Inspur

CN12900 Series

INOS-CN VXLAN Configuration Guide

Inspur-Cisco Networking Technology Co.,Ltd. provides customers with comprehensive technical support and services. For any assistance, please contact our local office or company headquarters.
Website: http://www.inspur.com/
Technical Support Tel: 400-691-1766
Technical Support Email:icnt_service@inspur.com
Technical Document Support Email: icnt_service@inspur.com
Address: 1036 Langchao Road, Lixia District, Jinan City, Shandong Province
Postal code: 250101

Notice

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute the warranty of any kind, express or implied

Preface

Objectives

This guide describes main functions of the CN12900 Series. To have a quick grasp of the CN12900 Series, please read this manual carefully.

Versions

The following table lists the product versions related to this document.

Product name	Version
CN12900 Series	

Conventions

Symbol conventions

The symbols that may be found in this document are defined as follows.

Symbol	Description
⚠ Warning	Indicates a hazard with a medium or low level of risk which, if not avoided, could result in minor or moderate injury.
⚠ Caution	Indicates a potentially hazardous situation that, if not avoided, could cause equipment damage, data loss, and performance degradation, or unexpected results.
✎ Note	Provides additional information to emphasize or supplement important points of the main text.
🔍 Tip	Indicates a tip that may help you solve a problem or save time.

General conventions

Convention	Description
Boldface	Names of files, directories, folders, and users are in **boldface**. For example, log in as user **root**.

Convention	Description
Italic	Book titles are in *italics*.
`Lucida Console`	Terminal display is in `Lucida Console`.

Command conventions

Convention	Description
Boldface	The keywords of a command line are in **boldface**.
Italic	Command arguments are in *italics*.
[]	Items (keywords or arguments) in square brackets [ ] are optional.
{ x | y | ... }	Alternative items are grouped in braces and separated by vertical bars. One is selected.
[x | y | ...]	Optional alternative items are grouped in square brackets and separated by vertical bars. One or none is selected.
{ x | y | ... } *	Alternative items are grouped in braces and separated by vertical bars. A minimum of one or a maximum of all can be selected.
[x | y | ...] *	The parameter before the & sign can be repeated 1 to n times.

GUI conventions

Convention	Description
Boldface	Buttons, menus, parameters, tabs, windows, and dialog titles are in **boldface**. For example, click **OK**.
>	Multi-level menus are in boldface and separated by the ">" signs. For example, choose **File** > **Create** > **Folder**.

Keyboard operation

Format	Description
Key	Press the key. For example, press **Enter** and press **Tab**.
Key 1+Key 2	Press the keys concurrently. For example, pressing **Ctrl+C** means the two keys should be pressed concurrently.
Key 1, Key 2	Press the keys in turn. For example, pressing **Alt**, **A** means the two keys should be pressed in turn.

Mouse operation

Action	Description
Click	Select and release the primary mouse button without moving the pointer.
Double-click	Press the primary mouse button twice continuously and quickly without moving the pointer.
Drag	Press and hold the primary mouse button and move the pointer to a certain position.

Change history

Updates between document versions are cumulative. Therefore, the latest document version contains all updates made to previous versions.

Issue 01 (2020-02-24)

Initial commercial release

Contents

Figure

Table

Preface

This preface includes the following sections:

Audience

This publication is for network administrators who install, configure, and maintain Inspur CN12904 and CN12908 switches.

Document Conventions

Command descriptions use the following conventions:

Convention	Description
bold	Bold text indicates the commands and keywords that you enter literally as shown.
Italic	Italic text indicates arguments for which the user supplies the values.
[x]	Square brackets enclose an optional element (keyword or argument).
[x | y]	Square brackets enclosing keywords or arguments separated by a vertical bar indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments separated by a vertical bar indicate a required choice.
[x {y | z}]	Nested set of square brackets or braces indicate optional or required choices within optional or required elements. Braces and a vertical bar within square brackets indicate a required choice within an optional element.
`variable`	Indicates a variable for which you supply values, in context where italics cannot be used.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.
`screen font`	Terminal sessions and information the switch displays are in screen font.
`boldface screen font`	Information you must enter is in boldface screen font.
italic screen font	Arguments for which you supply values are in italic screen font.
< >	Nonprinting characters, such as passwords, are in angle brackets.
[ ]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line

Convention	Description
	of code indicates a comment line.

CHAPTER 1 New and Changed Information

This chapter provides release-specific information for each new and changed feature in the *Inspur CN12900 Series INOS-CN VXLAN Configuration Guide*.

1.1 New and Changed Information

This table summarizes the new and changed features for the *Inspur CN12900 Series INOS-CN VXLAN Configuration Guide* and where they are documented.

Table 1： New and Changed Features

Feature	**Description**	**Changed in Release**	**Where Documented**
DHCP IPV4 Relay	Support added for DHCP IPv4	9.2(2)	DHCP_Relay_in_VXLAN_BGP_EVPN
Initial release		9.2(1i)	

CHAPTER 2 Overview

2.1 VXLAN Overview

Inspur CN12900 Series switches are designed for hardware-based VXLAN function. It provides Layer 2 connectivity extension across the Layer 3 boundary and integrates between VXLAN and non-VXLAN infrastructures. This can enable virtualized and multitenant data center designs over a shared common physical infrastructure.

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

When deployed as a VXLAN gateway, Inspur CN12900 Series switches can connect VXLAN and classic VLAN segments to create a common forwarding domain so that tenant devices can reside in both environments.

VXLAN has the following benefits:

- Flexible placement of multitenant segments throughout the data center.

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

- Higher scalability to address more Layer 2 segments.

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

- Utilization of available network paths in the underlying infrastructure.

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

2.2 VXLAN Encapsulation and Packet Format

VXLAN is a Layer 2 overlay scheme over a Layer 3 network. It uses MAC Address-in-User Datagram Protocol (MAC-in-UDP) encapsulation to provide a means to extend Layer 2 segments across the data center network.

VXLAN is a solution to support a flexible, large-scale multitenant environment over a shared common physical infrastructure. The transport protocol over the physical data center network is IP plus UDP.

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

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

2.3 VXLAN Tunnel Endpoint

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

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

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

2.4 VXLAN Packet Forwarding Flow

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

2.5 Inspur CN12900 as Hardware-Based VXLAN Gateway

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

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

An Inspur CN12900 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 Inspur CN12900 Series eliminates the need for an additional physical or virtual

device to be the gateway. The hardware-based encapsulation and de-encapsulation provides line-rate performance for all frame sizes.

2.6 vPC Consistency Check for vPC VTEPs

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

Parameter	vPC Check Type	Description
VLAN-VNI mapping	Type-1-nongraceful	Brings down the affected VLANs on vPC ports on both sides.
VTEP-Member-VNI	Type-1-nongraceful	Member VNIs must be the same on both nodes. VNIs that are not common bring down the corresponding VLANs on vPC ports on both sides. (The attributes considered are mcast group address, suppress-arp, and Layer 3 VRF VNI.)
VTEP-emulated IP	Type-1-nongraceful	If an emulated IP address is not the same on both nodes, all gateway vPC ports on one side (secondary) are brought down. Alternatively, one side of all vPC ports is brought down.

Parameter	vPC Check Type	Description
		The VTEP source loopback on the vPC secondary is also brought down if the emulated IP address is not the same on both sides.
NVE Oper State	Type-1-nongraceful	The NVE needs to be in the oper UP state on both sides for the vPC consistency check. If both VTEPs are not in the OPER_UP state, the secondary leg is brought down along with the VTEP source loopback on the vPC secondary.
NVE Host-Reachability Protocol	Type-1-nongraceful	The vPC on both sides must be configured with the same host-reachability protocol. Otherwise, the secondary leg is brought down along with the VTEP source loopback on the vPC secondary.

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

The following situations are detected as inconsistencies:

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

The following is an example of displaying vPC information:

```
           switch# sh vpc consistency-parameters global

               Legend:
                   Type 1 : vPC will be suspended in case of mismatch

Name                        Typ   Local Value             Peer Value
                            e
-------------               ---   ---------------------   ----------------------
                            -     -                       -
Vlan to Vn-segment Map      1     1024 Relevant Map(s)    1024 Relevant Map(s)
STP Mode                    1     MST                     MST
STP Disabled                1     None                    None
STP MST Region Name         1     ""                      ""
STP MST Region Revision     1     0                       0
STP  MST  Region  Instance  1
to
 VLAN Mapping               1     Disabled                Disabled
STP Loopguard
STP Bridge Assurance        1     Enabled                 Enabled
STP Port Type, Edge         1     Normal, Disabled,       Normal, Disabled,
BPDUFilter,           Edge  1     Disabled                Disabled
BPDUGuard
STP MST Simulate PVST             Enabled                 Enabled
Nve Oper State, Secondary   1     Up, 4.4.4.4             Up, 4.4.4.4
IP                          1     10002-11025             10002-11025
Nve Vni Configuration
Allowed VLANs               -     1-1025                  1-1025
```

```
          Local suspended VLANs       -     -                  -
```

2.7 Static Ingress Replication

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

With static ingress replication:

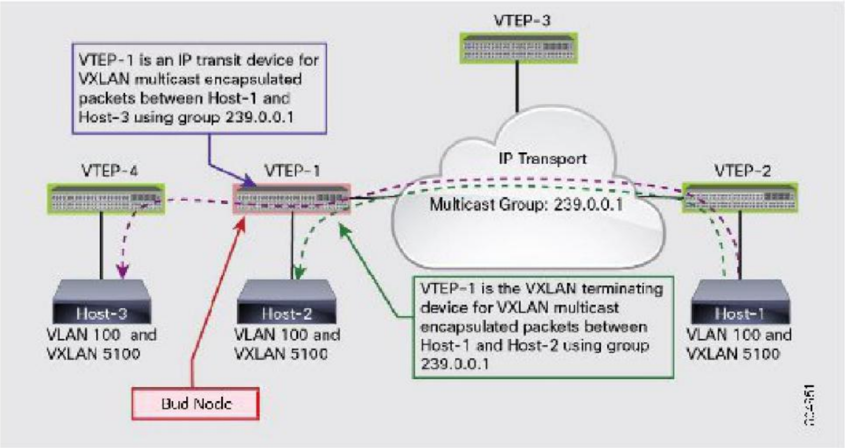
- Remote peers are statically configured.
- Multi-destination packets are unicast encapsulated and delivered to each of the statically configured remote peers.

2.8 Bud Node Topology

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

The Inspur CN12900 Series switches provide support for the bud node topology. The application leaf engine (ALE) of the device enables it to be a VXLAN VTEP device and an IP transit device at the same time so the device can become a bud node.

Figure 1： VXLAN Bud-Node Topology



VTEP-1 is an IP transit device for VXLAN multicast encapsulated packets between Host-1 and Host-3 using group 239.0.0.1

VTEP-1 is the VXLAN terminating device for VXLAN multicast encapsulated packets between Host-1 and Host-2 using group 239.0.0.1

2.9 VXLAN BGP EVPN Control Plane

An Inspur CN12900 Series switch can be configured to provide a BGP ethernet VPN (EVPN) control plane using a distributed anycast gateway, with Layer 2 and Layer 3 VXLAN overlay networks.

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

- Flexible workload placement that is not restricted with physical topology of the data center network.
- Virtual machines may be placed anywhere in the data center, without considerations of physical boundaries of racks.
- Optimal east-west traffic between servers within and across data centers

- East west traffic between servers/virtual machines is achieved by most specific routing at the first hop router, where the first hop routing is done at the access layer. Host routes must be exchanged to ensure most specific routing to and from servers/hosts. Virtual machine mobility is supported via detecting of virtual machine attachment and signaling new location to rest of the network.
- Eliminate or reduce flooding in the data center.
- Flooding is reduced by distributing MAC reachability information via BGP EVPN to optimize flooding relating to L2 unknown unicast traffic. Optimization of reducing broadcasts associated with ARP/IPv6 Neighbor solicitation is achieved via distributing the necessary information via BGP

EVPN and caching it at the access switches, address solicitation request can then locally responded without sending a broadcast.

- Standards based control plane that can be deployed independent of a specific fabric controller.
- The BGP EVPN control plane approach provides:
- IP reachability information for the tunnel endpoints associated with a segment and the hosts behind a specific tunnel endpoint.
- Distribution of host MAC reachability to reduce/eliminate unknown unicast flooding.
- Distribution of host IP/MAC bindings to provide local ARP suppression.
- Host mobility.
- A single address family (BGP EVPN) to distribute both L2 and L3 route reachability information.
- Segmentation of Layer 2 and Layer 3 traffic
- Traffic segmentation is achieved with using VXLAN encapsulation, where VNI acts as segment identifier.

CHAPTER 3 Configuring VXLAN

3.1 Information About VXLAN

3.1.1 Guidelines and Limitations for VXLAN

VXLAN has the following guidelines and limitations:

- 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
- 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
198.51.100.1/8 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 198.51.100.1/8 bidir
```

- When entering the **no feature pim** command, NVE ownership on the route is not removed so the route stays and traffic continues to flow. Aging is done by PIM. PIM does not age out entries having a VXLAN encap flag.
- 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.
- 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 Inspur CN12900 Series INOS-CN Unicast Routing Configuration Guide.
- For the Inspur CN12904 and CN12908 switches with -R line cards, the following features are not supported:
- VXLAN with vPC is not supported.
- DHCP snooping, ACL, and QoS policies are not supported on VXLAN VLANs.
- IGMP snooping is not supported on VXLAN enabled VLANs.
- For the Inspur CN12904 and CN12908 switches with -R line cards, VXLAN Layer 2 Gateway is supported on the CN129-X636C-R line card. VXLAN and MPLS cannot be enabled on the Inspur CN12908 switch at the same time.
- For the Inspur CN12904 and CN12908 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 CN129-X636C-R.
- For the Inspur CN12904 and CN12908 switches with -R line cards, PIM/ASM is supported in the underlay ports. PIM/Bidir is not supported. For more information, see the Inspur CN12900 Series INOS-CN Multicast Routing Configuration Guide.
- For the Inspur CN12904 and CN12908 switches with -R line cards, IPv6 hosts routing in the overlay is

supported.

- For the Inspur CN12904 and CN12908 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 Inspur CN12900 Series INOS-CN Unicast Routing Configuration Guide.

To overcome this, configure the sub-interfaces or move the uplinks to 10 G ports (non-GEM module ports) instead of 40 G ports.

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

- 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.
- **show** commands with the **internal** keyword are not supported.
- DHCP snooping (Dynamic Host Configuration Protocol snooping) is not supported on VXLAN VLANs.
- SPAN TX is not supported on VXLAN VTEP.
- RACLs are not supported on Layer 3 uplinks for VXLAN traffic. Egress VACLs support is not available for de-capsulated packets in the network to access direction on the inner payload.

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

- QoS classification is not supported for VXLAN traffic in the network to access direction on the Layer 3 uplink interface.
- The QoS buffer-boost feature is not applicable for VXLAN traffic.
- VTEP does not support Layer 3 subinterface uplinks that carry VXLAN encapsulated traffic.
- Layer 3 interface uplinks that carry VXLAN encapsulated traffic do not support subinterfaces for non-VxLAN encapsulated traffic.
- Non-VXLAN sub-interface VLANs cannot be shared with VXLAN VLANs.
- Subinterfaces on 40G (ALE) uplink ports are not supported on VXLAN VTEPs.
- Point to multipoint Layer 3 and SVI uplinks are not supported. Since both uplink types can only be enabled point-to-point, they cannot span across more than two switches.
- 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 Inspur INOS-CN, the UDP port number is 4789. It complies with IETF standards and is not configurable.
- VXLAN is supported on Inspur CN12900 Series switches with the following line cards:
- CN129-X6136YC-R
- CN129-X636Q-R

- CN129-X636C-R

VXLAN is not supported on the Inspur CN129-636 line cards. For VXLAN functionality on a modular platform, all line cards in that chassis have to support VXLAN features. You cannot have a chassis that is a hybrid of VXLAN enabled line cards and non-VXLAN enabled line cards.

- MDP is not supported for VXLAN configurations.
- Consistency checkers are not supported for VXLAN tables.
- ARP suppression is only supported for a VNI if the VTEP hosts the First-Hop Gateway (Distributed Anycast Gateway) for this VNI. The VTEP and the SVI for this VLAN have to be properly configured for the Distributed Anycast Gateway operation, for example, global anycast gateway MAC address configured and anycast gateway feature with the virtual IP address on the SVI.
- 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 co-existence with the GRE tunnel feature or the MPLS (static or segment-routing) feature on Inspur CN12900 Series switches with a Network Forwarding Engine (NFE).
- VXLAN does not support co-existence with MVR and MPLS for Inspur CN12904 and CN12908 with -R line cards.
- Resilient hashing (port-channel load-balancing resiliency) and VXLAN configurations are not compatible with VTEPs using ALE uplink ports.

3.2 Considerations for VXLAN Deployment

- The “System Routing Mode: template-vxlan-scale” is not applicable.
- 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.
- 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.
- 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.)
- VXLAN tunnels cannot have more than one underlay next hop on a given underlay port. For example, on a given output underlay port, only one destination MAC address can be derived as the outer MAC on a given output port.

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

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

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

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

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

    ip pim anycast-rp 1.1.1.10 1.1.1.2
```

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

3.3 VPC Considerations for VXLAN Deployment

- Bind NVE to a loopback address that is separate from other loopback addresses that are required by Layer 3 protocols. A best practice is to use a dedicated loopback address for VXLAN.
- On VPC VXLAN, it is recommended to increase the **delay restore interface-vlan** timer under the VPC configuration, if the number of SVIs are scaled up. For example, if there are 1000 VNIs with 1000 SVIs, it is recommended to increase the **delay restore interface-vlan** timer to 45 Seconds.
- If a ping is initiated to the attached hosts on VXLAN VLAN from a vPC VTEP node, the source IP address used by default is the anycast IP that is configured on the SVI. This ping can fail to get a response from the host in case the response is hashed to the vPC peer node. This issue can happen when a ping is initiated from a VXLAN vPC node to the attached hosts without using a unique source IP address. As a workaround for this situation, use VXLAN OAM or create a unique loopback on each vPC VTEP and route the unique address via a backdoor path.
- The loopback address used by NVE needs to be configured to have a primary IP address and a secondary IP address.

