InterAS Option B

This chapter explains the different InterAS option B configuration options. The available options are InterAS option B, InterAS option B (with RFC 3107), and InterAS option B lite. The InterAS option B (with RFC 3107) implementation ensures complete IGP isolation between the data centers and WAN. When BGP advertises a particular route to ASBR, it also distributes the label which is mapped to that route.

Information About InterAS

An autonomous system (AS) is a single network or group of networks that is controlled by a common system administration group and using a single, clearly defined protocol. In many cases, virtual private networks (VPNs) extend to different ASes in different geographical areas. Some VPNs must extend across multiple service providers; these VPNs are called overlapping VPNs. The connection between ASes must be seamless to the customer, regardless of the complexity or location of the VPNs.

InterAS and ASBR

Separate ASes from different service providers can communicate by exchanging information in the form of VPN IP addresses. The ASBRs use EBGP to exchange that information. The IBGP distributes the network layer information for IP prefixes throughout each VPN and each AS. The following protocols are used for sharing routing information:

- Within an AS, routing information is shared using IBGP.
- Between ASes, routing information is shared using EBGP. EBGP allows service providers to set up an interdomain routing system that guarantees loop-free exchange of routing information between separate ASes.

The primary function of EBGP is to exchange network reachability information between ASes, including information about the list of AS routes. The ASes use EBGP border edge routers to distribute the routes, which includes label-switching information. Each border edge router rewrites the next-hop and MPLS labels.

InterAS configuration supported in this MPLS VPN can include an interprovider VPN, which is MPLS VPNs that include two or more ASes, connected by separate border edge routers. The ASes exchange routes use EBGP, and no IBGP or routing information is exchanged between the ASes.

Exchanging VPN Routing Information

ASes exchange VPN routing information (routes and labels) to establish connections. To control connections between ASes, the PE routers and EBGP border edge routers maintain a label forwarding information base (LFIB). The LFIB manages the labels and routes that the PE routers and EBGP border edge routers receive during the exchange of VPN information.

The ASes use the following guidelines to exchange VPN routing information:

- Routing information includes:
  - The destination network.
  - The next-hop field associated with the distributing router.
  - A local MPLS label
- A route distinguisher (RD1) is part of a destination network address. It makes the VPN IP route globally unique in the VPN service provider environment.

The ASBRs are configured to change the next-hop when sending VPN NLRIs to the IBGP neighbors. Therefore, the ASBRs must allocate a new label when they forward the NLRI to the IBGP neighbors.

InterAS Options

Nexus 9508 series switches support the following InterAS options:

- **InterAS option A** - In an interAS option A network, autonomous system border router (ASBR) peers are connected by multiple subinterfaces with at least one interface VPN that spans the two ASes. These ASBRs associate each subinterface with a VPN routing and forwarding (VRF) instance and a BGP session to signal unlabeled IP prefixes. As a result, traffic between the back-to-back VRFs is IP. In this scenario, the VPNs are isolated from each other and, because the traffic is IP Quality of Service (QoS) mechanisms that operate on the IP traffic can be maintained. The downside of this configuration is that one BGP session is required for each subinterface (and at least one subinterface is required for each VPN), which causes scalability concerns as the network grows.
- **InterAS option B** - In an interAS option B network, ASBR ports are connected by one or more subinterfaces that are enabled to receive MPLS traffic. A Multiprotocol Border Gateway Router (MP-BGP) session distributes labeled VPN prefixes between the ASBRs. As a result, the traffic that flows between the ASBRs is labeled. The downside of this configuration is that, because the traffic is MPLS, QoS mechanisms that are applied only to IP traffic cannot be carried and the VRFs cannot be isolated. InterAS option B provides better scalability than option A because it requires only one BGP session to exchange all VPN prefixes between the ASBRs. Also, this feature provides nonstop forwarding (NSF) and Graceful Restart. The ASBRs must be directly connected in this option.

Some functions of option B are noted below:

