Configuring VXLAN BGP EVPN using Ingress Replication

This section talks about how to configure VXLAN BGP EVPN using ingress replication:

Prerequisites for VXLAN EVPN Ingress Replication

- Virtual Extensible LAN (VXLAN) needs to be enabled.

- Ingress Replication (IR) on Ethernet VPN (EVPN) over VXLAN requires the Inclusive Multicast Ethernet Tag (IMET) route, also known as Route Type 3 (RT3), to be supported for VXLAN encapsulation.

- The switch on which VXLAN EVPN ingress replication is configured should support 300 remote Virtual Tunnel End Points (VTEPs), and a total of 200 Layer 2 Virtual Network Identifiers (L2VNIs) or VLANs.

Information About VXLAN EVPN Ingress Replication

Ingress replication feature has been introduced on BGP EVPN over VXLAN to forward Broadcast, Unknown Unicast, and Multicast (BUM) traffic to the relevant recipients in a network. Ingress replication for VXLAN EVPN is deployed when IP multicast underlay network is not used. Ingress replication, or headend replication, is a unicast approach to handle multi destination trafffic. Handling BUM traffic in a network using ingress replication involves an ingress device replicating every BUM packet and sending them as a separate unicast to the remote egress devices.

Ingress Replication Operation

BGP EVPN uses ingress replication through IMET routing, also called Route Type 3 (RT3), for the auto discovery of remote peers in order to set up the BUM tunnels over VXLAN. IMET routes carry the remote (egress) Virtual Network Identifiers (VNIs) advertised from the remote peers, which can be different from the local VNI. These remote VNIs are called downstream assigned VNIs.

A VXLAN tunnel adjacency is created after receiving the IMET IR routes from remote Network Virtualization Edge (NVE) peers. The tunnel adjacency is a mid-chain adjacency that contains IP or UDP encapsulation for the VXLAN tunnel. If there is more than one VNI along the tunnel, the tunnel is shared by multiple VNIs. IMET ingress replication on EVPN can have multiple unicast tunnel adjacencies and different egress VNIs for each remote peer.

Note   When no NVE peer is using a VXLAN tunnel adjacency, the tunnel adjacency is deleted.

A flooded replication list, also known as a flood list in data plane, is built using the routes advertised by each VTEP. The dynamic replication list stores all the remote destination peers in the same Layer2 VNIs that are discovered on a BGP IMET route. When a VNI is configured on a remote peer, the replication list gets updated. A VXLAN encapsulation is created and linked to the corresponding tunnel adjacency. The tunnel adjacency and VXLAN encapsulation are added to the replication list when it receives an IMET IR route from a remote NVE peer. After the remote NVE peer withdraws the IMET IR route, the tunnel adjacency and VXLAN encapsulation are removed from the replication list. After the replication list is built, any BUM traffic that reaches the ingress device gets replicated and forwarded throughout the network to all the remote peers in a VNI.

Note   High availibility (HA) and stateful switchover (SSO) are supported for VXLAN EVPN ingress replication.

Configuring VXLAN BGP EVPN using Ingress Replication

This section talks about how to configure VXLAN BGP EVPN using ingress replication:

Configuring EVPN Replication Type on Leaf Switch

To configure the replication type on the leaf switch, perform the following steps:

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** `Device> enable`	Enters privileged configuration mode. Enter password, if prompted.
Step 2	configure terminal **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	l2vpn evpn **Example:** `Device(config)# l2vpn evpn`	Enters EVPN configuration mode.

	Command or Action	Purpose
Step 4	**replication-type ingress** **Example:** Device(config-evpn)# **replication-type ingress**	Configures the L2VPN EVPN replication type as ingress replication.
Step 5	**end** **Example:** Device(config-evpn)# **end**	Returns to privileged EXEC mode.

Configuring Encapsulation Type for EVPN Instance

To configure the encapsulation type for an EVPN instance, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** Device> **enable**	Enters privileged configuration mode. • Enter your password, if prompted.
Step 2	**configure terminal** **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	**l2vpn evpn instance** *evpn-identifier-value* **vlan-based** **Example:** Device(config)# **l2vpn evpn instance 1 vlan-based**	Enters the EVPN-EVI configuration mode for the specified VLAN-based EVPN instance.
Step 4	**encapsulation vxlan** **Example:** code Device(config-evpn-evi)# **encapsulation vxlan**	Configures the encapsulation type for the VLAN-based EVPN instance as VXLAN.
Step 5	end **Example:** Device(config-evpn-evi)# **end**	Returns to privileged EXEC mode.