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

- VPC peers must have identical configurations.
- Consistent VLAN to VN-segment mapping.
- Consistent NVE1 binding to the same loopback interface
- Using the same secondary IP address.
- Using different primary IP addresses.
- Consistent VNI to group mapping.
- For multicast, the VPC node that receives the (S, G) join from the RP (rendezvous point) becomes the DF (designated forwarder). On the DF node, encap routes are installed for multicast.

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

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

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

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

- When peer-link is shut, the loopback interface used by NVE on the VPC secondary is brought down and the status is **Admin Shut**. This is done so that the route to the loopback is withdrawn on the upstream and that the upstream can divert all traffic to the VPC primary.
- When peer-link is no-shut, the NVE loopback address is brought up again and the route is advertised upstream, attracting traffic.
- For VPC, the loopback interface has 2 IP addresses: the primary IP address and the secondary IP address. The primary IP address is unique and is used by Layer 3 protocols.

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

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

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

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

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

```
switch# sh ru vpc

version 9.2(2)
feature vpc
vpc domain 2
  peer-switch
  peer-keepalive destination 172.29.206.65 source 172.29.206.64
  peer-gateway
    ipv6 nd synchronize
    ip arp synchronize
```

- When the NVE or loopback is shut in VPC configurations:
- If the NVE or loopback is shut only on the primary VPC switch, the global VXLAN VPC consistency checker fails. Then the NVE, loopback, and VPCs are taken down on the secondary VPC switch.
- If the NVE or loopback is shut only on the secondary VPC switch, the global VXLAN VPC consistency checker fails. Then the NVE, loopback, and secondary VPC are brought down on the secondary. Traffic continues to flow through the primary VPC switch.

As a best practice, you should keep both the NVE and loopback up on both the primary and secondary VPC switches.

- Redundant anycast RPs configured in the network for multicast load-balancing and RP redundancy are supported on VPC VTEP topologies.
- Enabling vpc peer-gateway configuration is mandatory. For peer-gateway functionality, at least one backup routing SVI is required to be enabled across peer-link and also configured with PIM. This provides a backup routing path in the case when VTEP loses complete connectivity to the spine. Remote peer reachability is re-routed over peer-link in his case. In BUD node topologies, the backup SVI needs to be added as a static OIF for each underlay multicast group.

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

```
sswithch# sh ru int vlan 2

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

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

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

Using the **ip forward** command enables the VTEP to forward the VXLAN de-capsulated packet destined to its router IP to the SUP/CPU.

3.4 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, Inspur CN12900 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 Inspur CN12900 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.

3.5 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.*
- The use of the **system nve infra-vlans** command is required for the VLANs carried on the peer-link that are not locally mapped to an L2VNI segment.

3.6 Configuring VXLAN

3.6.1 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.

3.6.2 Mapping VLAN to VXLAN VNI

Procedure

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

3.7 Configuring Port VLAN Mapping on a Trunk Port

You can configure VLAN translation between the ingress (incoming) VLAN and a local (translated) VLAN on a port. For the traffic arriving on the interface where VLAN translation is enabled, the incoming VLAN is mapped to a translated VLAN that is VXLAN enabled.

On the underlay, this is mapped to a VNI, the inner dot1q is deleted, and switched over to the VXLAN network. On the egress switch, the VNI is mapped to a translated VLAN. On the outgoing interface, where VLAN translation is configured, the traffic is converted to the original VLAN and egress out. Refer to the VLAN counters on the translated VLAN for the traffic counters and not on the ingress VLAN. Port VLAN (PV) mapping is an access side feature and is supported with both multicast and ingress replication for flood and learn and BGP EVPN mode for VXLAN.

Notes for Port VLAN Mapping:

- 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.
- 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 Inspur CN12900 Series switches with a Network Forwarding Engine (NFE).
- When changing a property on a translated VLAN, the port that has mapping a configuration with that VLAN as the translated VLAN, should be flapped to ensure correct behavior.

For example:

```
Int eth 1/1
switchport vlan mapping 101 10
.
.
.
/***Deleting vn-segment from vlan 10.***/
/***Adding vn-segment back.***/
/***Flap Eth 1/1 to ensure correct behavior.***/
```

- The following is an example of overlapping VLAN for PV translation. In the first statement, VLAN-102 is a translated VLAN with VNI mapping. In the second statement, VLAN-102 the VLAN where it is translated to VLAN-103 with VNI mapping.

```
interface ethernet1/1 switchport
vlan mapping 101 102 switchport
vlan mapping 102 103/
```

When adding a member to an existing port channel using the **force** command, the "mapping enable" configuration must be consistent.

For example:

```
Int po 101
switchport vlan mapping enable
switchport vlan mapping 101 10
switchport trunk allowed vlan 10
int eth 1/8
/***No configuration***/
```

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

```
int eth 1/8
```

```
switchport vlan mapping enable
channel-group 101 force
```

Port VLAN switching is supported on Inspur CN12900 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 an L2 VNI. There is an operational benefit of terminating different customer VLANs and mapping them to the fabric-managed-VLANs, L2 VNIs.

- An NVE interface with VNI mapping should be configured for PV translation to work.

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.
- Ensure that all translated VLANs are VXLAN enabled.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**interface** *type port*	Enters interface configuration mode.
Step 3	[**no**] **switchport vlan mapping enable**	Enables VLAN translation on the switch port. VLAN translation is disabled by default. Note Use the **no** form of this command to disable VLAN translation.
Step 4	[**no**] **switchport vlan mapping** *vlan-id translated-vlan-id*	Translates a VLAN to another VLAN. • The range for both the *vlan-id* and *translated-vlan-id* arguments is from 1 to 4094. • You can configure VLAN translation between the ingress (incoming) VLAN and a local (translated) VLAN on a port. For the traffic arriving on the interface where VLAN translation is enabled, the incoming 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

	Command or Action	Purpose
		interface, where VLAN translation is configured, the traffic is converted to the original VLAN and egress out. **Note** Use the **no** form of this command to clear the mappings between a pair of VLANs.
Step 5	[**no**] **switchport vlan mapping all**	Removes all VLAN mappings configured on the interface.
Step 6	(Optional) **copy running-config startup-config**	Copies the running configuration to the startup configuration. **Note** The VLAN translation configuration does not become effective until the switch port becomes an operational trunk port
Step 7	(Optional) **show interface** [*if-identifier*] **vlan mapping**	Displays VLAN mapping information for a range of interfaces or for a specific interface.

Example

This example shows how to configure VLAN translation between (the ingress) VLAN 10 and (the local) VLAN 100. The **show vlan counters** command output shows the statistic counters as translated VLAN instead of customer VLAN.

```
switch# config t
switch(config)# interface ethernet1/1
switch(config-if)#   switchport vlan mapping enable
switch(config-if)# switchport vlan mapping 10 100
switch(config-if)# switchport trunk allowed vlan 100
switch(config-if)# show interface ethernet1/1 vlan mapping
Interface eth1/1:
Original VLAN          Translated VLAN
------------------     ---------------
10                          100
switch(config-if)# show vlan counters
Vlan Id                            :100
Unicast Octets In                  :292442462
Unicast Packets In                 :1950525
Multicast Octets In                :14619624
Multicast Packets In               :91088
Broadcast Octets In                :14619624
Broadcast Packets In               :91088
Unicast Octets Out                 :304012656
Unicast Packets Out                :2061976
L3 Unicast Octets In               :0
L3 Unicast Packets In              :0
```

3.8 Configuring Inner VLAN and Outer VLAN Mapping on a Trunk Port

You can configure VLAN translation from an inner VLAN and an outer VLAN to a local (translated) VLAN on a port. For the double tag VLAN traffic arriving on the interfaces where VLAN translation is enabled, the inner VLAN and outer VLAN are mapped to a translated VLAN that is VXLAN enabled.

Notes for configuring inner VLAN and outer VLAN mapping:

- Inner and outer VLAN cannot be on the trunk allowed list on a port where inner VLAN and outer VLAN is configured.

For example:

```
switchport vlan mapping 11 inner 12 111
switchport trunk allowed vlan 11-12,111 /***Not valid because 11 is outer VLAN and
12 is inner VLAN.***/
```

- On the same port, no two mapping (translation) configurations can have the same outer (or original) or translated VLAN. Multiple inner VLAN and outer VLAN mapping configurations can have the same inner VLAN.

For example:

```
switchport vlan mapping 101 inner 102 1001
switchport vlan mapping 101 inner 103 1002  /***Not valid because 101 is already used
as an original VLAN.***/
switchport vlan mapping 111 inner 104 1001  /***Not valid because 1001 is already used
 as a translated VLAN.***/
switchport vlan mapping 106 inner 102 1003  /***Valid because inner vlan can be the
same.***/
```

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**interface** *type port*	Enters interface configuration mode.
Step 3	[**no**] **switchport mode trunk**	Enters trunk configuration mode.
Step 4	**switchport vlan mapping enable**	Enables VLAN translation on the switch port. VLAN translation is disabled by default. **Note** Use the **no** form of this command to disable VLAN translation.
Step 5	**switchport vlan mapping** *outer-vlan-id* **inner** *inner-vlan-id translated-vlan-id*	Translates inner VLAN and outer VLAN to another VLAN.
Step 6	(Optional) **copy running-config startup-config**	Copies the running configuration to the startup configuration. **Note** The VLAN translation configuration

		does not become effective until the switch port becomes an operational trunk port
Step 7	(Optional) **show interface** [*if-identifier*] **vlan mapping**	Displays VLAN mapping information for a range of interfaces or for a specific interface.

Example

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

```
switch# config t
switch(config)# interface ethernet1/1
switch(config-if)# switchport mode trunk
switch(config-if)# switchport vlan mapping enable
switch(config-if)# switchport vlan mapping 11 inner 12
111
switch(config-if)# switchport trunk allowed vlan 101-170
switch(config-if)# no shutdown
switch(config-if)# show mac address-table dynamic vlan 111

Legend:
        * - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
        age - seconds since last seen,+ - primary entry using vPC Peer-Link,
        (T) - True, (F) - False
   VLAN      MAC Address      Type      age      Secure NTFY  Ports
         +-----------------+         +--------   +------+----  +------------------
*---------111      0000.0092.0001  dynamic--------  0         F      F    nve1(100.100.100.254)
*  111      0000.0940.0001  dynamic  0         F      F    Eth1/1
```

3.9 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** *x*	Creates a VXLAN overlay interface that terminates VXLAN tunnels. **Note** Only 1 NVE interface is allowed on the switch.
Step 3	**source-interface** *src-if*	The source interface must be a loopback interface that is configured on the switch with a valid /32 IP address. This /32 IP address must be known by the transient devices in the transport network and the remote VTEPs. This

	Command or Action	Purpose
		is accomplished by advertising it through a dynamic routing protocol in the transport network .
Step 4	**member vni** *vni*	Associate VXLAN VNIs (Virtual Network Identifiers) with the NVE interface.
Step 5	**mcast-group** *start-address* [*end-address*]	Assign a multicast group to the VNIs. **Note** used only for BUM traffic

3.10 Disabling VXLANs

Procedure

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

3.11 Configuring BGP EVPN Ingress Replication

The following enables BGP EVPN with ingress replication for peers.

Procedure

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

		is accomplished by advertising it through a dynamic routing protocol in the transport network.
Step 4	**member vni** *vni*	Associate VXLAN VNIs (Virtual Network Identifiers) with the NVE interface.
Step 5	**ingress-replication protocol bgp**	Enables BGP EVPN with ingress replication for the VNI.

3.12 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** *x*	Creates a VXLAN overlay interface that terminates VXLAN tunnels. **Note** Only 1 NVE interface is allowed on the switch.
Step 3	**member vni** [*vni-id* | *vni-range*]	Maps VXLAN VNIs to the NVE interface.
Step 4	**ingress-replication protocol static**	Enables static ingress replication for the VNI.
Step 5	**peer-ip** *n.n.n.n*	Enables peer IP.

3.13 Configuring Q-in-VNI

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

Notes about configuring a Q-in-VNI:

- Q-in-VNI only supports VXLAN bridging. It does not support VXLAN routing.
- The dot1q mode does not support 40G ports on Inspur CN12900 platform switches.

Before you begin

Configuring the Q-in-VNI feature requires:

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

Procedure

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

	Command or Action	Purpose
Step 2	**interface** *type port*	Enters interface configuration mode.
Step 3	**switchport mode dot1q-tunnel**	Creates a 802.1Q tunnel on the port.
Step 4	**switchport access vlan** *vlan-id*	Specifies the port assigned to a VLAN.
Step 5	**spanning-tree bpdufilter enable**	Enables BPDU Filtering for the specified spanning tree edge interface. By default, BPDU Filtering is disabled.
Step 6	**interface nve** *x*	Creates a VXLAN overlay interface that terminates VXLAN tunnels.
Step 7	**overlay-encapsulation vxlan-with-tag**	Enables Q-in-VNI.

Example

- The following is an example of configuring a Q-in-VNI:

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

3.14 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).

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

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

- Configuring selective Q-in-VNI on one VXLAN and configuring plain Q-in-VNI on the VXLAN peer is supported. Configuring one port with selective Q-in-VNI and the other port with plain Q-in-VNI on the same switch is supported.
- Selective Q-in-VNI is an ingress VLAN tag-policing feature. Only ingress VLAN tag policing is performed with respect to the selective Q-in-VNI configured range.

For example, selective Q-in-VNI customer VLAN range of 100-200 is configured on VTEP1 and customer VLAN range of 200-300 is configured on VTEP2. When traffic with VLAN tag of 175 is sent from VTEP1 to VTEP2, the traffic is accepted on VTEP1, since the VLAN is in the configured range and it is forwarded to the VTEP2. On VTEP2, even though VLAN tag 175 is not part of the configured range, the packet egresses out of the selective Q-in-VNI port. If a packet is sent with VLAN tag 300 from VTEP1, it is dropped because 300 is not in VTEP1's selective Q-in-VNI configured range.

- Configure the **system dot1q-tunnel transit** CLI on the vPC switches with selective Q-in-VNI configurations. This CLI configuration is required to retain the inner Q-tag as the packet goes over the vPC peer link when one of the vPC peers has an orphan port. With this CLI configuration, the **vlan dot1Q tag native** functionality does not work.
- The native VLAN configured on the selective Q-in-VNI port cannot be a part of the customer VLAN range. If the native VLAN is part of the customer VLAN range, the configuration is rejected.

The provider VLAN can overlap with the customer VLAN range. For example, **switchport vlan mapping 100-1000 dot1q-tunnel 200**

- By default, the native VLAN on any port is VLAN 1. If VLAN 1 is configured as part of the customer VLAN range using the **switchport vlan mapping** <*range*>**dot1q-tunnel** <*sp-vlan*> CLI command, the traffic with customer VLAN 1 is not carried over as VLAN 1 is the native VLAN on the port. If customer wants VLAN 1 traffic to be carried over the VXLAN cloud, they should configure a dummy native VLAN on the port whose value is outside the customer VLAN range.
- To remove some VLANs or a range of VLANs from the configured switchport VLAN mapping range on the selective Q-in-VNI port, use the **no** form of the **switchport vlan mapping** <*range*>**dot1q-tunnel** <*sp-vlan*> CLI command.

For example, VLAN 100-1000 is configured on the port. To remove VLAN 200-300 from the configured range, use the **no switchport vlan mapping** <*200-300*> **dot1q-tunnel** <*sp-vlan*> CLI command.