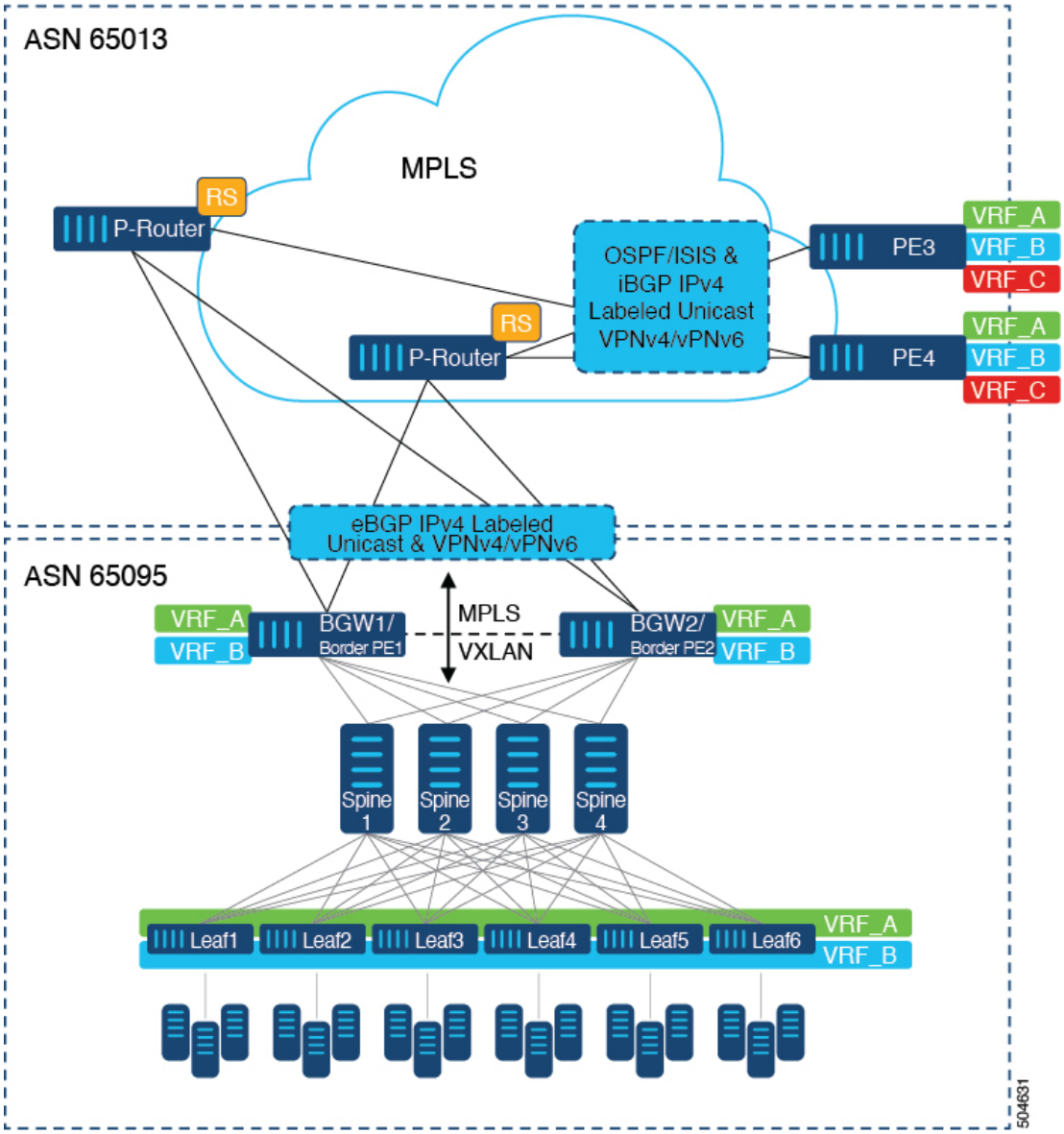
- You can have an IBGP VPNv4/v6 session between Nexus 9508 series switches within an AS and you can have an EBGP VPNv4/v6 session between data center edge routers and WAN routers.
- There is no requirement for a per VRF IBGP session between data center edge routers, like in the lite version.
- – LDP distributes IGP labels between ASBRs.
- **InterAS option B (with BGP-3107 or RFC 3107 implementation)**
- You can have an IBGP VPNv4/v6 implementation between Nexus 9508 switches within an AS and you can have an EBGP VPNv4/v6 session between data center edge routers and WAN routers.
- BGP-3107 enables BGP packets to carry label information without using LDP between ASBRs.
- The label mapping information for a particular route is piggybacked in the same BGP update message that is used to distribute the route itself.
- When BGP is used to distribute a particular route, it also distributes an MPLS label which is mapped to that route. Many ISPs prefer this method of configuration since it ensures complete IGP isolation between the data centers.
- **InterAS option B lite** – Support for the InterAS option B feature is restricted in the Cisco NX-OS 6.2(2) release. Details are noted in the Configuring InterAS Option B (lite version) section.

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

Data Center (DC) deployments have adopted VXLAN EVPN for its benefits such as EVPN control-plane learning, multitenancy, seamless mobility, redundancy, and easier horizontal scaling. Similarly, the Core network transitions to different technologies with their respective capabilities. MPLS with Label Distribution Protocol (LDP) and Layer-3 VPN (L3VPN) is present in many Core networks interconnecting Data Centers.