Adding a VNI Node to NVE

To add a Virtual Network Identifier (VNI) node to the Network Virtualization Edge (NVE), perform the following task:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** Device> **enable**	Enters privileged configuration mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** Device# **configure terminal**	Enters global configuration mode.
Step 3	**interface** *nve-interface-name* **Example:** Device(config)# **interface nve1**	Defines the interface to be configured as a trunk, and enters interface configuration mode.
Step 4	**host-reachability protocol bgp** **Example:** Device(config-if)# **host-reachability protocol bgp**	Configures BGP as the host-rechability protocol on the interface.
Step 5	**member vni** *vni-id* **ingress-replication** **Example:** Device(config-if)# **member vni 20015 ingress-replication**	Adds the VNI member with Ingress Replication mode to the NVE.
Step 6	end **Example:** Device(config-if-nve-vni)# **end**	Returns to privileged EXEC mode.

Configuration Examples for VXLAN EVPN Ingress Replication

This section provides the various examples that show the configuration and monitoring of IMET ingress replication for VXLAN EVPN:

Example: Configuring IMET Ingress Replication over EVPN VXLAN

The following example shows how to configure IMET ingress replication for VXLAN:

```
Device# configure terminal
Device(config)# l2vpn evpn
Device(config-evpn)# replication-type ingress
Device(config-evpn)# exit
Device(config)# l2vpn evpn instance 1 vlan-based
Device(config-evpn-evi)# encapsulation vxlan
Device(config-evpn-evi)# exit
Device(config)# int nve1
Device(config-if)# host-reachability protocol bgp
Device(config-if)# member vni 20015 ingress-replication
```

Producing and Managing an IMET Ingress Replication Route

This section contains the various examples that show the production and management of an IMET ingress replication route:

Example: EVPN Manager-Produced Local IMET Route

The following example shows a local IMET route produced by the EVPN manager:

```
Device# show l2vpn evpn evi 1 detail

EVPN instance: 1 (VLAN Based)
RD: 10.3.3.3:1 (auto)
Import-RTs: 3:1
Export-RTs: 3:1
Per-EVI Label: none
State: Established
Replication Type: Ingress (global)
Encapsulation: vxlan
IP Local Learn: Enable (global)
Vlan: 11
Ethernet-Tag: 0
State: Established
Core If: Vlan100
Access If: Vlan11
NVE If: nve1
RMAC: 2c5a.0f1c.da21
Core Vlan: 100
L2 VNI: 10000
L3 VNI: 100000
VTEP IP: 10.33.33.33
VRF: Red
IPv4 IRB: Enabled
IPv6 IRB: Enabled
Pseudoports:
GigabitEthernet1/0/7 service instance 11
GigabitEthernet1/0/8 service instance 11
```

Example: L2RIB Local IMET Route

The following example shows a L2RIBB Local IMET route:

```
Device# l2route evpn imet detail

EVPN Instance: 1
Ethernet Tag: 0
Producer Name: BGP
```

```
          Router IP Addr: 10.44.44.44
          Route Ethernet Tag: 0
          Tunnel Flags: 0
          Tunnel Type: Ingress Replication
          Tunnel Labels: 10001
          Tunnel ID: 10.44.44.44
          Next Hop(s): V:0 10.44.44.44
```

Example: BGP Local IMET Route with VXLAN Encapsulation and VNI

The following example shows a BGP local IMET route with VXLAN encapsulation and VNI:

```
Device# show ip bgp l2vpn evpn evi 1 route-type 3

BGP routing table entry for [3][10.10.10.10:1][0][32][10.10.10.1]/17, version 6
Paths: (1 available, best #1, table evi_1)
  Advertised to update-groups:
     1
  Refresh Epoch 1
  Local
  Updated on Jul 30 2018 18:16:34 PST
    :: (via default) from 0.0.0.0 (10.10.10.10)
      Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
      Extended Community: RT:1:1 ENCAP:8
      PMSI Attribute: Flags:0x0, Tunnel type:6, length 4, vni:10000 tunnel parameters: 0000
 0000
      Local irb vxlan vtep:
        vrf:not found, l3-vni:0
        local router mac:0000.0000.0000
        core-irb interface:(not found)
        vtep-ip:10.10.10.1
      rx pathid: 0, tx pathid: 0x0
```