```
interface Ethernet1/32
  switchport switchport
  mode trunk
  switchport trunk native vlan 4049
  switchport vlan mapping 100-1000 dot1q-tunnel 21
  switchport trunk allowed vlan 21,4049 spanning-
  tree bpdufilter enable
  no shutdown

switch(config-if)# no sw vlan mapp 200-300 dot1q-tunnel

21 switch(config-if)# sh run int e 1/32

version 9.2(2)

interface Ethernet1/32
  switchport switchport
  mode trunk
  switchport trunk native vlan 4049
  switchport vlan mapping 100-199,301-1000 dot1q-tunnel 21
  switchport trunk allowed vlan 21,4049
  no shutdown
```

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

See the following configuration examples.

- See the following example for the provider VLAN configuration:

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

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

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

- See the following example for the interface nve configuration:

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

- See the following example for the native VLAN configuration:

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

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

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

3.15 Configuring Q-in-VNI with LACP Tunneling

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

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**interface** *type port*	Enters interface configuration mode.
Step 3	**switchport mode dot1q-tunnel**	Enables dot1q-tunnel mode.
Step 4	**switchport access vlan** *vlan-id*	Specifies the port assigned to a VLAN.
Step 5	**interface nve** *x*	Creates a VXLAN overlay interface that terminates VXLAN tunnels.
Step 6	**overlay-encapsulation vxlan-with-tag tunnel-control-frames lacp**	Enables Q-in-VNI for LACP tunneling.

	Command or Action	Purpose

Example

- The following is an example of configuring a Q-in-VNI for LACP tunneling:

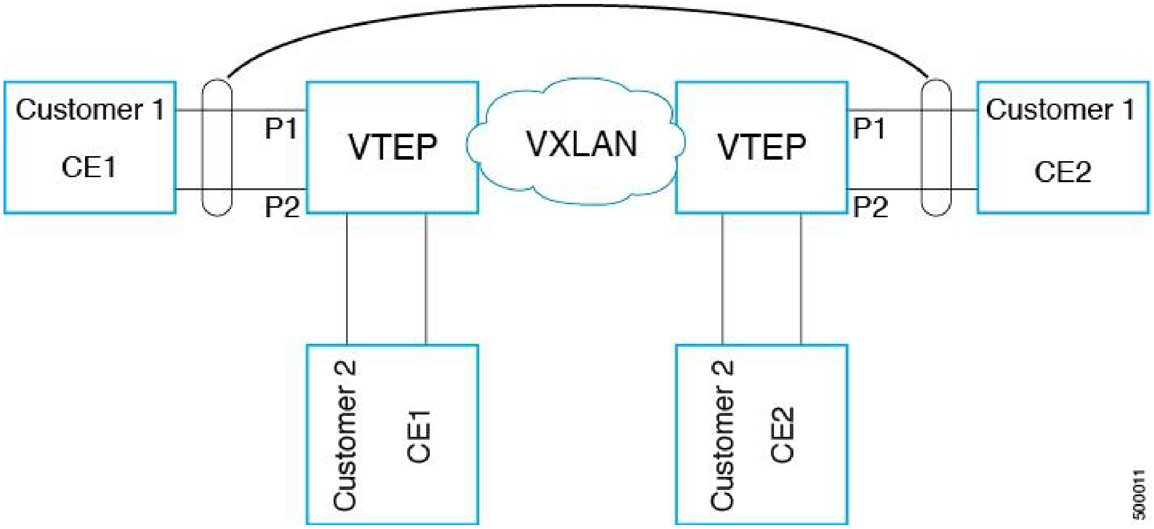
```
switch# config terminal
switch(config)# interface ethernet 1/4
switch(config-if)# switchport mode dot1q-tunnel
switch(config-if)# switchport access vlan 10
switch(config-if)# spanning-tree bpdufilter
enable
switch(config-if)# interface nve1
switch(config-if)# overlay-encapsulation vxlan-with-tag tunnel-control-frames
```

- The following is an example of configuring a Q-in-VNI for LACP tunneling:

```
switch# config terminal
switch(config)# interface ethernet 1/4
switch(config-if)# switchport mode dot1q-tunnel
switch(config-if)# switchport access vlan 10
switch(config-if)# spanning-tree bpdufilter
enable
switch(config-if)# interface nve1
switch(config-if)# overlay-encapsulation vxlan-with-tag tunnel-control-frames lacp
```

- The following is an example topology that pins each port of a port-channel pair to a unique VM. The port-channel is stretched from the CE perspective. There is no port-channel on VTEP. The traffic on P1 of CE1 transits to P1 of CE2 using Q-in-VNI.

Figure 2：Tunneling Over VXLAN P2P Tunnels



3.15.1 Removing a VNI

Use this procedure to remove a VNI.

Procedure

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

3.16 Configuring IGMP Snooping Over VXLAN

3.16.1 Overview of IGMP Snooping Over VXLAN

You can configure IGMP snooping over VXLAN. This feature is available on the Inspur CN12908 switch with CN129-X636C-R line cards.

You can configure IGMP snooping over VXLAN. The configuration of IGMP snooping is same in VXLAN as in configuration of IGMP snooping in regular VLAN domain. For more information on IGMP snooping, see the *Configuring IGMP Snooping* section in Inspur CN12900 Series INOS-CN Multicast Routing Configuration Guide.

3.16.2 Guidelines and Limitations for IGMP Snooping Over VXLAN

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

- For IGMP snooping over VXLAN, all the guidelines and limitations of VXLAN apply.

3.16.3 Configuring IGMP Snooping Over VXLAN

Procedure

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

Step 3	switch(config)#**system nve ipmc global index-size ?**
	Example:
	`switch(config)# system nve ipmc global`
	`index-size ?`
	`<1000-7000> Ipmc allowed size`
		Configures the VXLAN global IPMC index size. IGMP snooping over VXLAN uses the IPMC indexes from the NVE global range on the Inspur CN12900 Series switches with Network Forwarding Engine (NFE). You need to reconfigure the VXLAN global IPMC index size according to the scale using this command.
		Inspur recommends to reserve 6000 IPMC indexes using this CLI command. The default

	Command or Action	Purpose
		IPMC index size is 3000. **Note** This command is not available on the Inspur CN12908 platform switch.

Step 4	switch(config)# **ip igmp snooping vxlan-umc drop vlan ?**
	Example:
	`switch(config)# ip igmp snooping vxlan-umc drop vlan ?`
	`<1-3863> VLAN IDs for which unknown multicast traffic is dropped`
		Configures IGMP snooping over VXLAN to drop all the unknown multicast traffic on per VLAN basis using this global CLI command. On Inspur CN12900 Series switches with Network Forwarding Engine (NFE), the default behavior of all unknown multicast traffic is to flood to the bridge domain.
		Note This command is not available on the Inspur CN12908 platform switch.

3.17 Configuring Line Cards for VXLAN

This procedure applies only to the Inspur CN12908 switch.

This procedure configures lines cards for either VXLAN or MPLS. All line cards in the chassis must be either VXLAN or MLPS. They cannot be mixed.

Procedure

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

Step 2	**hardware profile [vxlan | mpls] module** {*module* | **all**}
	Example:
	`switch(config)# hardware profile vxlan module all`
		Configures VXLAN on all line cards.
		Note All line cards must be either VXLAN or MLPS. They cannot be mixed.

Step 3	switch(config)# **reload** **Example:** `switch(config)# reload`	Reloads the Inspur INOS-CN software.
Step 4	switch(config)# **show hardware profile module** [*module* | **all**] **Example:** `switch(config)# show hardware profile module all`	Displays the line cards that are configured with VXLAN.

3.18 Centralized VRF Route Leaking using Default-Routes and Aggregates

3.18.1 Overview

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

3.18.2 Deploying EVPN

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

Figure 3： EVPN Brown-field Deployment

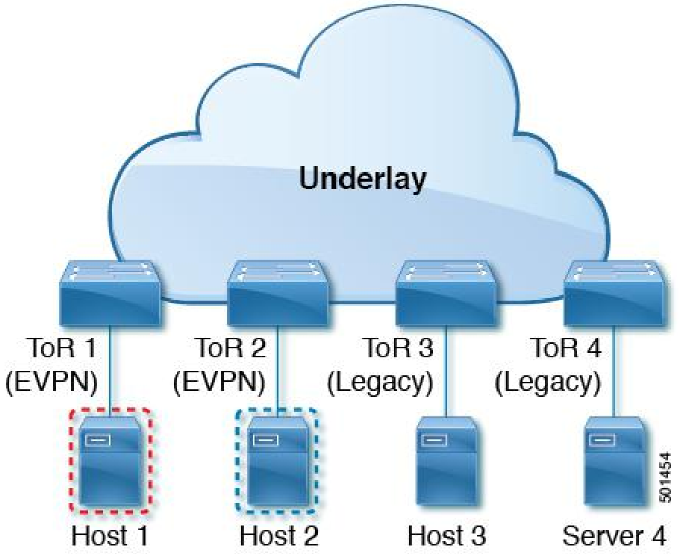


Figure 4：Border Leaf Connection to Core / Internet via Default-VRF

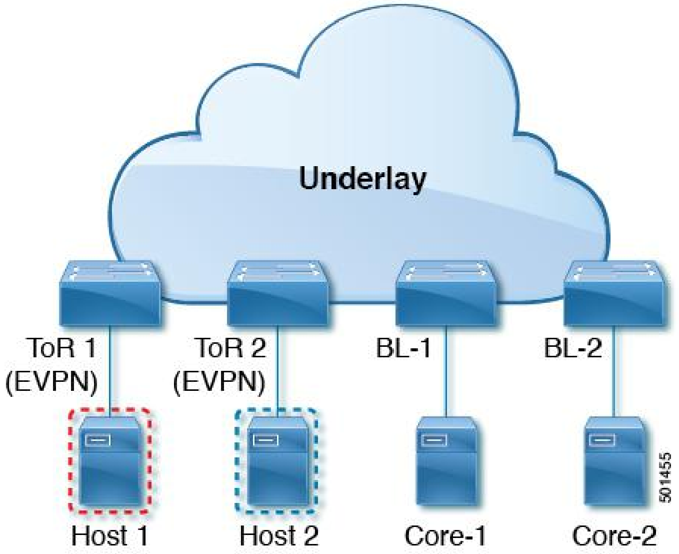
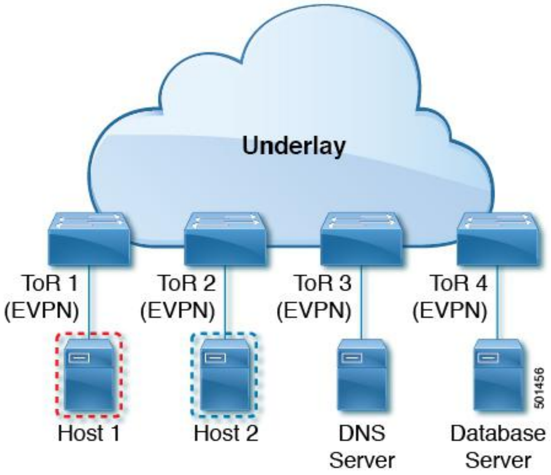


Figure 5：Common Services



3.18.3 Reachability between Leaves

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

3.18.4 VPN to Default-VRF Reachability

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

3.18.5 Guidelines and Limitations

- Each prefix needs to be imported into each VRF for full EVPN Cross-VRF Reachability.
- Memory complexity of the deployment can be described by a O(NxM) formula, where N is the number of prefixes, M is the number of EVPN VRFs.
- You must configure “feature bgp” to have access to “export vrf default” command. In order to achieve the full Centralized Route Leaking on EVPN, you must support downstream VNI assignment.
- Centralized route leaking applies the longest prefix matching. A leaf with a less specific local route, may not be able to reach a more specific address of that route’s subnet from another VNI, unless you manually configure the border leaf switch to generate those advertisements.
- Hardware support for VXLAN packet re-encapsulation at BL is required for this functionality to work in EVPN.

3.18.6 Configuration Examples for Centralized VRF Route Leak

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

```
vrf context vrf120
  vni 300120
  ip route 0.0.0.0/0 Null0                 // static default route
  ipv6 route ::/0 Null0       // static default route

 rd auto
  address-family ipv4 unicast
    route-target import 65535:120
    route-target import 65535:120 evpn
    route-target export 65535:120
    route-target export 65535:120 evpn
    import vrf default map permitall   // Imports from default VRF to tenant VRF
    export vrf default 100 map block_default allow-vpn
  address-family ipv6 unicast
    route-target import 65535:120
    route-target import 65535:120 evpn
    route-target export 65535:120
    route-target export 65535:120 evpn
    import vrf default map permitall
    export vrf default 100 map block_default_v6 allow-vpn
```

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

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

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

```
    switch# sh run vrf vrf150
    !Command: show running-config vrf vrf150
    !Time: Mon Oct 1 16:54:57 2018
    version 9.2(2)
    interface Vlan150


  vrf member vrf150
vrf context vrf150
  vni 300150
  rd auto
  address-family ipv4 unicast
    route-target import 65535:150
    route-target import 65535:150 evpn
    route-target import 65535:250                //import VRF250 routes
    route-target import 65535:250 evpn       //import VRF250 routes
    route-target export 65535:150
    route-target export 65535:150 evpn
  address-family ipv6 unicast
    route-target import 65535:150
    route-target import 65535:150 evpn
    route-target import 65535:250               //import VRF250 routes
    route-target import 65535:250 evpn      //import VRF250 routes
    route-target export 65535:150
    route-target export 65535:150 evpn
router bgp 1001
  vrf vrf150
    address-family ipv4 unicast
      advertise l2vpn evpn
      redistribute hmm route-map permitall
      aggregate-address 12.50.0.0/15             //VRF250 has network 12.50.0.0/16
      aggregate-address 22.50.0.0/15             //VRF250 has network 22.50.0.0/16
      maximum-paths 64
      maximum-paths ibgp 64
    address-family ipv6 unicast
      advertise l2vpn evpn
      redistribute hmm route-map permitall
      aggregate-address 2001:0:12:50::/63      //VRF250 has network 2001:0:12:50::/64
      aggregate-address 2001:0:22:50::/63      //VRF250 has network 2001:0:12:50::/64
      maximum-paths 64
      maximum-paths ibgp 64


switch# sh run vrf vrf250
!Command: show running-config vrf vrf250
!Time: Thu Oct  3 17:21:22 2017
version 9.2(2)
interface Vlan250
  vrf member vrf250
vrf context vrf250
  vni 300250
  rd auto
  address-family ipv4 unicast
    route-target import 65535:150
    route-target import 65535:150 evpn
    route-target import 65535:250
    route-target import 65535:250 evpn
    route-target export 65535:250
    route-target export 65535:250 evpn
  address-family ipv6 unicast
    route-target import 65535:150
    route-target import 65535:150 evpn
    route-target import 65535:250
    route-target import 65535:250 evpn
    route-target export 65535:250
    route-target export 65535:250 evpn
router bgp 1001
  vrf vrf250
    address-family ipv4 unicast
      advertise l2vpn evpn

          redistribute hmm route-map permitall
          aggregate-address 11.50.0.0/15
          aggregate-address 21.50.0.0/15
          maximum-paths 64
          maximum-paths ibgp 64
        address-family ipv6 unicast
          advertise l2vpn evpn
          redistribute hmm route-map permitall
          aggregate-address 2001:0:11:50::/63
          aggregate-address 2001:0:21:50::/63
          maximum-paths 64
          maximum-paths ibgp 64
```

3.19 Verifying the VXLAN Configuration

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

Table 2：Display VXLAN configuration information

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

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

Table 3：Display VXLAN configuration information

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

Command	Purpose
	destination or UDP port.
show mac address-table static interface nve 1	Displays static MAC information.
show vxlan interface	Displays VXLAN interface status for 9200 platform switches. .
show vxlan interface | count	Displays VXLAN VLAN logical port VP count. **Note** A VP is allocated on a per-port per-VLAN basis. The sum of all VPs across all VXLAN-enabled Layer 2 ports gives the total logical port VP count. For example, if there are 10 Layer 2 trunk interfaces, each with 10 VXLAN VLANs, then the total VXLAN VLAN logical port VP count is 10*10 = 100.

Table 4： Display VXLAN configuration information

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

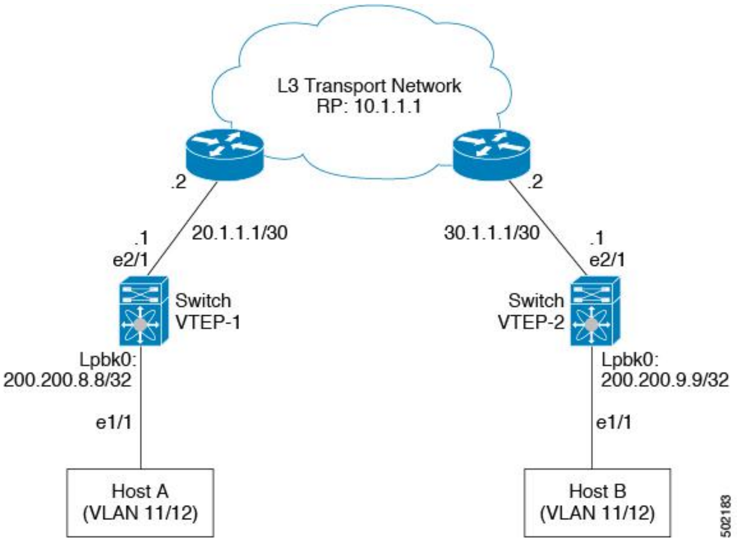
Table 5： Display VXLAN configuration information

Command	Purpose
show vlan private vlan	Displays the mappings between the primary and secondary VLANs and also the ports associated with each of the VLANs
show tech-support private vlan	Displays PVLAN related tech-support information. This is useful for debugging.

3.20 Example of VXLAN Bridging Configuration

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

Figure 6： VXLAN topology for VTEP



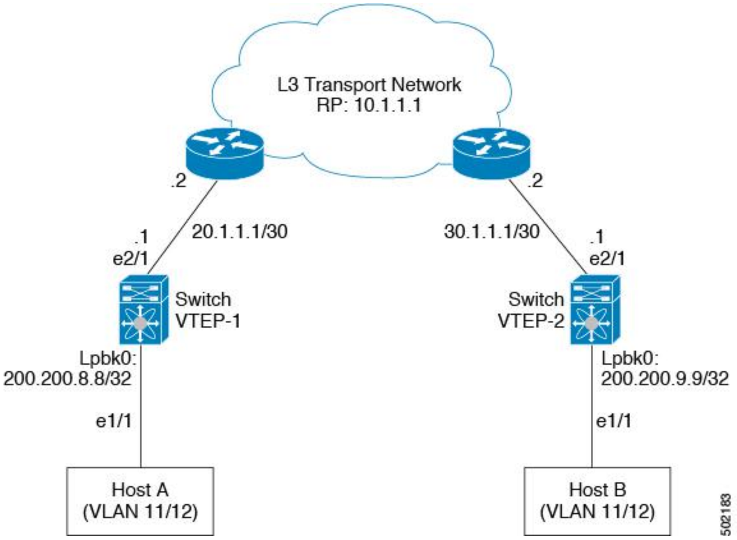
- Inspur CN12900 VTEP-1 configuration:

```
switch-vtep-1(config)# feature ospf
switch-vtep-1(config)# feature pim
switch-vtep-1(config)# router ospf 1
switch-vtep-1(config-router)# router-id 100.100.100.1
switch-vtep-1(config)# ip pim rp-address 10.1.1.1 group-list 224.0.0.0/4
switch-vtep-1(config)# interface loopback0
switch-vtep-1(config-if)# ip address 100.100.100.1/32
switch-vtep-1(config-if)# ip router ospf 1 area
0.0.0.0
switch-vtep-1(config-if)# ip pim sparse-mode
switch-vtep-1(config)# interface e2/1
switch-vtep-1(config-if)# ip address 20.1.1.1/30
switch-vtep-1(config-if)# ip router ospf 1 area
0.0.0.0
switch-vtep-1(config-if)# ip pim sparse-mode
switch-vtep-1(config)# feature nv overlay
switch-vtep-1(config)# feature vn-segment-vlan-based
switch-vtep-1(config)# interface e1/1
switch-vtep-1(config-if)# switchport
switch-vtep-1(config-if)# switchport access vlan
10 switch-vtep-1(config-if)# no shutdown
switch-vtep-1(config)# interface nve1
switch-vtep-1(config-if)# no shutdown
switch-vtep-1(config-if)# source-interface
loopback0
switch-vtep-1(config-if)# member vni 10000 mcast-group
230.1.1.1
switch-vtep-1(config)# vlan 10
switch-vtep-1(config-vlan)# vn-segment
10000
switch-vtep-1(config-vlan)# exit
switch-vtep-2(config)# feature ospf
```

```
switch-vtep-2(config)# feature pim
switch-vtep-2(config)# router ospf 1
switch-vtep-2(config-router)# router-id
100.100.100.2
switch-vtep-2(config)# ip pim rp-address 10.1.1.1 group-list
224.0.0.0/4
switch-vtep-2(config)# interface loopback0
switch-vtep-2(config-if)# ip address 100.100.100.2/32
switch-vtep-2(config-if)# ip router ospf 1 area
0.0.0.0
switch-vtep-2(config-if)# ip pim sparse-mode
switch-vtep-2(config)# interface e2/1
switch-vtep-2(config-if)# ip address 30.1.1.1/30
switch-vtep-2(config-if)# ip router ospf 1 area
0.0.0.0
switch-vtep-2(config-if)# ip pim sparse-mode
switch-vtep-2(config)# feature nv overlay
switch-vtep-2(config)# feature vn-segment-vlan-
based
switch-vtep-2(config)# interface e1/1
switch-vtep-2(config-if)# switchport
switch-vtep-2(config-if)# switchport access vlan 10
switch-vtep-2(config-if)# no shutdown
switch-vtep-2(config)# interface nve1
switch-vtep-2(config-if)# no shutdown
switch-vtep-2(config-if)# source-interface
loopback0

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

- An example of an ingress replication topology:

Figure 7：Ingress Replication topology



- Inspur CN12900 VTEP-1 configuration:

```
switch-vtep-1(config)# feature ospf
switch-vtep-1(config)# router ospf 1
switch-vtep-1(config-router)# router-id 200.200.8.8
switch-vtep-1(config)# interface loopback0
switch-vtep-1(config-if)# ip address 200.200.8.8/32
switch-vtep-1(config-if)# ip router ospf 1 area 0.0.0.0
switch-vtep-1(config)# interface e2/1
switch-vtep-1(config-if)# ip address 20.1.1.1/30
switch-vtep-1(config-if)# ip router ospf 1 area 0.0.0.0
switch-vtep-1(config-if)# ip pim sparse-mode
switch-vtep-1(config)# feature nv overlay
switch-vtep-1(config)# feature vn-segment-vlan-based
switch-vtep-1(config)# interface e1/1
switch-vtep-1(config-if)# switchport
switch-vtep-1(config-if)# switch port mode trunk
switch-vtep-1(config-if)# switch port allowed vlan 11-12
switch-vtep-1(config-if)# no shutdown
switch-vtep-1(config)# vlan 11
switch-vtep-1(config-vlan)# vn-segment 10011
switch-vtep-1(config)# vlan 12
switch-vtep-1(config-vlan)# vn-segment 10012
switch-vtep-1(config)# interface nve1
switch-vtep-1(config-if)# no shutdown
switch-vtep-1(config-if)# source-interface loopback0
switch-vtep-1(config-if)# member vni 10011
switch-vtep-1(config-if)# ingress-replication protocol static
switch-vtep-1(config-if)# peer_ip 200.200.9.9
switch-vtep-1(config-if)# member vni 10012
switch-vtep-1(config-if)# ingress-replication protocol static
switch-vtep-1(config-if)# peer_ip 200.200.9.9
switch-vtep-1(config-vlan)# exit
switch-vtep-1# show nve vni ingress-replication

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

- Inspur CN12900 VTEP-2 configuration:

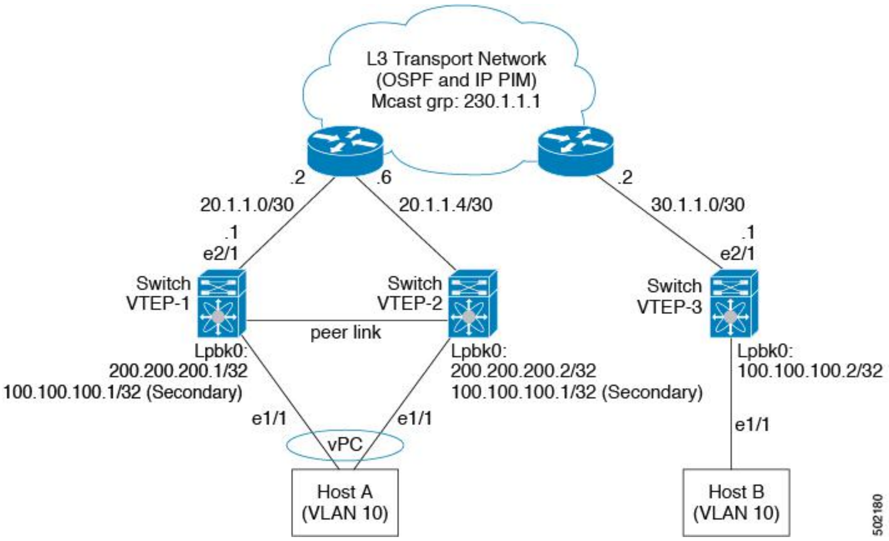
```
switch-vtep-2(config)# feature ospf
switch-vtep-2(config)# router ospf 1
switch-vtep-2(config-router)# router-id 200.200.9.9 switch-vtep-2(config)#
interface loopback0
switch-vtep-2(config-if)# ip address 200.200.9.9/32 switch-vtep-2(config-
if)# ip router ospf 1 area 0.0.0.0 switch-vtep-2(config)#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

```
switch-vtep-2(config-if)# ingress-replication protocol static
switch-vtep-2(config-if)# peer_ip 200.200.8.8
switch-vtep-2(config-if)# member vni 10012
switch-vtep-2(config-if)# ingress-replication protocol static
switch-vtep-2(config-if)# peer_ip 200.200.8.8
switch-vtep-2(config-vlan)# exit
```

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

nve1      10012    200.200.8.8       07:42:23
```

- For a vPC VTEP configuration, the loopback address requires a secondary IP. An example of a vPC VTEP configuration

Figure 8：VXLAN topology for vPC VTEP



- Switch VTEP-1 configuration:

```
switch-vtep-1(config)# feature nv overlay
switch-vtep-1(config)# feature vn-segment-vlan-based
switch-vtep-1(config)# feature ospf
switch-vtep-1(config)# feature pim
switch-vtep-1(config)# router ospf 1
switch-vtep-1(config-router)# router-id 200.200.200.1
switch-vtep-1(config)# ip pim rp-address 10.1.1.1 group-list 224.0.0.0/4
switch-vtep-1(config)# interface loopback0
switch-vtep-1(config-if)# ip address 200.200.200.1/32
switch-vtep-1(config-if)# ip address 100.100.100.1/32 secondary
switch-vtep-1(config-if)# ip router ospf 1 area 0.0.0.0
switch-vtep-1(config-if)# ip pim sparse-mode
switch-vtep-1(config)# interface e2/1
switch-vtep-1(config-if)# ip address 20.1.1.1/30
switch-vtep-1(config-if)# ip router ospf 1 area 0.0.0.0
switch-vtep-1(config-if)# ip pim sparse-mode

switch-vtep-1(config)# interface port-channel 10
switch-vtep-1(config-if)# vpc 10
switch-vtep-1(config-if)# switchport
switch-vtep-1(config-if)# switchport mode access
switch-vtep-1(config-if)# switchport access vlan 10
switch-vtep-1(config-if)# no shutdown
switch-vtep-1(config)# interface e1/1
switch-vtep-1(config-if)# channel-group 10 mode
active
switch-vtep-1(config-if)# no shutdown

switch-vtep-1(config)# interface nve1
switch-vtep-1(config-if)# no shutdown
switch-vtep-1(config-if)# source-interface loopback0
switch-vtep-1(config-if)# member vni 10000 mcast-group
230.1.1.1
```

```
switch-vtep-1(config)# vlan 10
switch-vtep-1(config-vlan)# vn-segment
10000 switch-vtep-1(config-vlan)# exit
```

- Inspur CN12900 VTEP-2 configuration:

```
switch-vtep-2(config)# feature nv overlay
switch-vtep-2(config)# feature vn-segment-vlan-
based
switch-vtep-2(config)# feature ospf
switch-vtep-2(config)# feature pim
switch-vtep-2(config)# router ospf 1
switch-vtep-2(config-router)# router-id
200.200.200.2
switch-vtep-2(config)# ip pim rp-address 10.1.1.1 group-list
224.0.0.0/4
switch-vtep-2(config)# interface loopback0
switch-vtep-2(config-if)# ip address 200.200.200.2/32
switch-vtep-2(config-if)# ip address 100.100.100.1/32
secondary
switch-vtep-2(config-if)# ip router ospf 1 area 0.0.0.0
switch-vtep-2(config-if)# ip pim sparse-mode
switch-vtep-2(config)# interface e2/1
switch-vtep-2(config-if)# ip address 20.1.1.5/30
switch-vtep-2(config-if)# ip router ospf 1 area 0.0.0.0
switch-vtep-2(config-if)# ip pim sparse-mode

switch-vtep-2(config)# interface port-channel 10
switch-vtep-2(config-if)# vpc 10
switch-vtep-2(config-if)# switchport
switch-vtep-2(config-if)# switchport mode access
switch-vtep-2(config-if)# switchport access vlan 10
switch-vtep-2(config-if)# no shutdown
switch-vtep-2(config)# interface e1/1
switch-vtep-2(config-if)# channel-group 10 mode
active
switch-vtep-2(config-if)# no shutdown

switch-vtep-2(config)# interface nve1
switch-vtep-2(config-if)# no shutdown
switch-vtep-2(config-if)# source-interface
loopback0
switch-vtep-2(config-if)# member vni 10000 mcast-group
230.1.1.1
switch-vtep-2(config)# vlan 10
switch-vtep-2(config-vlan)# vn-segment
10000
switch-vtep-2(config-vlan)# exit
```

- Inspur CN12900 VTEP-3 configuration:

```
switch-vtep-3(config)# feature nv overlay
switch-vtep-3(config)# feature vn-segment-vlan-
based
```

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

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

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

CHAPTER 4 Configuring VXLAN BGP EVPN

4.1 Information About VXLAN BGP EVPN

4.1.1 Guidelines and Limitations for VXLAN BGP EVPN

VXLAN BGP EVPN has the following guidelines and limitations:

- The following guidelines and limitations apply to VXLAN/VTEP:
- SPAN source or destination is supported on any port.
- Rx SPAN is supported. Tx or both (Tx and Rx) are not supported.
- SPAN Tx for VXLAN encapsulated traffic is not supported for the Layer 3 uplink interface.

For more information, see the Inspur CN12900 Series INOS-CN System Management Configuration Guide.

- 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.
- For the Inspur CN12904 and CN12908 with -R line cards, VXLAN Layer 2 Gateway is supported only on the CN129-X636C-R line card. VXLAN and MPLS cannot be enabled on the Inspur CN12908 switch at the same time.
- For Inspur CN12904 and CN12908 with -R line cards, if VXLAN is enabled, the Layer 2 Gateway cannot be enabled when there is any line card other than the CN129-X636C-R.
- You can configure EVPN over segment routing or MPLS. See the Inspur CN12900 Series INOS-CN Label Switching Configuration Guide for more information.
- You can use MPLS tunnel encapsulation using the CLI encapsulation mpls command. You can configure the label allocation mode for the EVPN address family.
- In VXLAN EVPN setup that has 2K VNI scale configuration, the control plane down time takes more than 200 seconds. To avoid BGP flap, configure the graceful restart time to 300 seconds.
- SVI and sub-interfaces as core links are not supported in multisite EVPN.
- In a VXLAN EVPN setup, border leaves must use unique route distinguishers, preferably using **auto rd** command. It is not supported to have same route distinguishers in different border leaves.
- ARP suppression is only supported for a VNI if the VTEP hosts the First-Hop Gateway (Distributed Anycast Gateway) for this VNI. The VTEP and the SVI for this VLAN have to be properly configured for the distributed anycast gateway operation, for example, global anycast gateway MAC address configured and anycast gateway feature with the virtual IP address on the SVI.
- The **show** commands with the **internal** keyword are not supported.
- DHCP snooping (Dynamic Host Configuration Protocol snooping) is not supported on VXLAN VLANs.
- SPAN TX for VXLAN encapsulated traffic is not supported for the Layer 3 uplink interface.
- RACLs are not supported on Layer 3 uplinks for VXLAN traffic. Egress VACLs support is not available for de-capsulated packets in the network to access direction on the inner payload.

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

See the Inspur CN12900 Series INOS-CN Security Configuration Guide for other guidelines and limitations for the VXLAN ACL feature.

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

See the Inspur CN12900 Series INOS-CN Quality of Service Configuration Guide for other guidelines and limitations for the VXLAN QoS feature.

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

- VTEP does not support Layer 3 subinterface uplinks that carry VXLAN encapsulated traffic.
- Layer 3 interface uplinks that carry VXLAN encapsulated traffic do not support subinterfaces for non-VXLAN encapsulated traffic.
- Non-VxLAN sub-interface VLANs cannot be shared with VXLAN VLANs.
- Subinterfaces on 40G (ALE) uplink ports are not supported on VXLAN VTEPs.
- Point to multipoint Layer 3 and SVI uplinks are not supported. Since both uplink types can only be enabled point-to-point, they cannot span across more than two switches.
- For EBGP, it is recommended to use a single overlay EBGP EVPN session between loopbacks.
- Bind NVE to a loopback address that is separate from other loopback addresses that are required by Layer 3 protocols. A best practice is to use a dedicated loopback address for VXLAN.
- VXLAN BGP EVPN does not support an NVE interface in a non-default VRF.
- It is recommended to configure a single BGP session over the loopback for an overlay BGP session.
- Changing the “System Routing Mode” requires a reload of the switch.
- When Inspur CN12900 platform switches are used as VTEPs, 100G line cards are not supported on Inspur CN12900 platform switches.
- The VXLAN UDP port number is used for VXLAN encapsulation. For Inspur INOS-CN, the UDP port number is 4789. It complies with IETF standards and is not configurable.
- 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 co-existence with the GRE tunnel feature or the MPLS (static or segment-routing) feature on Inspur CN12900 Series switches with a Network Forwarding Engine (NFE).
- Resilient hashing (port-channel load-balancing resiliency) and VXLAN configurations are not compatible with VTEPs using ALE uplink ports.

4.2 Configuring VXLAN BGP EVPN

4.2.1 Enabling VXLAN

Enable VXLAN and the EVPN.

Procedure

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

4.2.2 Configuring VLAN and VXLAN VNI

Procedure

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

4.2.3 Configuring VRF for VXLAN Routing

Configure the tenant VRF.

Procedure

	Command or Action	Purpose
Step 1	**vrf context** *vxlan*	Configure the VRF.
Step 2	**vni** *number*	Specify VNI.
Step 3	**rd auto**	Specify VRF RD (route distinguisher).
Step 4	**address-family ipv4 unicast**	Configure address family for IPv4.
Step 5	**route-target both auto**	**Note** Specifying the **auto** option is applicable only for IBGP. Manually configured route targets are required for EBGP.
Step 6	**route-target both auto evpn**	**Note** Specifying the **auto** option is applicable only for IBGP. Manually configured route targets are required for EBGP.
Step 7	**address-family ipv6 unicast**	Configure address family for IPv6.
Step 8	**route-target both auto**	**Note** Specifying the **auto** option is applicable only for IBGP. Manually configured route targets are required for EBGP.
Step 9	**route-target both auto evpn**	**Note** Specifying the **auto** option is applicable only for IBGP. Manually configured route targets are required for EBGP.

4.2.4 Configuring SVI for Hosts for VXLAN Routing

Configure the SVI for hosts.

Procedure

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

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

4.2.5 Configuring VRF for VXLAN Routing

Configure the tenant VRF.

Procedure

	Command or Action	Purpose
Step 1	**vrf context** *vxlan*	Configure the VRF.
Step 2	**vni** *number*	Specify VNI.
Step 3	**rd auto**	Specify VRF RD (route distinguisher).
Step 4	**address-family ipv4 unicast**	Configure address family for IPv4.
Step 5	**route-target both auto**	**Note** Specifying the **auto** option is applicable only for IBGP. Manually configured route targets are required for EBGP.
Step 6	**route-target both auto evpn**	**Note** Specifying the **auto** option is applicable only for IBGP. Manually configured route targets are required for EBGP.
Step 7	**address-family ipv6 unicast**	Configure address family for IPv6.
Step 8	**route-target both auto**	**Note** Specifying the **auto** option is applicable only for IBGP. Manually configured route targets are required for EBGP.
Step 9	**route-target both auto evpn**	**Note** Specifying the **auto** option is applicable only for IBGP. Manually configured route targets are required for EBGP.

4.2.6 Configuring VNI Under VRF for VXLAN Routing

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

Procedure

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

4.2.7 Configuring Anycast Gateway for VXLAN Routing

Procedure

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

4.2.8 Configuring the NVE Interface and VNIs

Procedure

	Command or Action	Purpose
Step 1	**interface** *nve-interface*	Configure the NVE interface.
Step 2	**host-reachability protocol bgp**	This defines BGP as the mechanism for host reachability advertisement
Step 3	**member vni** *vni* **associate-vrf**	Add Layer-3 VNIs, one per tenant VRF, to the overlay. **Note** Required for VXLAN routing only.
Step 4	**global mcast-group** *ip-address* **{L2 | L3}**	Configures the mcast group on a per-NVE interface basis. This applies to all Layer 2 VNIs.
Step 5	**member vni** *vni*	Add Layer 2 VNIs to the tunnel interface.
Step 6	**mcast-group** *address*	Configure the mcast group on a per-VNI basis

4.2.9 Configuring BGP on the VTEP

Procedure

	Command or Action	Purpose
Step 1	**router bgp** *number*	Configure BGP.
Step 2	**router-id** *address*	Specify router address.
Step 3	**neighbor** *address* **remote-as** *number*	Define MP-BGP neighbors. Under each neighbor define l2vpn evpn.
Step 4	**address-family ipv4 unicast**	Configure address family for IPv4.
Step 5	**address-family l2vpn evpn**	Configure address family Layer 2 VPN EVPN under the BGP neighbor. **Note** Address-family ipv4 evpn for vxlan host-based routing
Step 6	(Optional) **Allowas-in**	Allows duplicate AS numbers in the AS path. Configure this parameter on the leaf for eBGP when all leafs are using the same AS, but the spines have a different AS than leafs.
Step 7	**send-community extended**	Configures community for BGP neighbors.
Step 8	**vrf** *vrf-name*	Specify VRF.
Step 9	**address-family ipv4 unicast**	Configure address family for IPv4.
Step 10	**advertise** *l2vpn* **evpn**	Enable advertising EVPN routes.
Step 11	**address-family ipv6 unicast**	Configure address family for IPv6.
Step 12	**advertise** *l2vpn* **evpn**	Enable advertising EVPN routes.

4.2.10 Configuring RD and Route Targets for VXLAN Bridging

Procedure

	Command or Action	Purpose
Step 1	**evpn**	Configure VRF.
Step 2	**vni** *number* **l2**	**Note** Only Layer 2 VNIs need to be specified.
Step 3	**rd auto**	Define VRF RD (route distinguisher) to configure VRF context.

	Command or Action	Purpose
Step 4	**route-target import auto**	Define VRF Route Target and import policies.
Step 5	**route-target export auto**	Define VRF Route Target and export policies.

4.2.11 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.

Before you begin

The following are required before configuring VXLAN EVPN ingress replication:

- 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** *nve-interface*	Configure the NVE interface.
Step 2	**host-reachability protocol bgp**	This defines BGP as the mechanism for host reachability advertisement
Step 3	**member vni** *vni* **associate-vrf**	Add Layer-3 VNIs, one per tenant VRF, to the overlay. **Note** Required for VXLAN routing only.
Step 4	**member vni** *vni*	Add Layer 2 VNIs to the tunnel interface.
Step 5	**ingress-replication protocol bgp**	Enables the VTEP to exchange local and remote VTEP IP addresses on the VNI in order to create the ingress replication list. This enables sending and receiving BUM traffic for the VNI. **Note** Using **ingress-replication protocol bgp** avoids the need for any multicast configurations that might have been required for configuring the underlay.

4.2.12 Configuring BGP for EVPN on the Spine

Procedure

	Command or Action	Purpose

Step 1	**route-map permitall permit 10**	Configure route-map. **Note** The route-map keeps the next-hop unchanged for EVPN routes. • Required for eBGP. • Optional for iBGP.
Step 2	**set ip next-hop unchanged**	Set next-hop address. **Note** The route-map keeps the next-hop unchanged for EVPN routes. • Required for eBGP. • Optional for iBGP. **Note** When two next hops are enabled, next hop ordering is not maintained. If one of the next hops is a VXLAN next hop and the other next hop is local reachable via FIB/AM/Hmm, the local next hop reachable via FIB/AM/Hmm is always taken irrespective of the order. Directly/locally connected next hops are always given priority over remotely connected next hops.
Step 3	**router bgp** *autonomous system number*	Specify BGP.
Step 4	**address-family l2vpn evpn**	Configure address family Layer 2 VPN EVPN under the BGP neighbor.
Step 5	**retain route-target all**	Configure retain route-target all under address-family Layer 2 VPN EVPN [global]. **Note** Required for eBGP. Allows the spine to retain and advertise all EVPN routes when there are no local VNI configured with matching import route targets.
Step 6	**neighbor** *address* **remote-as** *number*	Define neighbor.
Step 7	**address-family l2vpn evpn**	Configure address family Layer 2 VPN EVPN under the BGP neighbor.
Step 8	**disable-peer-as-check**	Disables checking the peer AS number during route advertisement. Configure this parameter on the spine for eBGP when all leafs are using

	Command or Action	Purpose
Step 1	**route-map permitall permit 10**	Configure route-map. the same AS but the spines have a different AS than leafs. Note Required for eBGP.
Step 9	**send-community extended**	Configures community for BGP neighbors.
Step 10	**route-map permitall out**	Applies route-map to keep the next-hop unchanged. Note Required for eBGP.

4.2.13 Suppressing ARP

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

Procedure

	Command or Action	Purpose
Step 1	**hardware access-list tcam region arp-ether** *size* **double-wide**	Configure TCAM region to suppress ARP. *tcam-size*—TCAM size. The size has to be a multiple of 256. If the size is more than 256, it has to be a multiple of 512. Note Reload is required for the TCAM configuration to be in effect.
Step 2	**interface nve 1**	Create the network virtualization endpoint (NVE) interface.
Step 3	**member vni** *vni-id*	Specify VNI ID.
Step 4	**suppress-arp**	Configure to suppress ARP under Layer 2 VNI.
Step 5	**copy running-config start-up-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

4.2.14 Disabling VXLANs

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters configuration mode.
Step 2	**no nv overlay evpn**	Disables EVPN control plane.

	Command or Action	Purpose
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.

4.3 Duplicate Detection for IP and MAC Addresses

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

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

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

```
2018 Oct 1 10:27:34 leaf %$ VDC-1 %$ %USER-2-SYSTEM_MSG: Unfreeze limit (3) hit, MAC
 0000.0033.3333in topo: 200 is permanently frozen - l2rib
2018 Oct 1 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
2018 Oct 1 10:27:34 leaf %$ VDC-1 %$ %USER-2-SYSTEM_MSG: Unfreeze limit (3) hit, MAC
 0000.0033.3334in topo: 200 is permanently frozen - l2rib
2018 Oct 1 10:27:34 leaf %$ VDC-1 %$ %USER-2-SYSTEM_MSG: Detected duplicate host
0000.0033.3334, topology 200, during Local update, with host l
```

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

Command	Description

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

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

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

Command	Description

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

switch(config)# l2rib dup-host-mac-detection 100 ? <2-36000>	The duplicate detection timeout in seconds for the number of host moves. The range is 2 to 36000 seconds; default is 180 seconds.
switch(config)# l2rib dup-host-mac-detection 100 10	Detects duplicate host addresses (limited to 100 moves) in a period of 10 seconds.

4.3.1 Enabling Nuage Controller Interoperability

The following steps enable Nuage controller interoperability.

Procedure

	Command or Action	Purpose
Step 1	**nuage controller interop**	Global command to enable interoperability mode.

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

Example

The following is an example to enable Nuage controller interoperability:

```
/*** Enable interoperability mode at global level. ***/
switch(config)# nuage controller interop
/*** Configure BGP to enable interoperability mode. ***/
switch(config)# router bgp 1001
switch(config-router)# address-family l2vpn
evpn
switch(config-router-af)# advertise-system-mac
switch(config-router-af)# allow-vni-in-ethertag
/*** Advertise host tenant routes as evpn type-5 routes for interoperability. ***/
switch(config)# route-map permitall permit 10
switch(config)# router bgp 1001
switch(config-router)# vrf vni-491830
switch(config-router-vrf)# address-family ipv4
unicast
switch(config-router-vrf-af)# advertise l2vpn evpn
switch(config-router-vrf-af)# redistribute hmm route-map permitall
```

4.4 Verifying the VXLAN BGP EVPN Configuration

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

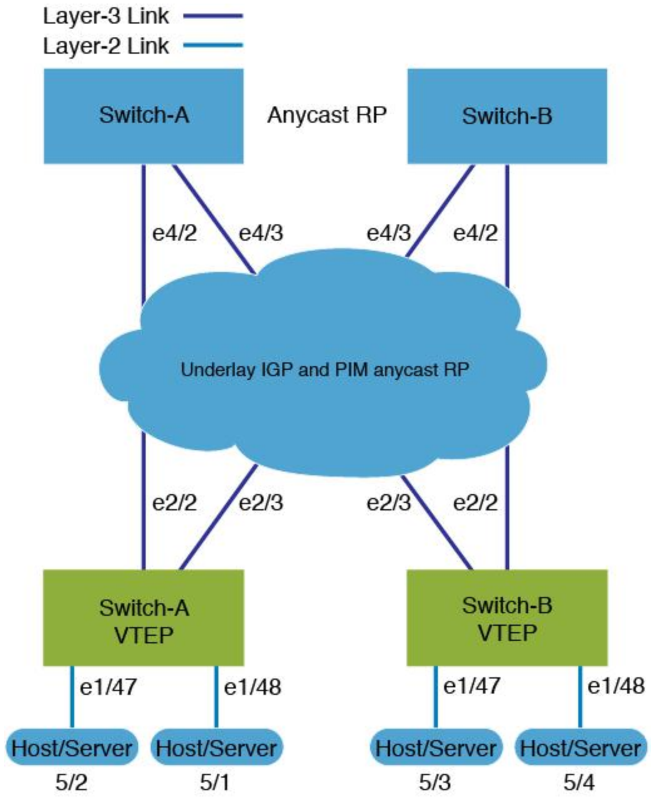
Command	Purpose
show nve vrf	Displays VRFs and associated VNIs

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

4.5 Example of VXLAN BGP EVPN (EBGP)

An example of a VXLAN BGP EVPN (EBGP):

Figure 9：VXLAN BGP EVPN Topology (EBGP)



EBGP between Spine and Leaf

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

```
nv overlay evpn
```

- Enable the relevant protocols

```
feature bgp
feature pim
```

- Configure Loopback for local VTEP IP, and BGP

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

- Configure Loopback for Anycast RP

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

- Configure Anycast RP

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

- Configure route-map used by EBGP for Spine

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

- Configure interfaces for Spine-leaf interconnect

```
interface Ethernet4/2
  ip address 192.168.1.42/24
  ip pim sparse-mode
  no shutdown
interface Ethernet4/3
  ip address 192.168.2.43/24
  ip pim sparse-mode
  no shutdown
```

- Configure the BGP overlay for the EVPN address family.

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


neighbor 40.1.1.1 remote-as 200
  update-source loopback0 ebgp-
  multihop 3 address-family
  l2vpn evpn
    disable-peer-as-check
    send-community extended
    route-map permitall out
```

- Configure the BGP underlay.

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

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

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

- Configure Anycast RP

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

- Configure interfaces for Spine-leaf interconnect

```
interface Ethernet4/2

  ip address 192.168.4.42/24
  ip pim sparse-mode
  no shutdown

interface Ethernet4/3

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

- Configure Loopback for local VTEP IP, and BGP

```
interface loopback0

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

- Configure Loopback for Anycast RP

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

- Configure the BGP overlay for the EVPN address family.

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

```
                        200
     ebgp-multihop 3 address-family l2vpn evpn
       disable-peer-as-check send-community
       extended route-map permitall out
```

- Configure the BGP underlay.