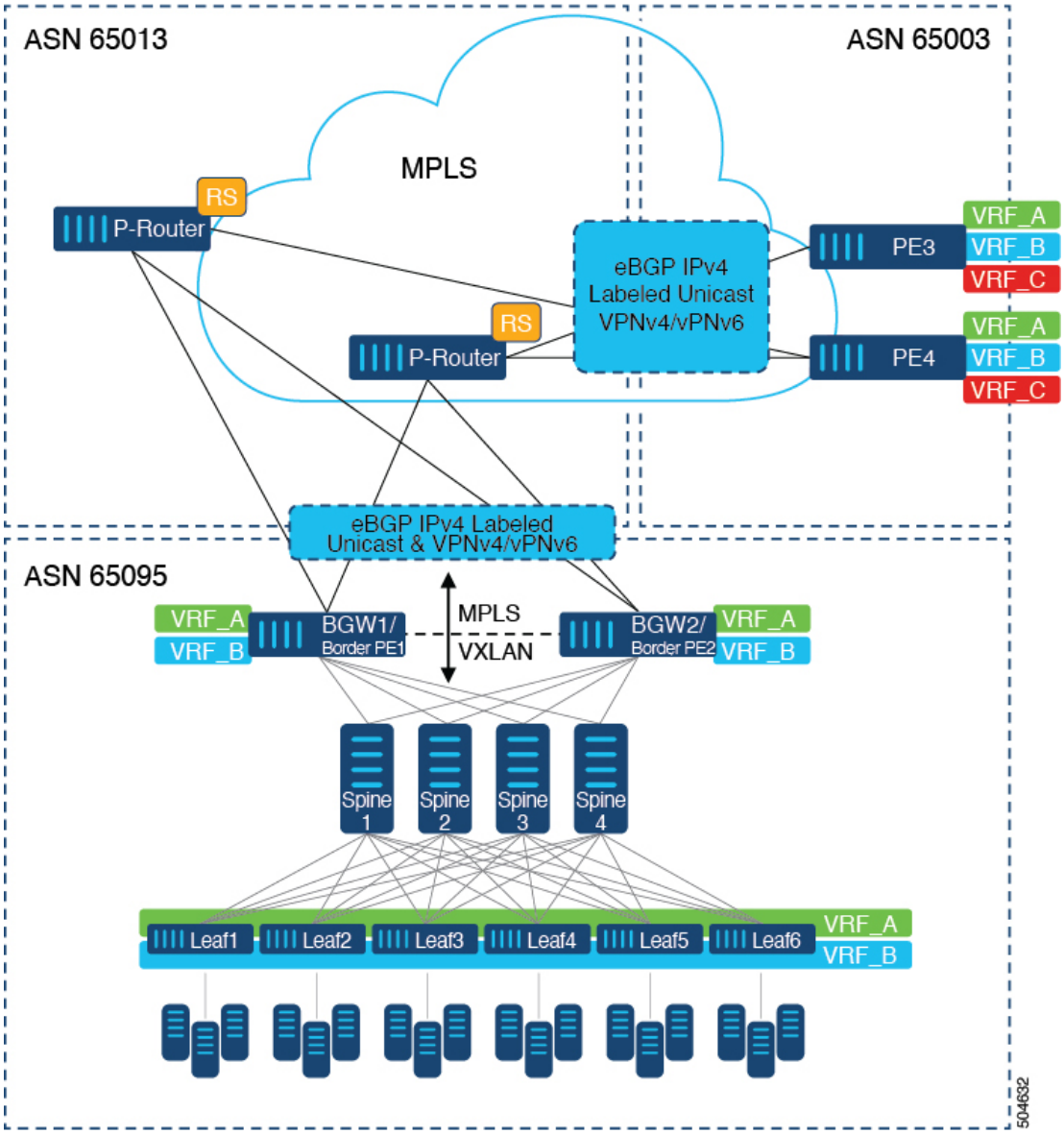
With the data center (DC) established on VXLAN EVPN and the Core network requiring multitenant capable transport, there is a natural necessity to seamless integration. To provide this seamless integration between different control-plane protocols and encapsulations, in this case here from VXLAN to an MPLS-based Core network, the Cisco Nexus 9000 Series Switch provides the Border Provider Edge (Border PE) capability by interfacing the Data Center and the Core routers (Provider Routers or Provider Edge-Routers).

