# Implementing VXLAN

This module provides conceptual information for VXLAN in general and configuration information for layer 2 VXLAN on Cisco ASR 9000 Series Router. For configuration information of layer 3 VXLAN, see *Implementing L3 VXLAN* chapter in the *Cisco ASR 9000 Series Aggregation Services Router MPLS Layer 3 VPN Configuration Guide*. VXLAN provides the same Ethernet Layer 2 network services as VLAN, but with greater extensibility and flexibility.

***Table 1: Feature History for VXLAN***

| Release | Modification |
|---|---|
| Release 5.2.0 | This feature was introduced on Cisco ASR 9000 Series Router. |
| Release 5.3.1 | VXLAN Anycast Gateway feature was introduced |


- Prerequisites for implementing VXLANs, on page 1
- Information about Implementing VXLAN, on page 1
- Configuring a Layer 2 VXLAN gateway, on page 4
- Configuration Example for Implementing Layer 2 VXLAN Gateway, on page 10


# Prerequisites for implementing VXLANs

This prerequisite applies to implementing VXLANs:

You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command.

If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

# Information about Implementing VXLAN

To implement VXLAN, you must understand these concepts:

# VXLAN

VXLAN provides the same Ethernet Layer 2 network services as VLAN does today, but with greater extensibility and flexibility. It 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 core network. VXLAN is a solution to support a flexible, large-scale multitenant environment over a shared common physical infrastructure. The transport protocol over the core network is IP plus UDP. Compared to VLAN, VXLAN offers the following benefits:

- Flexible placement of multitenant segments throughout the data center: It provides a solution to extend Layer 2 segments over the underlying shared network infrastructure so that tenant workload can be placed across physical pods in the data center.
- Higher scalability to address more Layer 2 segments: VLANs use a 12-bit VLAN ID to address Layer 2 segments, which results in limiting scalability of only 4094 VLANs. VXLAN uses a 24-bit segment ID known as the VXLAN network identifier (VNID), which enables up to 16 million VXLAN segments to co-exist in the same administrative domain.
- Better utilization of available network paths in the underlying infrastructure: VLAN uses the Spanning Tree Protocol for loop prevention, which ends up not using half of the network links in a network by blocking redundant paths. In contrast, VXLAN packets are transferred through the underlying network based on its Layer 3 header and can take complete advantage of Layer 3 routing, equal-cost multipath (ECMP) routing, and link aggregation protocols to use all available paths.

# VXLAN Anycast Gateway

The VXLAN anycast gateway feature extends anycast functionality to VXLAN. It enables the use of anycast routing on a network for underlay multicast load-balancing and redundancy.

The VXLAN anycast solution:

- Allows true active-active first hop gateways (active-active on a per flow basis).
- Does not involve any new control or management plane protocols or any form of external SDN controllers or NMS to co-ordinate and synchronize gateways.

The anycast gateway feature follows these basic concepts:

- Creating a virtual Layer 3 gateway and a virtual VTEP across multiple VXLAN gateways. These gateways use an identical configuration of an overlay IP address, overlay MAC address, and underlay VTEP IP address.
- Creating a private multicast group between the gateways to use as a data plane mirror for certain types of overlay control packets.

**Note** The VXLAN anycast gateway feature is supported on only Cisco ASR 9000 High Density 100GE Ethernet line cards.

### Recommendations

These are the recommendations that users must consider before configuring VXLAN anycast gateway feature:

- BGP does not work with VXLAN anycast feature within a data center.
- IGP works on the underlay network within a data center.
- BGP and IGP should be used on the WAN side.
- Data center top-of-rack (TOR) switches to use static routes between router customer IP to the anycast gateway.