Receiving and Programming an IMET IR Route

This section contains various examples that show an IMET ingress replication route being received and programmed:

Example: BGP Importing IMET Route with VxLAN Encapsulation and Showing the VNI

The following example shows the importing of IMET Route with VxLAN Encapsulation by BGP, along with the VNI:

```
Device# show ip bgp l2vpn evpn evi 1 route-type 3

BGP routing table entry for [3][10.10.10.10:1][1][32][10.1.1.1]/17, version 4
Paths: (1 available, best #1, table evi_1)
  Not advertised to any peer
  Refresh Epoch 1
  Local, imported path from [3][10.2.2.2:1000][1][32][10.1.1.1]/17 (global)
  Updated on Jul 30 2018 18:11:37 PST
    10.19.101.1 (via default) from 10.20.0.45 (19.0.0.1)
      Origin IGP, localpref 100, valid, internal, best
      Extended Community: RT:1:1 ENCAP:8
      Originator: 10.19.101.1, Cluster list: 10.19.0.1
      PMSI Attribute: Flags:0x0, Tunnel type:6, length 4, vni:16777215 tunnel parameters:
1300 6501
```

Example: L2FIB Remote IMET Route

The following examples provide details about an L2FIB remote IMET route:

```
Device# show l2fib bridge-domain 11 details

Bridge Domain : 11
Reference Count : 13
Replication ports count : 4
Unicast Address table size : 2
IP Multicast Prefix table size : 3

Flood List Information :
Olist: 1035, Ports: 4

VxLAN Information :

Port Information :
BD_PORT Gi1/0/7:11
BD_PORT Gi1/0/8:11
VXLAN_REP PL:1(1) T:VXLAN_REP [IR]10001:10.44.44.44
VXLAN_REP PL:57(1) T:VXLAN_REP [IR]10000:10.55.55.55

Unicast Address table information :
0013.0100.0001 VXLAN_CP L:10000:10.33.33.33 R:10000:10.55.55.55
d4e8.80b0.99bf VXLAN_CP L:10000:10.33.33.33 R:10000:10.55.55.55

IP Multicast Prefix table information :
Source: *, Group: 224.0.0.0/24, IIF: Null, Adjacency: Olist: 1035, Ports: 4
Source: *, Group: 224.0.1.39, IIF: Null, Adjacency: Olist: 1035, Ports: 4
Source: *, Group: 224.0.1.40, IIF: Null, Adjacency: Olist: 1035, Ports: 4


Device# show l2fib output-list 1035

ID : 1035
Bridge Domain : 11
Reference Count : 4
Flags : flood list
Port Count : 4
Port(s) : BD_PORT Gi1/0/8:11
: BD_PORT Gi1/0/7:11
: VXLAN_REP PL:1(1) T:VXLAN_REP [IR]10001:10.44.44.44
: VXLAN_REP PL:57(1) T:VXLAN_REP [IR]10000:10.55.55.55
```

Example: L2RIB Remote IMET Route

The following example provides details about an L2RIB remote IMET route:

```
Device# show l2route evpn imet det

EVPN Instance:          11
Ethernet Tag:           0
Producer Name:          BGP
Router IP Addr:         10.2.2.2
Route Ethernet Tag:     0
Tunnel Flags:           0
Tunnel Type:            Ingress Replication
Tunnel Labels:          20011
```

```
Tunnel ID:             10.2.2.2
Tunnel Encapsulation:  vxlan
```