Figure 1: Topology with DC to Core Network Domain Separation



In the above figure, a single Data Center Fabric running VXLAN EVPN is depicted. The VRFs (VRF_A, VRF_B) present in the Data Center require to be extended over a WAN/Core running MPLS. The Data Center Fabrics Border switches acts as Border Gateway/Border Provider Edge (BGW1/Border PE1, BGW2/Border PE2) interconnecting VXLAN BGP EVPN with the MPLS network using L3VPN (VPNv4/VPNv6). The BPEs are interconnected with the Provider Router (P-Router) via eBGP using the IPv4 Labeled-Unicast and VPNv4/VPNv6 Address-Family (AF). The P-Router act as BGP Route-Reflector for the mentioned AF and relays the necessary routes to the MPLS Provider Edge (PE3, PE4) via iBGP. Beyond the usage of BGP as the control-plane, between the MPLS nodes within the same Autonomous System (AS) uses a IGP (OSPF or IS-IS) for label distribution. From the PEs shown in the above figure (PE3, PE4), Inter-AS Option A can be used to extend the Data Center or Core network VRFs to another external network. Even as this diagram shows only one Data Center, the MPLS network can be used to interconnect multiple Data Center Fabrics.

Figure 2: Multiple Administrative Domains Within the Core Network



An alternative deployment scenario is when the Core network is separate into multiple Administrative Domains or Autonomous Systems (AS). In the above figure, a single Data Center Fabric running VXLAN EVPN is depicted. The VRFs (VRF_A, VRF_B) present in the Data Center requires to be extended over a WAN/Core running MPLS. The Data Center Fabrics Border switches acts as Border Gateway/Border Provider Edge (BGW1/Border PE1, BGW2/Border PE2) interconnecting VXLAN BGP EVPN with the MPLS network using L3VPN (VPNv4/VPNv6). The BPEs are interconnected with the Provider Router (P-Router) via eBGP using the IPv4 Labeled-Unicast and VPNv4/VPNv6 Address-Family (AF). The P-Router act as BGP Route Server for the mentioned AF and relays the necessary routes to the MPLS Provider Edge (PE3, PE4) via eBGP; no other control-plane protocol is used between the MPLS nodes. Similar as in the previous scenario, the PEs (PE3, PE4) can operate with Inter-AS Option A to extend the Data Center or Core network VRFs to the external network. Even as this diagram shows only one Data Center, the MPLS network can be used to interconnect multiple Data Center Fabrics.

Guidelines and Limitations for Configuring InterAS Option B

InterAS Option B has the following guidelines and limitations:

- InterAS option B is not supported with BGP confederation AS.
- InterAS option B is supported on Cisco Nexus 9500 platform switches with -R line cards.
- Beginning with Cisco NX-OS release 10.3(2)F, InterAS option B (with BGP-3107 or RFC 3107 implementation) is supported on Nexus 9300-FX/FX2/FX3/GX/GX2 and Cisco 9500 platform switches with -FX or -GX line cards with following limitations:
  - Only imposition of InterAS label for PUSH operation (IP to MPLS or VxLAN decap and MPLS encapsulation of InterAS label) is supported.
  - MPLS Label SWAP operation of InterAS label would not be supported and MPLS switching would not happen.

Configuring BGP for InterAS Option B

Configure DC Edge switches with IBGP & EBGP VPNv4/v6 with the following steps:

Before you begin

To configure BGP for InterAS option B, you need to enable this configuration on both the IBGP and EBGP sides. Refer to Figure 1 for reference.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**router bgp** *as-number* **Example:** `switch(config)# router bgp 100`	Enters the router BGP configuration mode and assigns an autonomous system (AS) number to the local BGP speaker device.
Step 3	**neighbor** *ip-address* **Example:** `switch(config-router)# neighbor 10.0.0.2`	Adds an entry to the BGP or multiprotocol BGP neighbor table, and enters router BGP neighbor configuration mode.
Step 4	**remote-as** *as-number* **Example:** `switch(config-router-neighbor)# remote-as 200`	The as-number argument specifies the autonomous system to which the neighbor belongs.

	Command or Action	Purpose
Step 5	**address-family {vpnv4 | vpnv6} unicast** **Example:** `switch(config-router-neighbor)# address-family vpnv4 unicast`	Enters address family configuration mode for configuring IP VPN sessions.
Step 6	**send-community {both | extended}** **Example:** `switch(config-router-neighbor-af)# send-community both`	Specifies that a communities attribute should be sent to both BGP neighbors.
Step 7	**retain route-target all** **Example:** `switch(config-router-neighbor-af)# retain route-target all`	(Optional). Retains VPNv4/v6 address configuration on the ASBR without VRF configuration. **Note** If you have a VRF configuration on the ASBR, this command is not required.
Step 8	**import l2vpn evpn reoriginate** **Example:** `switch(config-router-neighbor-af)# import l2vpn evpn reoriginate`	Configures import of routing information from the Layer 3 VPN BGP NLRIs that has route target identifier matching the normal route target identifier and exports this routing information after reorigination that assigns it with a stitching route target identifier, to the BGP EVPN neighbor.
Step 9	**vrf** *vrf-name* **Example:** `switch(config-router-neighbor-af)# vrf VPN1`	Associates the BGP process with a VRF.
Step 10	**address-family {ipv4 | ipv6} unicast** **Example:** `switch(config-router-vrf)# address-family ipv4 unicast`	Specifies the IPv4 or IPv6 address family and enters address family configuration mode.
Step 11	**exit** **Example:** `switch(config-vrf-af)# exit`	Exits IPv4 address family.
Step 12	**copy running-config startup-config** **Example:** `switch(config-router-vrf)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring Seamless Integration of EVPN with L3VPN (MPLS)