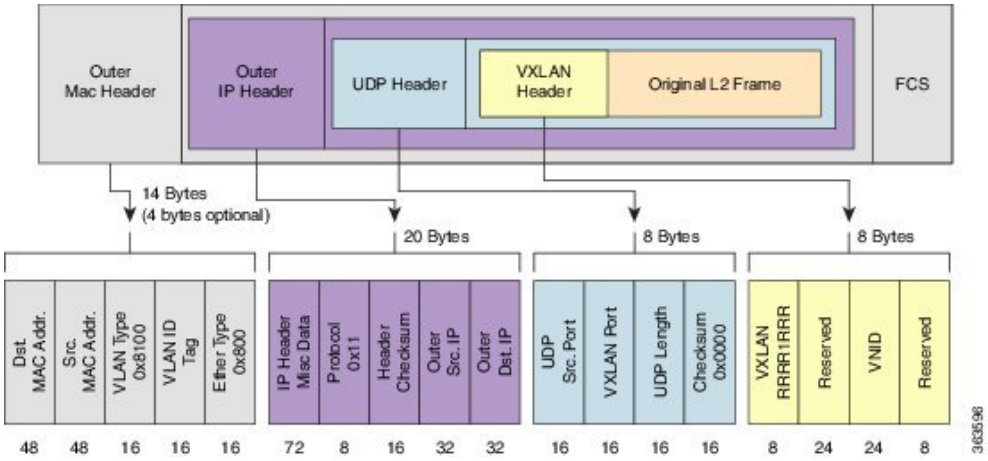
### Requirement to deploy VxLAN Anycast Gateway

Since the multicast group is used for mirroring the control frames, in case of IPv6 neighbor advertisements, the duplicate address detection (DAD) protocol will bring down the service because the same addresses are detected between two routers (or interface). Hence you must disable IPv6 DAD on the BVI interface and enable unsolicited node detection (ND) responses.

# VXLAN Packet Format

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. The VXLAN packet format is shown in the following figure.

***Figure 1: VXLAN Packet Format***



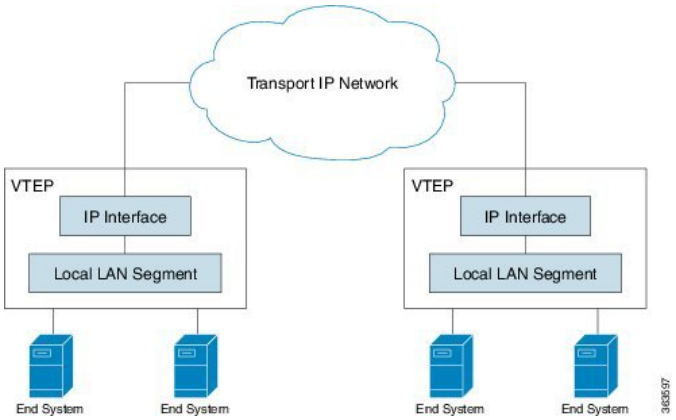

As shown in the above figure, VXLAN introduces an 8-byte VXLAN header that consists of a 24-bit VNID and a few reserved bits. The VXLAN header together with the original Ethernet frame goes in the UDP payload. The 24-bit VNID is used to identify Layer 2 segments and to maintain Layer 2 isolation between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

# VXLAN Tunnel Endpoint

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

The IP interface has a unique IP address that identifies the VTEP device on the transport IP network known as the infrastructure VLAN. The VTEP device uses this IP address to encapsulate Ethernet frames and transmits the encapsulated packets to the transport network through the IP interface. A VTEP device also discovers the remote VTEPs for its VXLAN segments and learns remote MAC Address-to-VTEP mappings through its IP interface. The functional components of VTEPs and the logical topology that is created for Layer 2 connectivity across the transport IP network is shown in the following figure.

***Figure 2: VTEP***




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.

# Configuring a Layer 2 VXLAN gateway

A Layer 2 VXLAN gateway bridges traffic between VXLAN and non-VXLAN segments (such as VLAN or VPLS) within the same layer 2 network. The operation of a VXLAN Layer 2 gateway is based on the data plane MAC address learning and flooding of multi-destination traffic such as unknown unicast, multicast, or broadcast frames, using IP multicast. The following sections show how to configure an ASR 9000 series router as a Layer 2 VXLAN gateway between a VLAN and a VXLAN segment in the same L2 domain.

## Prerequisites

The following are the prerequisites to configuring a Cisco ASR 9000 series router as a VXLAN Layer 2 gateway:

- Configure a loopback interface. It serves as a source interface for the local VTEP.
- Configure unicast reachability to remote VTEPs.
- Configure Bidirectional Protocol Independent Multicast (Bidir PIM) or PIM Sparse Mode. For more information, see the *Multicast Configuration Guide for Cisco ASR 9000 Series Routers*.