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

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

```
nv overlay evpn
```

- Enable the relevant protocols

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

- Configure DHCP relay for Tenant VRFs

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

- Enable VXLAN with distributed anycast-gateway using BGP EVPN

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

- Enable PIM RP

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

- Configure Loopback for BGP

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

- Configure Loopback for local VTEP IP

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

- Configure interfaces for Spine-leaf interconnect

```
                  interface Ethernet2/2
                    no switchport
```

```
  load-interval  counter  1  5
  ip address 192.168.1.22/24
  ip pim sparse-mode
  no shutdown
interface Ethernet2/3
  no switchport
  load-interval  counter  1  5
  ip address 192.168.3.23/24
  ip pim sparse-mode
  no shutdown
```

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

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

- Configure VRF overlay VLAN/SVI for the VRF

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

- Create VLAN and provide mapping to VXLAN

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

- Create VRF and configure VNI

```
vrf context vxlan-900001
```

```
  vni 900001

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

- Create server facing SVI and enable distributed anycast-gateway

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

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

- Configure ACL TCAM region for ARP suppression

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

`

Option 1

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

Option 2

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

- Configure interfaces for hosts/servers.

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

- Configure BGP

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

  advertise l2vpn evpn

evpn
 vni 2001001 l2
 vni 2001002 l2

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


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

advertise l2vpn evpn

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

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

- Enable VXLAN with distributed anycast-gateway using BGP EVPN

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

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

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

- Create VLAN and provide mapping to VXLAN

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

- Create VRF and configure VNI

```
vrf context vxlan-
  900001 vni 900001

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

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

- Configure ACL TCAM region for ARP suppression

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

- Configure internal control VLAN/SVI for the VRF

```
interface Vlan1

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

- Create server facing SVI and enable distributed anycast-gateway

```
interface Vlan1001 no shutdown
  vrf member vxlan-900001 ip address 4.1.1.1/24 ipv6
  address 4:1:0:1::1/64
  fabric forwarding mode anycast-gateway
interface Vlan1002 no shutdown
  vrf member vxlan-900001 ip address 4.2.2.1/24 ipv6
  address 4:2:0:1::1/64
  fabric forwarding mode anycast-gateway
```

- Create the network virtualization endpoint (NVE) interface

Option 1

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



                        mcast-group 225.1.1.1
```

Option 2

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

- Configure interfaces for hosts/servers

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

- Configure interfaces for Spine-leaf interconnect

```
interface Ethernet2/1

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

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

- Configure Loopback for BGP

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

- Configure Loopback for local VTEP IP

```
interface loopback

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

- Configure BGP

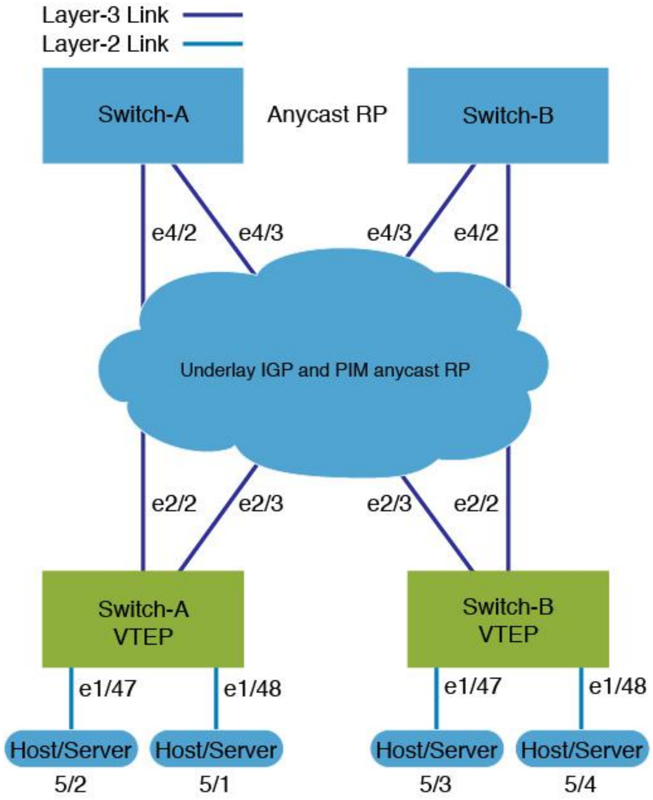
```
router bgp 200
router-id 40.1.1.1
  neighbor 10.1.1.1 remote-as
    100 update-source loopback0
    ebgp-multihop 3
      allowas-in send-
      community extended
    address-family l2vpn
      allowas-in send-
      community extended
  neighbor 20.1.1.1 remote-as
    100 update-source loopback0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

advertise l2vpn evpn

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
```

4.6 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 (Switch-A)
- Enable the EVPN control plane

```
nv overlay evpn
```

- Enable the relevant protocols

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

- Configure Loopback for local VTEP IP, and BGP

```
interface loopback0
  ip address 10.1.1.1/32
  ip router ospf 1 area
```

```
0.0.0.0 ip pim sparse-mode
```

- Configure Loopback for Anycast RP

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

- Configure Anycast RP

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

- Enable OSPF for underlay routing

```
router ospf 1
```

- Configure interfaces for Spine-leaf interconnect

```
interface Ethernet4/2
  ip address 192.168.1.42/24
  ip router ospf 1 area
  0.0.0.0 ip pim sparse-mode
  no shutdown
interface Ethernet4/3
  ip address 192.168.2.43/24
  ip router ospf 1 area
  0.0.0.0 ip pim sparse-mode
  no shutdown
```

- Configure BGP

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

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

```
feature telnet
feature nxapi
```

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

- Configure Anycast RP

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

  pim rp-candidate loopback1 group-list 225.0.0.0/8


ip pim ssm range 232.0.0.0/8
ip pim anycast-rp 100.1.1.1 10.1.1.1
ip pim anycast-rp 100.1.1.1 20.1.1.1
vlan 1-1002
```

- Configure interfaces for Spine-leaf interconnect

```
interface Ethernet4/2
  ip address 192.168.4.42/24
  ip router ospf 1 area
  0.0.0.0 ip pim sparse-mode
  no shutdown
interface Ethernet4/3
  ip address 192.168.3.43/24
  ip router ospf 1 area
  0.0.0.0 ip pim sparse-mode
  no shutdown
```

- Configure Loopback for local VTEP IP, and BGP

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

- Configure Loopback for Anycast RP

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

- Enable OSPF for underlay routing

```
router ospf 1
```

- Configure BGP

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

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

```
nv overlay evpn
```

- Enable the relevant protocols

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

- Enable VxLAN with distributed anycast-gateway using BGP EVPN

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

- Enabling OSPF for underlay routing

```
router ospf 1
```

- Configure Loopback for local VTEP IP, and BGP

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

- Configure interfaces for Spine-leaf interconnect

```
interface Ethernet2/2
  no switchport
  ip address 192.168.1.22/24
  ip router ospf 1 area
  0.0.0.0 ip pim sparse-mode
  no shutdown
interface Ethernet2/3
  no switchport
  ip address 192.168.3.23/24
  ip router ospf 1 area
  0.0.0.0 ip pim sparse-mode
  no shutdown
```

- Configure PIM RP

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

- Create overlay VRF VLAN and configure vn-segment

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

- Configure VRF overlay VLAN/SVI for the VRF

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

- Create VLAN and provide mapping to VXLAN

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

- Create VRF and configure VNI

```
vrf context vxlan-
  900001 vni 900001

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

- Create server facing SVI and enable distributed anycast-gateway

```
interface Vlan1001
  no shutdown
  vrf member vxlan-900001
  ip address 4.1.1.1/24
  ipv6 address 4:1:0:1::1/64
  fabric forwarding mode anycast-gateway
interface Vlan1002
  no shutdown
  vrf member vxlan-900001
  ip address 4.2.2.1/24
  ipv6 address 4:2:0:1::1/64
  fabric forwarding mode anycast-gateway
```

- Configure ACL TCAM region for ARP suppression

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

Create the network virtualization endpoint (NVE) interface

Option 1

```
interface nve1
  no shutdown


    source-interface loopback0
    host-reachability protocol bgp
    member vni 900001 associate-vrf
    member vni 2001001
      suppress-arp mcast-
      group 225.4.0.1
    member vni 2001002
      suppress-arp mcast-
      group 225.4.0.1
```

Option 2

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

- Configure interfaces for hosts/servers

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

- Configure BGP

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

```
    vni 2001001 l2
    vni 2001002 l2

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

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

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

- Enable VxLAN with distributed anycast-gateway using BGP EVPN

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

- Configure PIM RP

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

- Create overlay VRF VLAN and configure vn-segment

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

- Create VLAN and provide mapping to VXLAN

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

- Create VRF and configure VNI

```
vrf context vxlan-900001
  vni 900001

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

- Configure ACL TCAM region for ARP suppression

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

- Configure internal control VLAN/SVI for the VRF

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

- Create server facing SVI and enable distributed anycast-gateway

```
interface Vlan1001
  no shutdown
  vrf member vxlan-900001 ip
  address 4.1.1.1/24 ipv6
  address 4:1:0:1::1/64
  fabric forwarding mode anycast-gateway
interface Vlan1002
  no shutdown
  vrf member vxlan-900001 ip
  address 4.2.2.1/24 ipv6
  address 4:2:0:1::1/64
  fabric forwarding mode anycast-gateway
```

Create the network virtualization endpoint (NVE) interface

Option 1

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

Option 2

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

- Configure interfaces for hosts/servers

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

- Configure interfaces for Spine-leaf interconnect

```
interface Ethernet2/1

interface Ethernet2/2
  no switchport
  ip address 192.168.4.22/24
  ip router ospf 1 area
  0.0.0.0 ip pim sparse-mode
  no shutdown
interface Ethernet2/3
  no switchport
  ip address 192.168.2.23/24
  ip router ospf 1 area
  0.0.0.0 ip pim sparse-mode
  no shutdown
```

- Configure Loopback for local VTEP IP, and BGP

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

- Enabling OSPF for underlay routing

```
    router ospf 1
```

- Configure BGP

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

4.7 Example Show Commands

- **show nve peers**

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

- **show nve vni**

```
switch-B# show nve vni
Codes: CP - Control Plane        DP - Data Plane
       UC - Unconfigured         SA - Suppress ARP
Interface VNI      Multicast-group   State Mode Type [BD/VRF]      Flags
--------- -------- ----------------- ----- ---- ------------------ -----
nve1      900001   n/a               Up    CP   L3 [vxlan-900001]
nve1      2001001  225.4.0.1         Up    CP   L2 [1001]          SA
nve1      2001002  225.4.0.1         Up    CP   L2 [1002]          SA
```

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

```
      switch-B# show ip arp suppression-cache
                                      detail
Flags: + - Adjacencies synced via CFSoE
       L - Local Adjacency
       R - Remote Adjacency
       L2 - Learnt over L2 interface

Ip Address      Age       Mac Address   Vlan Physical-ifindex    Flags
```

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

- **show vxlan interface**

```
switch-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**

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

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

- **show bgp l2vpn evpn**

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


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

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

  Route Distinguisher: 40.1.1.1:33768     (L2VNI 2001001)
  *>i[2]:[0]:[0]:[48]:[d8b1.9071.e903]:[0]:[0.0.0.0]/216
                      30.1.1.1                          100          0 i
```

```
*>l[2]:[0]:[0]:[48]:[f8c2.8890.2a45]:[0]:[0.0.0.0]/216
                      40.1.1.1                           100      32768 i
*>i[2]:[0]:[0]:[48]:[d8b1.9071.e903]:[32]:[4.1.1.12]/272
                      30.1.1.1                           100          0 i
*>l[2]:[0]:[0]:[48]:[f8c2.8890.2a45]:[32]:[4.1.1.122]/272

                      40.1.1.1                           100      32768 i

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

                      40.1.1.1                           100      32768 i

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

                      30.1.1.1                           100          0 i
```

- **show l2route evpn mac all**

```
switch-B# show l2route evpn mac
all
Flags -(Rmac):Router MAC (Stt):Static (L):Local (R):Remote (V):vPC link
(Dup):Duplicate (Spl):Split (Rcv):Recv (AD):Auto-Delete (D):Del Pending
(S):Stale (C):Clear, (Ps):Peer Sync (O):Re-Originated (Nho):NH-Override

(Pf):Permanently-Frozen

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

1002        f8c2.8890.2a45 Local  L,            0          Eth1/47
```

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

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

1002        f8c2.8890.2a45 HMM    --            0          4.2.2.111      Local

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
```

4.8 Configuring ESI ARP Suppression

4.8.1 Overview of ESI ARP Suppression

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

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

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

ESI ARP suppression is a per-VNI (L2-VNI) feature. ESI ARP suppression is supported in both L2 (no SVI) and L3 modes. Only L3 mode is supported.

The ESI ARP suppression cache is built by:

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

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

4.8.2 Limitations for ESI ARP Suppression

See the following limitations for ESI ARP suppression:

- ESI

4.8.3 Configuring ESI ARP Suppression

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

```
Interface
   nve1 no
   shutdow
   n
```

```
source-interface loopback1
host-reachability protocol
bgp member vni 10000
    suppress-arp
mcast-group
224.1.1.10
```

4.8.4 Displaying Show Commands for ESI ARP Suppression

See the following Show commands output for ESI ARP suppression:

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

                switch# show ip arp suppression-cache local

                Flags: + - Adjacencies synced via CFSoE
                       L - Local Adjacency
                       R - Remote Adjacency
                       L2 - Learnt over L2 interface
                      PS - Added via L2RIB, Peer Sync
                      RO - Dervied from L2RIB Peer Sync Entry
Ip Address              Age                     Mac Address    Vlan Physical-ifindex  Flags    Remote

                 Vtep Addrs
                 61.1.1.20      00:07:54 0000.0610.0020  610 port-channel20     L
                 61.1.1.30      00:07:54 0000.0610.0030  610 port-channel2       L[PS RO]
                 61.1.1.10      00:07:54 0000.0610.0010  610 Ethernet1/96        L
                   switch# show arp suppression-cache
                             ip remote
                          - Adjacencies synced via
                  Flags: +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                                         Physical-
                          Ip Address     Age      Mac Address        Vlan    ifindex            Flags
                    Remote Vtep Addrs
                                00:48:37
                 61.1.1.40      0000.0610.0040           610    (null)                    R
                 VTEP1,
                 VTEP2..       VTEPn

                   switch# show arp suppression-cache
                             ip detail
                       Flags: + - Adjacencies synced via CFSoE
                        L - Local
                        Adjacency
                        R - Remote Adjacency
                         L2- Learnt over L2 interface
```

```
      PS - Added via L2RIB, Peer Sync
      RO - Derived from L2RIB Peer Sync Entry
             Ip Address     Age      Mac Address    Vlan Physical-ifindex    Flags
    Remote Vtep Addrs
61.1.1.20       00:00:07 0000.0610.0020  610 port-channel20     L
61.1.1.30       00:00:07 0000.0610.0030  610 port-channel2       L[PS RO]
61.1.1.10       00:00:07 0000.0610.0010  610 Ethernet1/96        L
61.1.1.40       00:00:07 0000.0610.0040  610 (null)                    R
VTEP1, VTEP2.. VTEPn


switch# show ip arp suppression-cache summary
 IP ARP suppression-cache Summary
 Remote          :1
 Local              :3
 Total               :4
switch# show ip arp suppression-cache statistics
 ARP packet statistics for suppression-cache
 Suppressed:
 Total 0, Requests 0, Requests on L2 0, Gratuitous 0, Gratuitous on L2 0
 Forwarded :
 Total: 364
  L3 mode :      Requests 364, Replies 0
  Request on core port 364, Reply on core port 0
                 Dropped 0
  L2 mode :      Requests 0, Replies 0
                 Request on core port 0, Reply on core port 0
                 Dropped 0
 Received:
 Total: 3016    Requests 376, Replies
 L3 mode:       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
============                         :1756 | 1756:Processed ADD from L2RIB Receieved DEL
 Receieved ADD from L2RIB                                                          from
L2RIB       :88 | 87:Processed DEL from L2RIB Receieved PC shut from L2RIB    :0 |
1755:Processed PC shut from L2RIB Receieved remote UPD from L2RIB :5004 | 0:Processed remote
 UPD from L2RIB
ERRORS
=======
Multihoming ADD error invalid flag         :0
Multihoming DEL error invalid flag         :0
Multihoming ADD error invalid current state:0
Multihoming DEL error invalid current state:0
Peer sync DEL error MAC mismatch           :0
Peer sync DEL error second delete          :0
Peer sync DEL error deleteing TL route     :0
True local DEL error deleteing PS RO route :0
switch#
```

CHAPTER 5 Configuring VIP/PIP

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

The **advertise-pip** command lets BGP use the PIP as next-hop when advertising prefix routes or leaf-generated routes if vPC is enabled.

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.

5.2 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.

5.3 DHCP Configuration in a vPC Setup

When DHCP or DHCPv6 relay function is configured on leaf switches in a vPC setup, and the DHCP server is in the non default, non management VRF, then configure the **advertise-pip** command on the vPC leaf switches. This allows BGP EVPN to advertise Route-type 5 routes with the next-hop using the primary IP address of the VTEP interface.

The following is a sample configuration:

```
switch(config)# router bgp 100
  address-family l2vpn evpn
    advertise-pip
interface nve 1
  advertise virtual-rmac
```

5.4 IP Prefix Advertisement in vPC Setup

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

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

On a vPC enabled leaf or border leaf switch, by default all Layer-3 routes are advertised with the secondary IP address (VIP) of the leaf switch VTEP as the BGP next-hop IP address. Prefix routes and leaf switch generated routes are not synced between vPC leaf switches. Using the VIP as the BGP next-hop for these types of routes can cause traffic to be forwarded to the wrong vPC leaf or border leaf switch and black-holed.

The provision to use the primary IP address (PIP) as the next-hop when advertising prefix routes or loopback interface routes in BGP on vPC enabled leaf or border leaf switches allows users to select the PIP as BGP next-hop when advertising these types of routes, so that traffic is always forwarded to the right vPC enabled leaf or border leaf switch.

The configuration command for advertising the PIP is **advertise-pip** .

The following is a sample configuration:

```
switch(config)# router bgp 100
  address-family 12vpn evpn
    advertise-pip
  interface nve 1
  advertise virtual-rmac
```

The **advertise-pip** command lets BGP use the PIP as next-hop when advertising prefix routes or leaf generated routes if vPC is enabled.

CHAPTER 6 DHCP Relay in VXLAN BGP EVPN

6.1 DHCP Relay in VXLAN BGP EVPN Overview

DHCP relay is supported by VXLAN BGP EVPN and is useful in a multi-tenant VXLAN EVPN deployment to provision DHCP service to EVPN tenant clients.

In a multi-tenant EVPN environment, DHCP relay uses the following sub-options of Option 82:

- Sub-option 151(0x97) - Virtual Subnet Selection (Defined in RFC#6607.)

Used to convey VRF related information to the DHCP server in an MPLS-VPN and VXLAN EVPN multi-tenant environment.

- Sub-option 11(0xb) - Server ID Override (Defined in RFC#5107.)

The server identifier (server ID) override sub-option allows the DHCP relay agent to specify a new value for the server ID option, which is inserted by the DHCP server in the reply packet. This sub-option allows the DHCP relay agent to act as the actual DHCP server such that the renew requests will come to the relay agent rather than the DHCP server directly. The server ID override sub-option contains the incoming interface IP address, which is the IP address on the relay agent that is accessible from the client. Using this information, the DHCP client sends all renew and release request packets to the relay agent. The relay agent adds all of the appropriate sub-options and then forwards the renew and release request packets to the original DHCP server. For this function, Inspur's proprietary implementation is sub-option 152(0x98). You can use the **ip dhcp relay sub-option type** 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, Inspur's proprietary implementation is sub-option 150(0x96). You can use the **ip dhcp relay sub-option type** command to manage the function.