The following procedure for Border Provider Edge (Border PE) imports and reoriginates the routes from the VXLAN domain to the MPLS domain and in the other direction.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**feature-set mpls** **Example:** `switch(config)# feature-set mpls`	Enables the MPLS feature set.
Step 3	**nv overlay evpn** **Example:** `switch(config)# nv overlay evpn`	Enables VXLAN.
Step 4	**feature bgp** **Example:** `switch(config)# feature bgp`	Enables BGP.
Step 5	**feature mpls l3vpn** **Example:** `switch(config)# feature mpls l3vpn`	Enables Layer 3 VPN.
Step 6	**feature interface-vlan** **Example:** `switch(config)# feature interface-vlan`	Enables the interface VLAN.
Step 7	**feature vn-segment-vlan-based** **Example:** `switch(config)# feature vn-segment-vlan-based`	Enables the VLAN-based VN segment.
Step 8	**feature nv overlay** **Example:** `switch(config)# feature nv overlay`	Enables VXLAN.
Step 9	**router bgp** *autonomous-system-number* **Example:** `switch(config)# router bgp 65095`	Configures BGP. The value of *autonomous-system-number* is from 1 to 4294967295.

	Command or Action	Purpose
Step 10	**address-family ipv4 unicast** **Example:** `switch(config-router)# address-family ipv4 unicast`	Configures the address family for IPv4.
Step 11	**network** *address* **Example:** `switch(config-router-af)# network 10.51.0.51/32`	Injects prefixes into BGP for the MPLS-SR domain. **Note** All viable next-hops for MPLS-SR tunnel deposition on the Border PE must be advertised via the network statement (/32 only).
Step 12	**allocate-label all** **Example:** `switch(config-router-af)# allocate-label all`	Configures label allocation for every prefix injected via the network statement.
Step 13	**exit** **Example:** `switch(config-router-af)# exit`	Exits command mode.
Step 14	**neighbor** *address* **remote-as** *number* **Example:** `switch(config-router)# neighbor 10.95.0.95 remote-as 65095`	Defines the iBGP neighbor IPv4 address and remote Autonomous-System (AS) number towards the Route-Reflector.
Step 15	**update-source** *type/id* **Example:** `switch(config-router)# update-source loopback0`	Defines the interface for eBGP peering.
Step 16	**address-family l2vpn evpn** **Example:** `switch(config-router)# address-family l2vpn evpn`	Configures the L2VPN EVPN address family.
Step 17	**send-community both** **Example:** `switch(config-router-af)# send-community both`	Configures the community for BGP neighbors.
Step 18	**import vpn unicast reoriginate** **Example:** `switch(config-router-af)# import vpn unicast reoriginate`	Reoriginates the route with a new Route-Target. It can be extended to use an optional route-map.

	Command or Action	Purpose
Step 19	**exit** **Example:** `switch(config-router-af)# exit`	Exits command mode.
Step 20	**neighbor** *address* **remote-as** *number* **Example:** `switch(config-router)# neighbor 10.51.131.131 remote-as 65013`	Defines the eBGP neighbor IPv4 address and remote Autonomous-System (AS) number towards the P-Router.
Step 21	**update-source** *type/id* **Example:** `switch(config-router)# update-source Ethernet1/1`	Defines the interface for eBGP peering.
Step 22	**address-family ipv4 labeled-unicast** **Example:** `switch(config-router)# address-family ipv4 labeled-unicast`	Configures the address family for IPv4 labeled-unicast.
Step 23	**send-community both** **Example:** `switch(config-router-af)# send-community both`	Configures the community for BGP neighbors.
Step 24	**exit** **Example:** `switch(config-router-af)# exit`	Exits command mode.
Step 25	**neighbor** *address* **remote-as** *number* **Example:** `switch(config-router)# neighbor 10.131.0.131 remote-as 65013`	Defines the eBGP neighbor IPv4 address and remote Autonomous-System (AS) number.
Step 26	**update-source** *type/id* **Example:** `switch(config-router)# update-source loopback0`	Defines the interface for eBGP peering.
Step 27	**ebgp-multihop** *number* **Example:** `switch(config-router)# ebgp-multihop 5`	Specifies multihop TTL for the remote peer. The range of *number*is from 2 to 255.
Step 28	**address-family vpnv4 unicast** **Example:** `switch(config-router)# address-family vpnv4 unicast`	Configures the address family for VPNv4 or VPNv6.