## Restrictions

Consider the following restrictions while configuring VXLAN:

- You configure VXLAN only on Overlay Transport Virtualization (OTV) and VXLAN UDP ports.
- The source interface can only be a loopback interface.
- You cannot share a VNI or a multicast group or a source interface across multiple NVE interfaces.
- The VNI range and the multicast range both can only be specified contiguously. A non-contiguous range with comma separated values is not supported.
- The VNI to multicast group mapping can be only either 1:1 or N:1. For example,
  - The "member vni 5000 mcast-group 239.1.1.1" command configures a valid 1:1 mapping.
  - The "member vni 5000-5005 mcast-group 239.1.1.1" command configures a valid N:1 mapping.
- When a VNI is configured as a part of a VNI range, it can be modified or deleted only as part of the same range. For example, if the "member vni 5000-5002 mcast-group 239.1.1.1" command is configured, you cannot disassociate just the VNI 5001 from the NVE interface with a "no member vni 5001" command.
- Static MAC configuration is not supported.
- You can configure a maximum of 128k Layer 2 and Layer 3 sub-interfaces per system. The configuration can be a combination of both Layer 2 sub-interfaces and Layer 3 sub-interfaces; or either fully Layer 2 sub-interfaces or Layer 3 sub-interfaces.

  Though the system allows you to configure more than 128k sub-interfaces per system, you cannot use this configuration for services. Though the system displays a warning message on reaching the threshold of 128k sub-interfaces, the configuration is still applied. However, you cannot use this configuration for services.

# Creating and Configuring the Network Virtualization Endpoint (NVE) interface

Perform this task to create an NVE interface and configure it as a VXLAN Tunnel EndPoint (VTEP) for VXLAN.

### SUMMARY STEPS

1. **interface nve** *nve-identifier*
2. *(Optional)* **overlay-encapsulation vxlan**
3. **source-interface loopback** *loopback-interface-identifier*
4. **member vni** *vni_number* [ *-end_vni_range* ] **mcast-group** *ip_address* [ *end_ip_address_range* ]
5. *(Optional)* **anycast source-interface loopback** *loopback-interface-identifier* **sync-group** *ip_address*
6. Use the **commit** or **end** command.

### DETAILED STEPS

**Procedure**

**Step 1** **interface nve** *nve-identifier*

**Example:**

```
RP/0/RSP0/CPU0:router(config)# interface nve 1
```

Creates the NVE interface and enters the NVE interface configuration sub-mode.

**Step 2** *(Optional)* **overlay-encapsulation vxlan**

**Example:**

```
RP/0/RSP0/CPU0:router(config-if)# overlay-encapsulation vxlan
```

Sets VXLAN encapsulation for the NVE interface. VXLAN is the default encapsulation for an NVE interface. This step is optional if you have not changed the encapsulation.

**Step 3** **source-interface loopback** *loopback-interface-identifier*

**Example:**

```
RP/0/RSP0/CPU0:router(config-if)# source-interface loopback 1
```

Sets a loopback interface as the source interface for the VTEP.

**Step 4** **member vni** *vni_number* [ *-end_vni_range* ] **mcast-group** *ip_address* [ *end_ip_address_range* ]

**Example:**

```
RP/0/RSP0/CPU0:router(config-if)# member vni 1-10 mcast-group 224.2.2.2 224.2.2.10
```

Associates a single VXLAN or a contiguous range of VXLANs with the NVE interface using their VXLAN Network Identifiers (VNIs) and specifies a multicast address or a contiguous multicast address range associated with these VNIs.

**Note**

- The mapping between the VNIs and the multicast groups is either one-to-one or many-to-one.
- To associate discontiguous VXLANs or VXLAN ranges with the NVE interface, perform this step for each VXLAN or VXLAN range. For instance,

```
RP/0/RSP0/CPU0:router(config-if)# member vni 10 mcast-group 224.2.2.10
RP/0/RSP0/CPU0:router(config-if)# member vni 23 mcast-group 224.2.2.23
RP/0/RSP0/CPU0:router(config-if)# member vni 50-59 mcast-group 224.2.2.50 224.2.2.59
RP/0/RSP0/CPU0:router(config-if)# member vni 100-120 mcast-group 224.2.2.100 224.2.2.120
```

**Step 5** *(Optional)* **anycast source-interface loopback** *loopback-interface-identifier* **sync-group** *ip_address*

**Example:**

```
RP/0/RSP0/CPU0:router(config-if)# anycast source-interface loopback 1 sync-group 192.23.2.20
```

Configures anycast mode parameters for this VTEP.

