spanning-tree bpdupfilter enable [198](#)

- statistics per-entry [239–240](#)
- suppress-arp [79](#)
- suppress-arp disable [79](#)
- switchport [235](#)
- switchport access vlan [198](#)
- switchport mode dot1q-tunnel [198](#)
- switchport mode trunk [217, 235](#)
- switchport trunk allowed vlan [235](#)
- switchport vlan mapping [217](#)
- switchport vlan mapping enable [217](#)

**V**

- vlan [25, 70, 72–73](#)
- vlan access-map [236, 239, 241](#)
- vn-segment [25, 70](#)
- vn-segment-vlan-based [70](#)
- vni [71, 128](#)
- vrf [76](#)
- vrf context [71, 128](#)
- vrf member [73, 237, 242](#)