	Command or Action	Purpose
Step 29	**send-community both** **Example:** `switch(config-router-af)# send-community both`	Configures the community for BGP neighbors.
Step 30	**import l2vpn evpn reoriginate** **Example:** `switch(config-router-af)# import l2vpn evpn reoriginate`	Reoriginates the route with a new Route-Target. It can be extended to use an optional route-map.
Step 31	**exit** **Example:** `switch(config-router-af)# exit`	Exits command mode.

Configuring BGP for InterAS Option B (with RFC 3107 implementation)

Configure DC Edge switches with IBGP & EBGP VPNv4/v6 along with BGP labeled unicast family with following steps:

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**router bgp** *as-number* **Example:** `switch(config)# router bgp 100`	Enters the router BGP configuration mode and assigns an autonomous system (AS) number to the local BGP speaker device.
Step 3	**address-family {vpnv4 | vpnv6} unicast** **Example:** `switch(config-router-neighbor)# address-family vpnv4 unicast`	Enters address family configuration mode for configuring IP VPN sessions.
Step 4	**redistribute direct route-map** *tag* **Example:** `switch(config-router-af)# redistribute direct route-map loopback`	Redistributes directly connected routes using the Border Gateway Protocol.

	Command or Action	Purpose
Step 5	**allocate-label all** **Example:** `switch(config-router-af)# allocate-label all`	Configures ASBRs with the BGP labeled unicast address family to advertise labels for the connected interface.
Step 6	**exit** **Example:** `switch(config-router-af)# exit`	Exits address family router configuration mode and enters router BGP configuration mode.
Step 7	**neighbor** *ip-address* **Example:** `switch(config-router)# neighbor 10.1.1.1`	Configures the BGP neighbor's IP address, and enters router BGP neighbor configuration mode.
Step 8	**remote-as** *as-number* **Example:** `switch(config-router-neighbor)# remote-as 100`	Specifies the BGP neighbor's AS number.
Step 9	**address-family {ipv4|ipv6} labeled-unicast** **Example:** `switch(config-router-neighbor)# address-family ipv4 labeled-unicast`	Configures the ASBR with the BGP labeled unicast address family to advertise labels for the connected interface. **Note** This is the command that implements RFC 3107.
Step 10	**retain route-target all** **Example:** `switch(config-router-neighbor-af)# retain route-target all`	(Optional). Retains VPNv4/v6 address configuration on the ASBR without VRF configuration. **Note** If you have a VRF configuration on the ASBR, this command is not required.
Step 11	**exit** **Example:** `Switch(config-router-neighbor-af)# exit`	Exits router BGP neighbor address family configuration mode and returns to router BGP configuration mode.
Step 12	**neighbor** *ip-address* **Example:** `switch(config-router)# neighbor 10.1.1.1`	Configures a loopback IP address, and enters router BGP neighbor configuration mode.
Step 13	**remote-as** *as-number* **Example:** `switch(config-router-neighbor)# remote-as 100`	Specifies the BGP neighbor's AS number.

	Command or Action	Purpose
Step 14	**address-family {vpnv4|vpnv6} unicast** **Example:** `switch(config-router-vrf)# address-family ipv4 unicast`	Configures the ASBR with the BGP VPNv4 unicast address family.
Step 15	**exit** **Example:** `switch(config-vrf-af)# exit`	Exits IPv4 address family.
Step 16	**address-family {vpnv4|vpnv6} unicast** **Example:** `switch(config-router-vrf)# address-family ipv4 unicast`	Configures the ASBR with the BGP VPNv4 unicast address family.
Step 17	**Repeat the process with ASBR2**	Configures ASBR2 with option B (RFC 3107) settings and implements complete IGP isolation between the two data centers DC1 and DC2.
Step 18	**copy running-config startup-config** **Example:** `switch(config-router-vrf)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example Configuration for Configuring Seamless Integration of EVPN with L3VPN (MPLS)

Scenario - 1 with DC to Core Network Domain Separation and IGP within MPLS network

The following is a sample CLI configuration that is required to import and reoriginate the routes from the VXLAN domain to the MPLS domain and in the reverse direction. The sample CLI configuration represents only the necessary configuration for the respective roles.