**Step 6** Use the **commit** or **end** command.

**commit** - Saves the configuration changes and remains within the configuration session.

**end** - Prompts user to take one of these actions:

- **Yes** - Saves configuration changes and exits the configuration session.
- **No** - Exits the configuration session without committing the configuration changes.
- **Cancel** - Remains in the configuration mode, without committing the configuration changes.

### What to do next

Use the **show nve interface** command to display the configured NVE interface information.

# Creating and configuring a layer 2 sub-interface

Perform this task to create a layer 2 sub-interface associated with a VLAN segment.

### SUMMARY STEPS

1. **interface gigabitEthernet** *interface-identifier* **l2transport**
2. **dot1q vlan** *vlan-identifier*
3. Use the **commit** or **end** command.

### DETAILED STEPS

**Procedure**

**Step 1** **interface gigabitEthernet** *interface-identifier* **l2transport**

**Example:**

```
RP/0/RSP0/CPU0:router(config)# interface gigabitEthernet 0/0/0/0.100 l2transport
```

Creates a layer 2 sub-interface and enters the sub-interface configuration mode.

**Step 2** **dot1q vlan** *vlan-identifier*

**Example:**

```
RP/0/RSP0/CPU0:router(config-if)# dot1q vlan 100
```

Sets the VLAN for the interface.

**Step 3** Use the **commit** or **end** command.

**commit** - Saves the configuration changes and remains within the configuration session.

**end** - Prompts user to take one of these actions:

- **Yes** - Saves configuration changes and exits the configuration session.
- **No** - Exits the configuration session without committing the configuration changes.
- **Cancel** - Remains in the configuration mode, without committing the configuration changes.

# Associating VLAN and VXLAN with a bridge domain

Perform this task to associate a VLAN and a VXLAN with a bridge domain.

### SUMMARY STEPS

1. **l2vpn**
2. **bridge group** *bridge-group-name*
3. **bridge-domain** *bridge-domain-name*
4. **member vni** *vxlan-identifier*
5. **interface gigabitEthernet** *sub-interface-identifier*
6. Use the **commit** or **end** command.

**DETAILED STEPS**

**Procedure**

**Step 1** **l2vpn**

**Example:**

```
RP/0/RSP0/CPU0:router(config)# l2vpn
```

Enters the l2vpn configuration mode.

**Step 2** **bridge group** *bridge-group-name*

**Example:**

```
RP/0/RSP0/CPU0:router(config-l2vpn)# bridge group bridgegroup1
```

Enters the bridge group configuration mode.

**Step 3** **bridge-domain** *bridge-domain-name*

**Example:**

```
RP/0/RSP0/CPU0:router(config-l2vpn-bg)# bridge-domain bdomain1
```

Enters the bridge domain configuration mode.

**Step 4** **member vni** *vxlan-identifier*

**Example:**

```
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# member vni 100
```

Associates a VXLAN with the bridge domain.

**Step 5** **interface gigabitEthernet** *sub-interface-identifier*

**Example:**

```
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# interface gigabitEthernet 0/0/0/0.200
```

Associates a VLAN with the bridge domain using the VLAN sub-interface.

**Step 6** Use the **commit** or **end** command.

**commit** - Saves the configuration changes and remains within the configuration session.

**end** - Prompts user to take one of these actions:

- **Yes** - Saves configuration changes and exits the configuration session.
- **No** - Exits the configuration session without committing the configuration changes.
- **Cancel** - Remains in the configuration mode, without committing the configuration changes.

# Configuring VXLAN source UDP port

This is an optional task. By default, the source UDP port of the encapsulating VXLAN segment is calculated via hash functions on the layer 2 address fields of the inner payload. Perform this task to configure the hash functions to be performed on either the layer 2 or the layer 3 address fields of the inner payload.