Example: Tunnel Adjacency for a Remote IMET Route

The following example shows the tunnel adjacency for a remote IMET route:

```
Device# show adjacency tu0 10.2.2.2 internal

Protocol Interface                Address
IP      Tunnel0                   10.2.2.2(4)
                                  237 packets, 92420 bytes
                                  epoch 0
                                  sourced in sev-epoch 2
                                  Encap length 28
                                  4500000000000000FF11AFE104040404
                                  0202020212B512B500000000
                                  Tun endpt
                                  Next chain element:
                                    IP adj out of Ethernet1/0, addr 10.3.1.2 7F21C7BE60A0
                                    parent oce 0x7F21C7BE6160
                                    frame originated locally (Null0)
                                  L3 mtu 4000
                                  Flags (0x4808C4)
                                  Fixup enabled (0x2)
                                      IP tunnel
                                  HWIDB/IDB pointers 0x7F21CDC4C218/0x7F21CDC4D5C8
                                  IP redirect disabled
                                  Switching vector: IPv4 midchain adj oce
Protocol Interface                Address
                                  IP Tunnel stack to 10.2.2.2 in Default (0x0)
                                   nh tracking enabled: 10.2.2.2/32
                                   IP adj out of Ethernet1/0, addr 10.3.1.2
                                  Adjacency pointer 0x7F21CE859D90
                                  Next-hop 10.2.2.2
```

Example: BGP Local IMET Route with VXLAN Encapsulation, and Remote IMET Route with VXLAN and MPLS

The following example shows a BGP local IMET route with VXLAN encapsulation, and a remote IMET route with VXLAN and MPLS:

```
Device# show ip bgp l2vpn evpn evi 1 route-type 3

BGP routing table entry for [3][10.10.10.10:1][0][32][10.10.10.1]/17, version 6
Paths: (1 available, best #1, table evi_1)
  Advertised to update-groups:
     1
  Refresh Epoch 1
  Local
  Updated on Jul 30 2018 18:16:34 PST
    :: (via default) from 0.0.0.0 (10.10.10.10)
      Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
      Extended Community: RT:1:1 ENCAP:8
      PMSI Attribute: Flags:0x0, Tunnel type:6, length 4, vni:10000 tunnel parameters: 0000
 0000
      Local irb vxlan vtep:
        vrf:not found, l3-vni:0
        local router mac:0000.0000.0000
        core-irb interface:(not found)
```

```
          vtep-ip:10.10.10.1
       rx pathid: 0, tx pathid: 0x0
BGP routing table entry for [3][10.10.10.10:1][1][32][1.1.1.1]/17, version 4
Paths: (1 available, best #1, table evi_1)
  Not advertised to any peer
  Refresh Epoch 1
  Local, imported path from [3][2.2.2.2:1000][1][32][1.1.1.1]/17 (global)
  Updated on Jul 30 2018 18:11:37 PST
    19.0.101.1 (via default) from 20.0.0.45 (19.0.0.1)
      Origin IGP, localpref 100, valid, internal, best
      Extended Community: RT:1:1 ENCAP:8
      Originator: 19.0.101.1, Cluster list: 19.0.0.1
      PMSI Attribute: Flags: 0x0, Tunnel type: 6, length 4, vni:16777215 tunnel parameters:
 1300 6501
       rx pathid: 0, tx pathid: 0x0
BGP routing table entry for [3][10.10.10.10:1][1][32][2.2.2.2]/17, version 5
Paths: (1 available, best #1, table evi_1)
  Not advertised to any peer
  Refresh Epoch 1
  Local, imported path from [3][2.2.2.2:1000][1][32][2.2.2.2]/17 (global)
  Updated on Jul 30 2018 18:11:37 PST
    19.0.101.2 (via default) from 20.0.0.45 (19.0.0.1)
      Origin IGP, localpref 100, valid, internal, best
      Extended Community: RT:1:1
      Originator: 19.0.101.2, Cluster list: 19.0.0.1
      PMSI Attribute: Flags: 0x0, Tunnel type: 6, length 4, label:1048575 tunnel parameters:
 1300 6502
       rx pathid: 0, tx pathid: 0x0
```

Feature Information for VXLAN EVPN Ingress Replication

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Feature Name	Release	Modification
VXLAN EVPN Ingress Replication	Cisco IOS XE Gibraltar 16.11.1	VXLAN EVPN Ingress Replication feature enables forwarding of broadcast, unknown unicast, and multicast traffic to the relevant recipients in a network. Ingress replication is a unicast approach to handling multi-destination traffic, and involves an ingress device replicating every BUM packet and then sending it as a separate unicast to remote egress devices..