Border PE

```
hostname BL51-N9336FX2
install feature-set mpls

feature-set mpls

feature bgp
feature mpls l3vpn
feature ospf
feature interface-vlan
feature vn-segment-vlan-based
feature nv overlay

nv overlay evpn
```

```
mpls label range 16000 23999 static 6000 8000

vlan 2000
  vn-segment 50000

vrf context VRF_A
  vni 50000
  rd auto
  address-family ipv4 unicast
    route-target both auto
    route-target both auto evpn
    route-target import 50000:50000
    route-target export 50000:50000
  address-family ipv6 unicast
    route-target both auto
    route-target both auto evpn
    route-target import 50000:50000
    route-target export 50000:50000

interface Vlan2000
  no shutdown
  vrf member VRF_A
  no ip redirects
  ip forward
  ipv6 address use-link-local-only
  no ipv6 redirects

interface nve1
  no shutdown
  host-reachability protocol bgp
  source-interface loopback1
  member vni 50000 associate-vrf

interface Ethernet1/1
  description TO_P-ROUTER
  ip address 10.51.131.51/24
  mpls ip forwarding
  no shutdown

interface Ethernet1/36
  description TO_SPINE
  ip address 10.95.51.51/24
  ip router ospf 10 area 0.0.0.0
  no shutdown

interface loopback0
  description ROUTER-ID
  ip address 10.51.0.51/32
  ip router ospf UNDERLAY area 0.0.0.0

interface loopback1
  description NVE-LOOPBACK
  ip address 10.51.1.51/32
  ip router ospf UNDERLAY area 0.0.0.0

router ospf UNDERLAY
  router-id 10.51.0.51

router bgp 65095
  address-family ipv4 unicast
    network 10.51.0.51/32
    allocate-label all
!
```

```
  neighbor 10.95.0.95
    remote-as 65095
    update-source loopback0
    address-family l2vpn evpn
      send-community
      send-community extended
      import vpn unicast reoriginate
!
  neighbor 10.51.131.131
    remote-as 65013
    update-source Ethernet1/1
    address-family ipv4 labeled-unicast
      send-community
      send-community extended
!
  neighbor 10.131.0.131
    remote-as 65013
    update-source loopback0
    ebgp-multihop 5
    address-family vpnv4 unicast
      send-community
      send-community extended
      import l2vpn evpn reoriginate
    address-family vpnv6 unicast
      send-community
      send-community extended
      import l2vpn evpn reoriginate

!
  vrf VRF_A
    address-family ipv4 unicast
      redistribute direct route-map fabric-rmap-redist-subnet
```

P-Router

```
hostname P131-N9336FX2
install feature-set mpls

feature-set mpls

feature bgp
feature isis
feature mpls l3vpn

mpls label range 16000 23999 static 6000 8000

route-map RM_NH_UNCH permit 10
  set ip next-hop unchanged

interface Ethernet1/1
  description TO_BORDER-PE
  ip address 10.51.131.131/24
  ip router isis 10
  mpls ip forwarding
  no shutdown

interface Ethernet1/11
  description TO_PE
  ip address 10.52.131.131/24
  ip router isis 10
  mpls ip forwarding
  no shutdown

interface loopback0
  description ROUTER-ID
```

```
  ip address 10.131.0.131/32
  ip router isis 10

router isis 10
  net 49.0000.0000.0131.00
  is-type level-2
  address-family ipv4 unicast
    segment-routing mpls

router bgp 65013
  event-history detail
  address-family ipv4 unicast
    allocate-label all
!
  neighbor 10.51.131.51
    remote-as 65095
    update-source Ethernet1/1
    address-family ipv4 labeled-unicast
      send-community
      send-community extended
!
  neighbor 10.51.0.51
    remote-as 65095
    update-source loopback0
    ebgp-multihop 5
    address-family vpnv4 unicast
      send-community
      send-community extended
      route-map RM_NH_UNCH out
    address-family vpnv6 unicast
      send-community
      send-community extended
      route-map RM_NH_UNCH out
!
  neighbor 10.52.131.52
    remote-as 65013
    update-source Ethernet1/11
    address-family ipv4 labeled-unicast
      send-community
      send-community extended
!
  neighbor 10.52.0.52
    remote-as 65013
    update-source loopback0
    address-family vpnv4 unicast
      send-community
      send-community extended
      route-reflector-client
      route-map RM_NH_UNCH out
    address-family vpnv6 unicast
      send-community
      send-community extended
      route-reflector-client
      route-map RM_NH_UNCH out
```