### SUMMARY STEPS

1. **l2vpn**
2. **load-balancing flow** [ *src-dst-mac* | *src-dst-ip* ]

### DETAILED STEPS

#### Procedure

**Step 1** **l2vpn**

**Example:**

```
RP/0/RSP0/CPU0:router(config)# l2vpn
```

Enters the l2vpn configuration mode.

**Step 2** **load-balancing flow** [ *src-dst-mac* | *src-dst-ip* ]

**Example:**

```
RP/0/RSP0/CPU0:router(config-l2vpn)# load-balancing flow src-dst-mac
```

Selects either the layer 2 or layer 3 address fields of the inner payload for hash function.

# Configuring VXLAN destination UDP port

The UDP port numbers 4789 and 8472 are assigned to VXLAN and OTV respectively. Perform this task to configure the destination UDP port number of the encapsulating VXLAN segment. This is an optional task because, by default, the destination UDP port number of the encapsulating VXLAN datagram is set to 4789. The destination UDP port number should be set to 8472 if the destination VTEP provides VXLAN support using an OTV port.

### SUMMARY STEPS

1. **vxlan udp port** *port-number*

### DETAILED STEPS

#### Procedure

**vxlan udp port** *port-number*

**Example:**

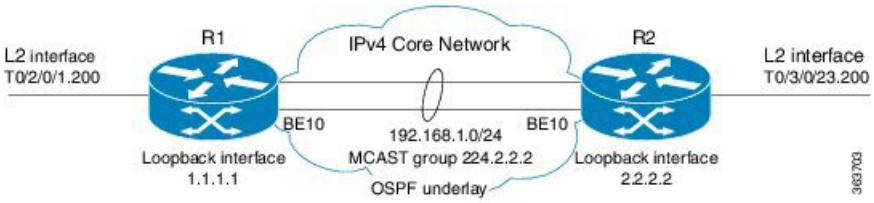
```
RP/0/RSP0/CPU0:router(config)# vxlan udp port 4789
```

Sets the destination UDP port number of the encapsulating VXLAN segment.

# Configuration Example for Implementing Layer 2 VXLAN Gateway

The following example shows layer 2 VXLAN gateway configuration on two Provider Edge (PE) routers, R1 and R2, from a sample network topology that has the core network simplified as a bundle link connection between the PE routers.

***Figure 3: Network with Layer 2 VXLAN Gateways***




### Configuration at R1:

```
interface Bundle-Ether10
  ipv4 address 192.168.1.1/24
!
interface Loopback0
  ipv4 address 1.1.1.1/32
!
interface T0/2/0/1
 no shut
!
interface T0/2/0/1.200 l2transport
 encapsulation dot1q 200
!
router ospf underlay
 router-id 1.1.1.1
 area 0
  interface Bundle-Ether10
  interface Loopback0
!
interface nve 1
  member vni 1 mcast-group 224.2.2.2 0.0.0.0
  overlay-encapsulation vxlan
  source-interface Loopback0
!
l2vpn
 bridge group vxlan
  bridge-domain vxlan
    interface T0/2/0/1.200
    member vni 1
  !
multicast-routing
 address-family ipv4
  interface Loopback0
    enable
  interface Bundle-Ether10
     enable
!
router pim
 address-family ipv4
  rp-address 1.1.1.1 bidir
```

### Configuration at R2:

```
interface Bundle-Ether10
  ipv4 address 192.168.1.2/24
!
interface Loopback0
  ipv4 address 2.2.2.2/32
!
interface T0/3/0/23
 no shut
!
interface T0/3/0/23.200 l2transport
 encapsulation dot1q 200
!
router ospf underlay
 router-id 2.2.2.2
 area 0
  interface Bundle-Ether10
  interface Loopback0
!
Interface nve 1
  member vni 1 mcast-group 224.2.2.2 0.0.0.0
  overlay-encapsulation vxlan
  source-interface Loopback0
!
l2vpn
 bridge group vxlan
  bridge-domain vxlan
    interface T0/3/0/23.200
    member vni 1
  !
multicast-routing
 address-family ipv4
  interface Loopback0
    enable
  interface Bundle-Ether10
     enable
!
router pim
 address-family ipv4
  rp-address 1.1.1.1 bidir
```