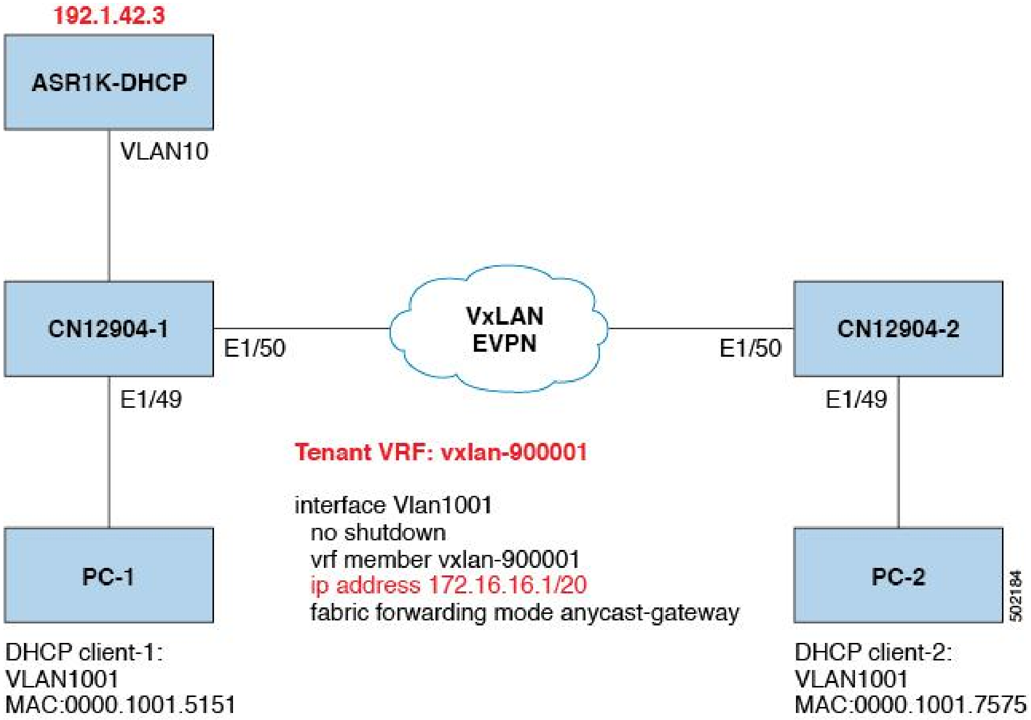
6.2 Guidelines and Limitations for DHCP Relay

The following are the guidelines and limitations for DHCP Relay in VXLAN BGP EVPN:

- Beginning in Inspur INOS-CN Release 9.2(2), support is added for Inspur CN12904 and CN12908 switches with -R line cards.
- For all Inspur CN12904 and CN12908 switches with -R line cards, IPv4 DHCP support is added.

6.3 DHCP Relay in VXLAN BGP EVPN Example

Figure 11: Example Topology



Topology characteristics:

- Switches CN12904-1 and CN12904-2 are VTEPs connected to VXLAN fabric.
- Client1 and client2 are DHCP clients in vlan1001. They belong to tenant VRF vxlan-900001.
- The DHCP server is ASR1K, a router that sits in vlan10.
- DHCP server configuration

```
ip vrf vxlan900001
ip dhcp excluded-address vrf vxlan900001 172.16.16.1 172.16.16.9
ip dhcp pool one
 vrf vxlan900001
 network 172.16.16.0 255.255.240.0
 defaultrouter 172.16.16.1
```

6.4 Basic VXLAN BGP EVPN Configuration

• CN12904-1
Option 1

```
interface nve1
  no shutdown
  source-interface loopback1
  host-reachability protocol bgp
  member vni 10000 associate-vrf
```

```
mcast-group 224.1.1.1
member vni 10001 associate-vrf
mcast-group 224.1.1.1
member vni20000
suppress-arp mcast-
group 225.1.1.1 member
vni 20001 suppress-arp
mcast-group 225.1.1.1
```

Option 2

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


    interfaca Ethernetl/49
      switchport mode trunk
      switchport trunk allowed vlan 10,1001
      spanning—tree port type edge trunk
    interface Ethernetl/50
      no switchport
      ip address 192.1.33.2/24
      ip router ospf 1 area 0.0.0.0
      ip pire sparse-mode

  no shutdown

interface loopback0
  ip address 1.1.1.1/32
  ip router ospf 1 area
  0.0.0.0 ip pim sparse—mode
interface loopbackl
  vrf member vxlan—900001 ip
  address 11.11.11.11/32
router bgp 65535
  router—id 1.1.1.1
  log—neighbor-changes
  neighbor 2.2.2.2 remote—as
    65535 update—source loopback0
    address-family l2vpn evpn
      send-community
  both vrf vxlen—900001
    address—family ipv4 unicast
    network 11.11.11.11/32
    network 192.1.42.0/24
    advertise l2vpn evpn
evpn
  vni 2001001 12

rd auto
    route—target import auto
    route—target export auto
```

- CN12904-2

```
version 9.2(2)
hostname
CN12904—1
nv overlay evpn
feature vn—segment—vlan—based
feature nv overlay
fabric forwarding anycast—gateway—mac 0000.1111.2222

vlan 101
  vn—segment 900001
vlan 1001
  vn—segment 2001001
vrf context vxlan—90000l
  vni 900001
  rd auto
  address—family ipv4 unicast
    route-target both auto
    route—target both auto evpn
interface VianlOl

  no shutdown
  vrf member vxlan-
  900001 ip forward
  interface Vlanl00l
    no shutdown
    vrf member vxlan—900001 ip
    address 172.16.16.1/20
    fabric forwarding mcde anycast—gateway

  rd auto
    address—family ipv4 unicast
      route-target both auto
      route—target both auto evpn

  interface VianlOl
  no shutdown
  vrf member vxlan-
  900001 ip forward
  interface Vlanl00l
    no shutdown
    vrf member vxlan—900001 ip
  address 172.16.16.1/20
  fabric forwarding mcde anycast—gateway
```

Option 1

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

Option 2

```
 interface nve1
  no shutdown

 source-interface loopback 1
 host-reachibility protocol
 bgp global suppress-arp
 global mcast-group 224.1.1.1 L3
 global mcast-group 255.1.1.1 L2
 member vni 10000 associate-vrf
 member vni 10001 associate-vrf
 member vni 10002 associate-vrf
 member vni 10003 associate-vrf
 member vni 10004 associate-vrf
 member vni 10005 associate-vrf
 member vni 20000
 member vni 20001
 member vni 20002
 member vni 20003
 member vni 20004
 member vni 20005


interface Ethernetl/49
 switchport mode trunk
 switchport trunk alluwed vlan 10,1001
 spanning—tree port type edge trunk
interface Ethernetl/50
 no switchport
 ip address 192.1.34.2/24
 ip router ospf 1 area
 0.0.0.0 ip pim sparse-mode
 no shutdown
interface loopback0
 ip address 2.2.2.2/32
```

```
  ip router ospf 1 area
  0.0.0.0 ip pim sparse—mode
interface loopbackl
  vrf member vxlan—900001 ip
  address 22.22.22.22/32
router bgp 65535
  router—id 2.2.2.2
  log—neighbor-changes
  neighbor 1.1.1.1 remote—as
    65535 update—source loopback0
    address-family l2vpn evpn
      send-community
  both vrf vxlen—900001
    address—family ipv4 unicast
    network 22.22.22.22/32
    advertise l2vpn evpn
evpn
  vni 2001001 12

    rd auto
        route—target import auto
        route—target export auto
```

6.5 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.

6.5.1 Client on Tenant VRF and Server on Layer 3 Default VRF

Put DHCP server (192.1.42.3) into the default VRF and make sure it is reachable from both switch-1 and switch-2 through the default VRF.

```
switch-1# sh run int vl 10

!Command: show running-config interface Vlan10
!Time: Mon Oct 1 07:51:16 2018
version 9.2(2)

interface Vlan10
  no shutdown
  ip address 192.1.42.1/24
  ip router ospf 1 area 0.0.0.0
switch-1# ping 192.1.42.3 cou 1

PING 192.1.42.3 (192.1.42.3): 56 data bytes
64 bytes from 192.1.42.3: icmp_seq=0 ttl=254 time=0.593
ms - 192.1.42.3 ping statistics -
```

```
1 packets transmitted, 1 packets received, 0.00% packet
loss roundtrip min/avg/max = 0.593/0.592/0.593 ms

switch-2# ping 192.1.42.3 cou 1

PING 192.1.42.3 (192.1.42.3): 56 data bytes
64 bytes from 192.1.42.3: icmp_seq=0 ttl=252 time=0.609
ms - 192.1.42.3 ping statistics -
1 packets transmitted, 1 packets received, 0.00% packet
loss round-trip min/avg/max = 0.609/0.608/0.609 ms
```

DHCP Relay Configuration

- switch-1

```
    switch—1# sh run dhcp

!Command: show running—config dhcp

!Time: Mon Oct 1 08:26:00 2018

version 9.2(2)

I1(3) feature dhcp

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

interface Vlanl001
  ip dhcp relay address 192.1.42.3 use—vrf default
```

- switch-2

```
switch-2# sh run dhcp

!Command: show running—config dhcp

!Time: Mon Oct 1 08:26:16 2018

version 9.2(2) feature

dhcp

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

 ipv6 dhcp relay
 interfaoe Vlanl001
   ip dhcp relay address 192.1.42.3 use—vrf default
```

Debug Output

- The following is a packet dump for DHCP interact sequences.

```
switch-1# ethanalyzer local interface inband display-filter

"udp.srcport==67 or udp.dstport==67" limit-captured frames 0

Capturing on inband

20150824 08:35:25.066530 0.0.0.0 -> 255.255.255.255 DHCP DHCP Discover - Transaction
ID 0x636a38fd
20150824 08:35:25.068141 192.1.42.1 -> 192.1.42.3 DHCP DHCP Discover - Transaction
ID 0x636a38fd
```

```
20150824 08:35:27.069494 192.1.42.3 -> 192.1.42.1 DHCP DHCP Offer Transaction - ID
0x636a38fd
20150824 08:35:27.071029 172.16.16.1 -> 172.16.16.11 DHCP DHCP Offer Transaction -
ID 0x636a38fd
20150824 08:35:27.071488 0.0.0.0 -> 255.255.255.255 DHCP DHCP Request Transaction -
ID 0x636a38fd
20150824 08:35:27.072447 192.1.42.1 -> 192.1.42.3 DHCP DHCP Request Transaction - ID
0x636a38fd
20150824 08:35:27.073008 192.1.42.3 -> 192.1.42.1 DHCP DHCP ACK Transaction -
ID 0x636a38fd
20150824 08:35:27.073692 172.16.16.1 -> 172.16.16.11 DHCP DHCP ACK Transaction - ID

  0x636a38fd
```

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

giaddr is set to 192.1.42.1 (ip address of vlan10) and suboptions 5/11/151 are set accordingly.

```
Bootp flags: 0x0000 (unicast)
client IP address: 0.0.0.0 (0.0.0.0)
Your (client) IP address: 0.0.0.0 (0.0.0.0)
Next server IP address: 0.0.0.0 (0.0.0.0)
Relay agent IP address: 192.1.42.1 (192.1.42.1)
client MAC address Hughes_01:51:51
(00:00:10:01:51:51) client hardware address padding:
00000000000000000000 Server host name not given
Boot file name not given
Magic cookie: DHCP
Option: (53) DHCP Message Type
  Length: 1
  DHCP: Discover (1)
Option: (55) Parameter Request List
  Length: 4
  Parameter Request List Item: (1) Subnet Mask
  Parameter Request List Item: (3) Router
  Parameter Request List Item: (58) Renewal Time Value
  Parameter Request List Item: (59) Rebinding Time Value
Option: (61) client identifier
  Length: 7
  Hardware type: Ethernet (0x01)
  Client MAC address: Hughes_01:51:51 (00:00:10:01:51:51)
Option: (82) Agent Information Option
  Length: 47
Option 82 Suboption: (1) Agent Circuit ID
  Length: 10
  Agent Circuit ID: 01080006001e88690030
Option 82 Suboption: (2) Agent Remote ID
  Length: 6
  Agent Remote ID: f8c2882333a5
Option 82 Suboption: (151) VRF name/VPN ID
Option 82 Suboption: (11) Server ID Override
  Length: 4
  Server ID Override: 172.16.16.1 (172.16.16.1)
Option 82 Suboption: (5) Link selection
  Length: 4
  Link selection: 172.16.16.0 (172.16.16.0)
```

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

```
              Hardware address/
              User name

         Bindings from VRF pool vxlan900001:

IP address ClientID/ Lease expiration Type State Interface
          Hardware address/
          User name
              0100.0010.0175.75 Oct 1 2018 09:21 AM Automatic Active GigabitEthernet2/1/0
              0100.0010.0151.51 Oct 1 2018 08:54 AM Automatic Active GigabitEthernet2/1/0

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

11.11.11.11/32, ubest/mbest: 2/0, attached *via
  11.11.11.11, Lo1, [0/0], 18:31:57, local *via
  11.11.11.11, Lo1, [0/0], 18:31:57, direct
22.22.22.22/32, ubest/mbest: 1/0
  *via 2.2.2.2%default, [200/0], 18:31:57, bgp65535,internal, tag 65535 (evpn)segid:
900001 tunnelid: 0x2020202
encap: VXLAN

172.16.16.0/20, ubest/mbest: 1/0, attached

 *via 172.16.16.1, Vlan1001, [0/0], 18:31:57, direct
172.16.16.1/32, ubest/mbest: 1/0, attached
 *via 172.16.16.1, Vlan1001, [0/0], 18:31:57, local
172.16.16.10/32, ubest/mbest: 1/0
 *via 2.2.2.2%default, [200/0], 00:00:47, bgp65535,internal, tag 65535 (evpn)segid:
900001 tunnelid: 0x2020202
encap: VXLAN

172.16.16.11/32, ubest/mbest: 1/0, attached

 *via 172.16.16.11, Vlan1001, [190/0], 00:28:10, hmm

switch-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

switch-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
```

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

Put DHCP server (192.1.42.3) into VRF of vxlan-900001 and make sure it is reachable from both switch-1 and switch-2 through VRF of vxlan-900001.

```
switch-1# sh run int vl 10
```

```
!Command: show running-config interface Vlan10
!Time: Mon Oct 1 09:10:26 2018
version 9.2(2)

interface Vlan10


no shutdown
vrf member vxlan-900001
ip address 192.1.42.1/24
```

Because 172.16.16.1 is an anycast address for vlan1001 configured on all the VTEPs, we need to pick up a unique address as the DHCP relay packet's source address to make sure the DHCP server can deliver a response to the original DHCP Relay agent. In this scenario, we use loopback1 and we need to make sure loopback1 is reachable from everywhere of VRF vxlan-900001.

```
switch-1# sh run int lo1

!Command: show running-config interface loopback1
!Time: Mon Oct 1 09:18:53 2018

version 9.2(2)

interface loopback1
  vrf member vxlan-900001
  ip address 11.11.11.11/32
switch-1# ping 192.1.42.3 vrf vxlan900001 source 11.11.11.11
cou 1 PING 192.1.42.3 (192.1.42.3) from 11.11.11.11: 56 data
bytes
64 bytes from 192.1.42.3: icmp_seq=0 ttl=254 time=0.575
ms - 192.1.42.3 ping statistics -
1 packets transmitted, 1 packets received, 0.00% packet
loss round-trip min/avg/max = 0.575/0.574/0.575 ms
switch-2# sh run int lo1

!Command: show running-config interface loopback1
!Time: Mon Oct 1 09:19:30 2018

version 9.2(2)

interface loopback1
  vrf member vxlan900001
  ip address 22.22.22.22/32
switch-2# ping 192.1.42.3 vrf vxlan-900001 source 22.22.22.22
cou 1 PING 192.1.42.3 (192.1.42.3) from 22.22.22.22: 56 data
bytes
64 bytes from 192.1.42.3: icmp_seq=0 ttl=253 time=0.662
ms - 192.1.42.3 ping statistics -
1 packets transmitted, 1 packets received, 0.00% packet
loss round-trip min/avg/max = 0.662/0.662/0.662 ms
```

DHCP Relay Configuration

- switch-1

```
  switch—1# sh run dhcp

  !Command: show running-config dhcp
  !Time: Mon Oct 1 08:26:00 2018

  version 9.2(2)

  feature dhcp

  service dhcp

  ip dhcp relay


ip dhcp relay information option
 I4ip dhcp relay information option vpn
 ipv6 dhcp relay

 interface Vlanl00l
   ip  dhcp relay address 192.1.42.3
   ip  dhcp relay source—interface loopback1
```

- switch-2

```
switch—2# sh run dhcp

!Command: show running-config dhcp
!Time: Mon Oct 1 08:26:16 2018

version 9.2(2) 11(3)

feature dhcp

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

interface Vlanl00l
  ip dhcp relay address 192.1.42.3
  ip dhcp relay source—interface loopback1
```

Debug Output

- The following is a packet dump for DHCP interact sequences.