Provider Edge (PE)

```
hostname L52-N93240FX2
install feature-set mpls

feature-set mpls

feature bgp
feature isis
feature mpls l3vpn
```

```
mpls label range 16000 23999 static 6000 8000

vrf context VRF_A
  rd auto
  address-family ipv4 unicast
    route-target import 50000:50000
    route-target export 50000:50000
  address-family ipv6 unicast
    route-target import 50000:50000
    route-target export 50000:50000

interface Ethernet1/49
  description TO_P-ROUTER
  ip address 10.52.131.52/24
  ip router isis 10
  mpls ip forwarding
  no shutdown

interface loopback0
  description ROUTER-ID
  ip address 10.52.0.52/32
  ip router isis 10

router isis 10
  net 49.0000.0000.0052.00
  is-type level-2
  address-family ipv4 unicast
    segment-routing mpls

router bgp 65013
  address-family ipv4 unicast
    network 10.52.0.52/32
    allocate-label all
!
  neighbor 10.52.131.131
    remote-as 65013
    update-source Ethernet1/49
    address-family ipv4 labeled-unicast
      send-community
      send-community extended
!
  neighbor 10.131.0.131
    remote-as 65013
    update-source loopback0
    address-family vpnv4 unicast
      send-community
      send-community extended
    address-family vpnv6 unicast
      send-community
      send-community extended
!
  vrf VRF_A
    address-family ipv4 unicast
      redistribute direct route-map fabric-rmap-redist-subnet
```

Scenario - 2 with DC to Core and within Core Network Domain Separation (eBGP within MPLS network)

The following is a sample CLI configuration that is required to import and reoriginate the routes from the VXLAN domain to the MPLS domain and in the reverse direction. The sample CLI configuration represents only the nodes that are different from Scenario #1, which are the P-Router and the Provider Edge (PE) roles. The Border PE remains the same for both scenarios.

P-Router

```
hostname P131-N9336FX2
install feature-set mpls

feature-set mpls

feature bgp
feature mpls l3vpn

mpls label range 16000 23999 static 6000 8000

route-map RM_NH_UNCH permit 10
  set ip next-hop unchanged

interface Ethernet1/1
  description TO_BORDER-PE
  ip address 10.51.131.131/24
  mpls ip forwarding
  no shutdown

interface Ethernet1/11
  description TO_PE
  ip address 10.52.131.131/24
  mpls ip forwarding
  no shutdown

interface loopback0
  description ROUTER-ID
  ip address 10.131.0.131/32
  ip router isis 10

router bgp 65013
  event-history detail
  address-family ipv4 unicast
    network 10.131.0.131/32
    allocate-label all
!
  address-family vpnv4 unicast
    retain route-target all
  address-family vpnv6 unicast
    retain route-target all
!
  neighbor 10.51.131.51
    remote-as 65095
    update-source Ethernet1/1
    address-family ipv4 labeled-unicast
      send-community
      send-community extended
!
  neighbor 10.51.0.51
    remote-as 65095
    update-source loopback0
    ebgp-multihop 5
    address-family vpnv4 unicast
      send-community
      send-community extended
      route-map RM_NH_UNCH out
    address-family vpnv6 unicast
      send-community
      send-community extended
      route-map RM_NH_UNCH out
!
  neighbor 10.52.131.52
```

```
    remote-as 65003
    update-source Ethernet1/11
    address-family ipv4 labeled-unicast
      send-community
      send-community extended
!
  neighbor 10.52.0.52
    remote-as 65003
    update-source loopback0
    ebgp-multihop 5
    address-family vpnv4 unicast
      send-community
      send-community extended
      route-map RM_NH_UNCH out
    address-family vpnv6 unicast
      send-community
      send-community extended
      route-map RM_NH_UNCH out
```

Provider Edge (PE)

```
hostname L52-N93240FX2
install feature-set mpls

feature-set mpls

feature bgp
feature mpls l3vpn

mpls label range 16000 23999 static 6000 8000

vrf context VRF_A
  rd auto
  address-family ipv4 unicast
    route-target import 50000:50000
    route-target export 50000:50000
  address-family ipv6 unicast
    route-target import 50000:50000
    route-target export 50000:50000

interface Ethernet1/49
  description TO_P-ROUTER
  ip address 10.52.131.52/24
  mpls ip forwarding
  no shutdown

interface loopback0
  description ROUTER-ID
  ip address 10.52.0.52/32
  ip router isis 10

router bgp 65003
  address-family ipv4 unicast
    network 10.52.0.52/32
    allocate-label all
!
  neighbor 10.52.131.131
    remote-as 65013
    update-source Ethernet1/49
    address-family ipv4 labeled-unicast
      send-community
      send-community extended
!
  neighbor 10.131.0.131
    remote-as 65013
```

```
    update-source loopback0
    ebgp-multihop 5
    address-family vpnv4 unicast
      send-community
      send-community extended
    address-family vpnv6 unicast
      send-community
      send-community extended
!
  vrf VRF_A
    address-family ipv4 unicast
      redistribute direct route-map fabric-rmap-redist-subnet
```