Configuring Seamless Integration of EVPN with L3VPN (MPLS LDP)

This chapter contains the following sections:

Information About Configuring Seamless Integration of EVPN with L3VPN (MPLS LDP)

Data center deployments have adopted VXLAN EVPN for its benefits like EVPN control-plane learning, multitenancy, seamless mobility, redundancy, and easier POD additions. Similarly, the Core is either an LDP-based MPLS L3VPN network or transitioning from traditional an MPLS L3VPN LDP-based underlay to a more sophisticated solution like segment routing (SR). Segment routing is adopted for its benefits like unified IGP and MPLS control planes, simpler traffic engineering methods, easier configuration, and SDN adoption.

With two different technologies, one within the data center and one in the Core, it is natural to handoff from VXLAN to an MPLS-based core at the DCI nodes. These nodes which sit on the edge of the DC domain, interfacing with the Core edge router.

Guidelines and Limitations for Configuring Seamless Integration of EVPN with L3VPN (MPLS LDP)

The following are the guidelines and limitations for Configuring Seamless Integration of EVPN with L3VPN (MPLS LDP):

The following features are supported:

- Layer 3 orphans
- MPLS extended ECMP (enabled by default)

The following features are not supported:

- Subnet stretches across the DC domain
- vPC
- SVI/Subinterfaces

Configuring Seamless Integration of EVPN with L3VPN (MPLS LDP)

These configuration steps are required on a DCI switch to import and re-originate the routes from a VXLAN domain to an MPLS domain and back to a VXLAN domain.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**feature mpls l3vpn** **Example:** `switch# feature mpls l3vpn`	Enables the MPLS Layer 3 VPN feature.
Step 3	**feature mpls ldp** **Example:** `switch# feature mpls ldp`	Enables the MPLS Label Distribution Protocol (LDP).
Step 4	**nv overlay evpn** **Example:** `switch(config)# nv overlay evpn`	Enables the EVPN control plane for VXLAN.
Step 5	**router bgp** *number* **Example:** `switch(config)# router bgp 100`	Configures BGP. The value of the *number* argument is from 1 to 4294967295.
Step 6	**address-family ipv4 unicast** **Example:** `switch(config-router)# address-family ipv4 unicast`	Configures the address family for IPv4.
Step 7	**redistribute direct route-map** *route-map-name* **Example:**	Configures the directly connected route map.

	Command or Action	Purpose
	`switch(config-router-af)# redistribute direct route-map passall`	
Step 8	**exit** **Example:** `switch(config-router-af)# exit`	Exits command mode.
Step 9	**address-family l2vpn evpn** **Example:** `switch(config-router)# address-family l2vpn evpn`	Configures the L2VPN address family.
Step 10	**exit** **Example:** `switch(config-router-af)# exit`	Exits command mode.
Step 11	**neighbor** *address* **remote-as** *number* **Example:** `switch(config-router)# neighbor 108.108.108.108 remote-as 22`	Configures a BGP neighbor. The range of the *number* argument is from 1 to 65535.
Step 12	**update-source** *type/id* **Example:** `switch(config-router-neighbor)# update-source loopback100`	Specifies the source of the BGP session and updates.
Step 13	**ebgp-multihop** *ttl-value* **Example:** `switch(config-router-neighbor)# ebgp-multihop 10`	Specifies the multihop TTL for the remote peer. The range of *ttl-value* is from 2 to 255.
Step 14	**address-family ipv4 unicast** **Example:** `switch(config-router-neighbor)# address-family ipv4 unicast`	Configures the unicast sub-address family.
Step 15	**send-community extended** **Example:** `switch(config-router-neighbor-af)# send-community extended`	Configures the community attribute for this neighbor.
Step 16	**exit** **Example:** `switch(config-router-neighbor-af)# exit`	Exits command mode.
Step 17	**address-family vpnv4 unicast** **Example:**	Configures the address family for IPv4.

	Command or Action	Purpose
	`switch(config-router-neighbor)# address-family vpnv4 unicast`	
Step 18	**send-community extended** **Example:** `switch(config-router)# send-community extended`	Sends the extended community attribute.
Step 19	**import l2vpn evpn reoriginate** **Example:** `switch(config-router)# import l2vpn evpn reoriginate`	Reoriginates the route with a new RT.
Step 20	**neighbor** *address* **remote-as** *number* **Example:** `switch(config-router)# neighbor 175.175.175.2 remote-as 1`	Defines the neighbor.
Step 21	**address-family ipv4 unicast** **Example:** `switch(config-router)# address-family ipv4 unicast`	Configures the address family for IPv4.
Step 22	**send-community extended** **Example:** `switch(config-router)# send-community extended`	Configures the community for BGP neighbors.
Step 23	**address-family ipv6 unicast** **Example:** `switch(config-router)# address-family ipv6 unicast`	Configures the IPv6 unicast address family, which is required for IPv6 over VXLAN with an IPv4 underlay.
Step 24	**send-community extended** **Example:** `switch(config-router)# send-community extended`	Configures the community for BGP neighbors.
Step 25	**address-family l2vpn evpn** **Example:** `switch(config-router)# address-family l2vpn evpn`	Configures the L2VPN address family.
Step 26	**send-community extended** **Example:** `switch(config-router)# send-community extended`	Configures the community for BGP neighbors.

	Command or Action	Purpose
Step 27	**import vpn unicast reoriginate** **Example:** `switch(config-router)# import vpn unicast reoriginate`	Reoriginates the route with a new RT.

Configuring Seamless Integration of EVPN with L3VPN (MPLS LDP)