```
switch-1# ethanalyzer local interface inband display-filter
"udp.srcport==67 or udp.dstport==67" limit-captured frames 0
Capturing on inband
20150824 09:31:38.129393 0.0.0.0 -> 255.255.255.255 DHCP DHCP Discover - Transaction
ID 0x860cd13
20150824 09:31:38.129952 11.11.11.11 -> 192.1.42.3 DHCP DHCP Discover - Transaction
 ID 0x860cd13
20150824 09:31:40.130134 192.1.42.3 -> 11.11.11.11 DHCP DHCP Offer - Transaction ID
0x860cd13
20150824 09:31:40.130552 172.16.16.1 -> 172.16.16.11 DHCP DHCP Offer - Transaction
ID 0x860cd13
20150824 09:31:40.130990 0.0.0.0 -> 255.255.255.255 DHCP DHCP Request - Transaction
 ID 0x860cd13
```

```
20150824 09:31:40.131457 11.11.11.11 -> 192.1.42.3 DHCP DHCP Request - Transaction
ID 0x860cd13
20150824 09:31:40.132009 192.1.42.3 -> 11.11.11.11 DHCP DHCP ACK - Transaction ID
0x860cd13
20150824 09:31:40.132268 172.16.16.1 -> 172.16.16.11 DHCP DHCP ACK - TransactionID
0x860cd13
```

• DHCP Discover packet swtich-1 sent to DHCP server.
giaddr is set to 11.11.11.11(loopback1) and suboptions 5/11/151 are set accordingly.

```
Bootstrap Protocol
  Message type: Boot Request (1)
  Hardware type: Ethernet (0x01)
  Hardware address length: 6
  Hops: 1
  Transaction ID: 0x0860cd13
  Seconds elapsed: O
  Bootp flags: 0x0000 (unicast)
  Client IP address: 0.0.0.0 (0.0.0.0)
  Your (client) IP address: 0.0.0.0 (0.0.0.0)
  Next server IP address: 0.0.0.0 (0.0.0.0)
  Relay agent iP address: 11.11.11.11 (11.11.11.11)
  Client MAC address: Hughes_01:51:51 (00:00:10:01:51:51)
  Client hardware address padding: 00000000000000000000
  Server host name not given
  Boot file name not given
  Magic cookie: DHCP
  Option: (53) DHCP Message Type
    Length: 1
    DHCP: Discover (1)
  Option: (55) Parameter Request List
  Option: (61) Client Identifier
  Option: (82) Agent Information Option
    Length: 47
  Option 82 suboption: (1) Aqent Circuit ID
  Option 82 suboption: (151) Agent Remote ID
  Option 82 suboption: (11) Server ID Override
    Length: 4
    Server ID override: 172.16.16.1 (172.16.16.1)
  Option 82 suboption: (5) Link selection
    Length: 4
    Link selection: 172.16.16.0 (172.16.16.0)
```

```
Switch-1# 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
             0100.0010.0175.75 Oct 1 2018 10:02 AM Automatic Active GigabitEthernet2/1/0
             0100.0010.0151.51 Oct 1 2018 09:50 AM Automatic Active GigabitEthernet2/1/0
switch-1# sh ip route vrf vxlan-900001
IP Route Table for VRF "vxlan-900001"
```

```
'*' denotes best ucast nexthop
'**' denotes best mcast nexthop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

11.11.11.11/32, ubest/mbest: 2/0, attached
  *via 11.11.11.11, Lo1, [0/0], 19:13:56, local


  *via 11.11.11.11, Lo1, [0/0], 19:13:56, direct
22.22.22.22/32, ubest/mbest: 1/0
  *via 2.2.2.2%default, [200/0], 19:13:56, bgp65535,internal, tag 65535 (evpn)segid:
900001 tunnelid: 0x2020202
encap: VXLAN
172.16.16.0/20, ubest/mbest: 1/0, attached
  *via 172.16.16.1, Vlan1001, [0/0], 19:13:56, direct
172.16.16.1/32, ubest/mbest: 1/0, attached
  *via 172.16.16.1, Vlan1001, [0/0], 19:13:56, local
172.16.16.10/32, ubest/mbest: 1/0
  *via 2.2.2.2%default, [200/0], 00:01:27, bgp65535,
internal, tag 65535 (evpn)segid: 900001 tunnelid: 0x2020202
encap: VXLAN
172.16.16.11/32, ubest/mbest: 1/0, attached
  *via 172.16.16.11, Vlan1001, [190/0], 00:13:56, hmm
192.1.42.0/24, ubest/mbest: 1/0, attached
  *via 192.1.42.1, Vlan10, [0/0], 00:36:08, direct
192.1.42.1/32, ubest/mbest: 1/0, attached
  *via 192.1.42.1, Vlan10, [0/0], 00:36:08, local
switch-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

switch-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
```

6.5.3 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.

```
switch-1# sh run int vl 10

!Command: show runningconfig interface Vlan10
!Time: Mon Oct 1 08:48:22 2018

version 9.2(2)
interface Vlan10
  no shutdown
  vrf member vxlan900002 ip
  address 192.1.42.1/24

switch-1# sh run int lo2
```

```
!Command: show runningconfig interface loopback2
!Time: Mon Oct 1 08:48:57 2018
version 9.2(2)
interface loopback2
  vrf member vxlan900002
  ip address 33.33.33.33/32

switch-2# sh run int lo2

!Command: show runningconfig interface loopback2
!Time: Mon Oct 1 08:48:44 2018


version 9.2(2)
interface loopback2
  vrf member vxlan900002
  ip address 44.44.44.44/32

switch-1# ping 192.1.42.3 vrf vxlan-900002 source 33.33.33.33
cou 1 PING 192.1.42.3 (192.1.42.3) from 33.33.33.33: 56 data
bytes
64 bytes from 192.1.42.3: icmp_seq=0 ttl=254 time=0.544
ms - 192.1.42.3 ping statistics -
1 packets transmitted, 1 packets received, 0.00% packet
loss round-trip min/avg/max = 0.544/0.544/0.544 ms

switch-2# ping 192.1.42.3 vrf vxlan-900002 source 44.44.44.44 count
1 PING 192.1.42.3 (192.1.42.3) from 44.44.44.44: 56 data bytes
64 bytes from 192.1.42.3: icmp_seq=0 ttl=253 time=0.678
ms - 192.1.42.3 ping statistics -
1 packets transmitted, 1 packets received, 0.00% packet
loss round-trip min/avg/max = 0.678/0.678/0.678 ms
```

DHCP Relay Configuration

- switch-1

```
switch—1# sh run dhcp

!Command: show running—config dhcp
!Time: Mon Oct 1 08:26:00 2018

version 9.2(2)

feature dhcp

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

interface VlanlOOl
  ip dhcp relay address 192.1.42.3 use—vrf vxlan—900002
  ip dhcp relay source—interface loopback2
```

- switch-2

```
!Command: show running-config dhcp
!Time: Mon Oct 1 08:26:16 2018
```

```
version 9.2(2)
feature dhcp
service dhcp
ip dhcp relay
ip dhcp relay information option
ip dhcp relay information option vpn
ipv6 dhcp relay
interface VlaniOOl
  ip dhcp relay address 192.1.42.3 use-vrf vxlan—900002
  ip dhcp relay source—interface loopback2
```

Debug Output

- The following is a packet dump for DHCP interact sequences.

```
switch-1# ethanalyzer local interface inband display-filter "udp.srcport==67
or udp.dstport==67" limit-captured-frames 0
Capturing on inband
20150825 08:59:35.758314 0.0.0.0 -> 255.255.255.255 DHCP DHCP Discover - Transaction
ID 0x3eebccae
20150825 08:59:35.758878 33.33.33.33 -> 192.1.42.3 DHCP DHCP Discover - Transaction
ID 0x3eebccae
20150825 08:59:37.759560 192.1.42.3 -> 33.33.33.33 DHCP DHCP Offer - Transaction ID
0x3eebccae
20150825 08:59:37.759905 172.16.16.1 -> 172.16.16.11 DHCP DHCP Offer - Transaction
ID 0x3eebccae
20150825 08:59:37.760313 0.0.0.0 -> 255.255.255.255 DHCP DHCP Request - Transaction
ID 0x3eebccae
20150825 08:59:37.760733 33.33.33.33 -> 192.1.42.3 DHCP DHCP Request - Transaction
ID 0x3eebccae
20150825 08:59:37.761297 192.1.42.3 -> 33.33.33.33 DHCP DHCP ACK - Transaction ID
0x3eebccae
20150825 08:59:37.761554 172.16.16.1 -> 172.16.16.11 DHCP DHCP ACK - Transaction ID
0x3eebccae
```

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

giaddr is set to 33.33.33.33 (loopback2) and suboptions 5/11/151 are set accordingly.

```
Bootstrap Protocol
  Message type: Boot Request (1)
  Hardware type: Ethernet (0x01)
  Hardware address length: 6
```

```
Hops: 1
Transaction ID: Ox3eebccae
Seconds elapsed: O
Bootp flags: 0x0000 (unicast)
Client IP address: 0.0.0.0 (0.0.0.0)
Your (client) IP address: 0.0.0.0 (0.0.0.0)
Next server IP address: 0.0.0.0 (0.0.0.0)
Relay agent IP address: 33.33.33.33 (33.33.33.33)
Client MAC address: i-iughes_01:51:51 (00:00:10:01:51:51)
Client hardware address padding: 00000000000000000000
Server host name not given
Boot file name not given
Magic cookie: DHCP
Option: (53) DHCP Message Type
  Length: 1
  DHCP: Discover (1)
Option: (55) Parameter Request List
Option: (61) client identifier
Option: (82) Agent Information option
  Length: 47
Option 82 Suboption: (1) Agent circuit W
Option 82 suboption: (2) Agent Remote 10
Option 82 suboption: (151) VRF name/VPN ID
Option 82 Suboption: (11) Server ID Override
  Length: 4
  Server ID Override: 172.16.16.1 (172.16.16.1)
Option 82 Suboption: (5) Link selection
  Length: 4
      Link selection: 172.16.16.0 (172.16.16.0)
```

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

The DHCP server is placed into the management VRF and is reachable the through M0 interface. The IP address changes to 10.122.164.147 accordingly.

```
switch-1# sh run int m0
!Command: show running-config interface mgmt0
!Time: Mon Oct 1 09:17:04 2018
version 9.2(2)
interface mgmt0
  vrf member management
  ip address 10.122.165.134/25
switch-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
switch-2# sh run int m0
!Command: show running-config interface mgmt0
!Time: Mon Oct 1 09:17:47 2018
version 9.2(2)
interface mgmt0
  vrf member management
  ip address 10.122.165.148/25

switch-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

- switch-1

```
switch—1# sh run dhcp switch—2# sh run dhcp

!Command: show running-config dhcp
!Time: Mon Oct 1 08:26:00 2018

version 9.2(2)
feature dhcp

service dhcp
ip dhcp relay
ip dhcp relay information option
ip dhcp relay information option vpn
 ipv6 dhcp relay
 interface VlanlOOl
   ip dhcp relay address 10.122.164.147 use—vrf management
```

- switch-2

```
switch-2# sh run dhcp
!Command: show running-config dhcp
!Time: Mon Oct 1 09:17:47 2018

version 9.2(2)
feature dhcp

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

interface VlanlOOl
ip dhcp relay address 10.122.164.147 use—vrf management
```

Debug Output

- The following is a packet dump for DHCP interact sequences.

```
switch-1# ethanalyzer local interface inband display-filter "udp.srcport==67
or udp.dstport==67" limit-captured-frames 0
Capturing on inband
20150825 09:30:54.214998 0.0.0.0 -> 255.255.255.255 DHCP DHCP Discover - Transaction
ID 0x28a8606d
20150825 09:30:56.216491 172.16.16.1 -> 172.16.16.11 DHCP DHCP Offer - Transaction
ID 0x28a8606d
20150825 09:30:56.216931 0.0.0.0 -> 255.255.255.255 DHCP DHCP Request - Transaction
 ID 0x28a8606d
20150825 09:30:56.218426 172.16.16.1 -> 172.16.16.11 DHCP DHCP ACK - Transaction ID
0x28a8606d

switch-1# ethanalyzer local interface mgmt display-filter "ip.src==10.122.164.147
or ip.dst==10.122.164.147" limit-captured-frames 0
Capturing on mgmt0
20150825 09:30:54.215499 10.122.165.134 -> 10.122.164.147 DHCP DHCP Discover -
 Transaction ID 0x28a8606d
20150825 09:30:56.216137 10.122.164.147 -> 10.122.165.134 DHCP DHCP Offer -
```

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

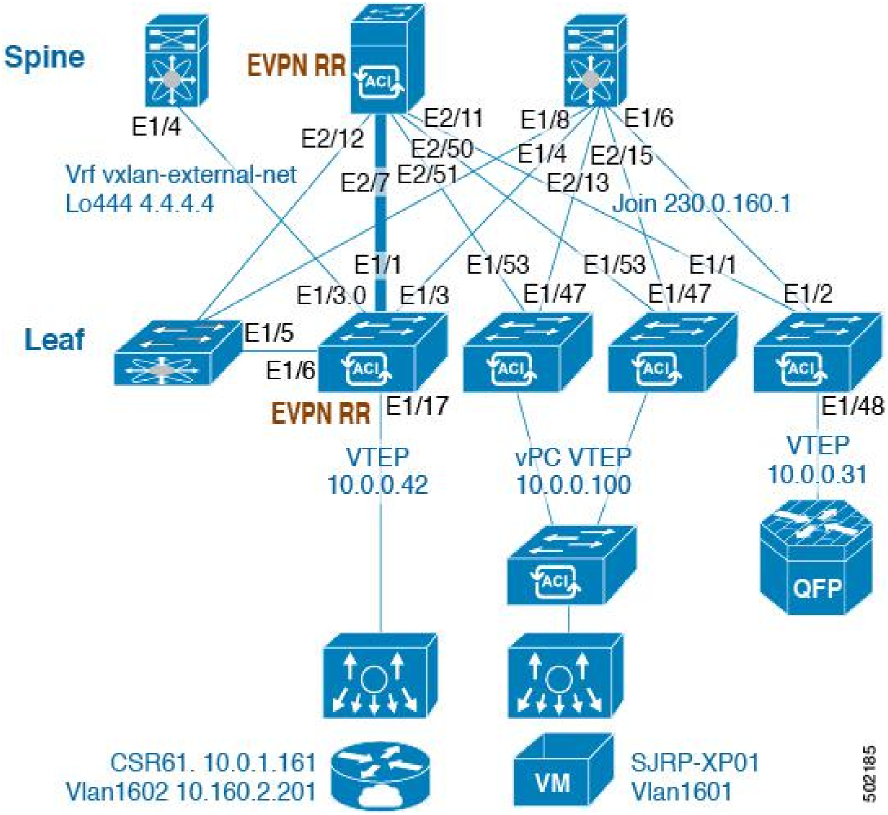
- DHCP Discover packet switch-1 sent to DHCP server.

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) Aqent 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)
```

6.6 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
ip dhcp relay information option vpn
```

- Create loopback under VRF where you need DHCP relay service.

```
interface loopback601
  vrf member evpn-tenant-kk1
  ip address 160.1.0.43/32
  ip router ospf 1 area 0     /* Only required for VPC VTEP. */
```

- Advertise LoX into the Layer 3 VRF BGP.

```
Router bgp 2
vrf X
   network 10.1.1.42/32
```

- Configure DHCP relay on the SVI under the VRF.

```
interface Vlan1601
  vrf member evpn-tenant-kk1
  ip address 10.160.1.254/24
  fabric forwarding mode anycast-gateway
  ip dhcp relay address 10.160.2.201
  ip dhcp relay source-interface loopback601
```

- Configure Layer 3 VNI SVI with **ip forward**.

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

- Create the routing VLAN/SVI forthe VPC VRF.

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

- Create the VRF routing.

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

6.7 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 192.168.1.1/30
  ip router ospf UNDERLAY area 0.0.0.0
```

```
/* vPC Peer-2 */

router ospf UNDERLAY

vrf tenant-vrf

interface Vlan2000
  no shutdown
  mtu 9216
  vrf member tenant-vrf
  ip address 192.168.1.2/30
  ip router ospf UNDERLAY area 0.0.0.0
```

- Configuration of SVI using static routes between vPC peers across the MCT:

```
/* vPC Peer-1 */

interface Vlan2000
  no shutdown
  mtu 9216
  vrf member tenant-vrf
  ip address 192.168.1.1/30

vrf context tenant-vrf

ip route 192.168.1.2/30 192.168.1.1


/* vPC Peer-2 */

interface Vlan2000
  no shutdown
  mtu 9216
  vrf member tenant-vrf
  ip address 192.168.1.2/30

vrf context tenant-vrf

ip route 192.168.1.1/30 192.168.1.2
```