DC-PE Router in Cisco ACI to SR-MPLS Hand-off

SR-MPLS Hand-off is an interconnection option that enables Cisco ACI to WAN interconnect using Segment Routing (SR) MPLS underlay.

From Cisco IOS XE 17.14.1a, Cisco ASR 1000 Series Aggregation Services Routers and Cisco Catalyst 8500 Series Edge Platforms can be used as intermediate DC-PE devices in an ACI to SR-MPLS Hand-off interconnection.

Prerequisites

There are no specific prerequisites for DC-PE Router in ACI to SR-MPLS Hand-off.

Restrictions

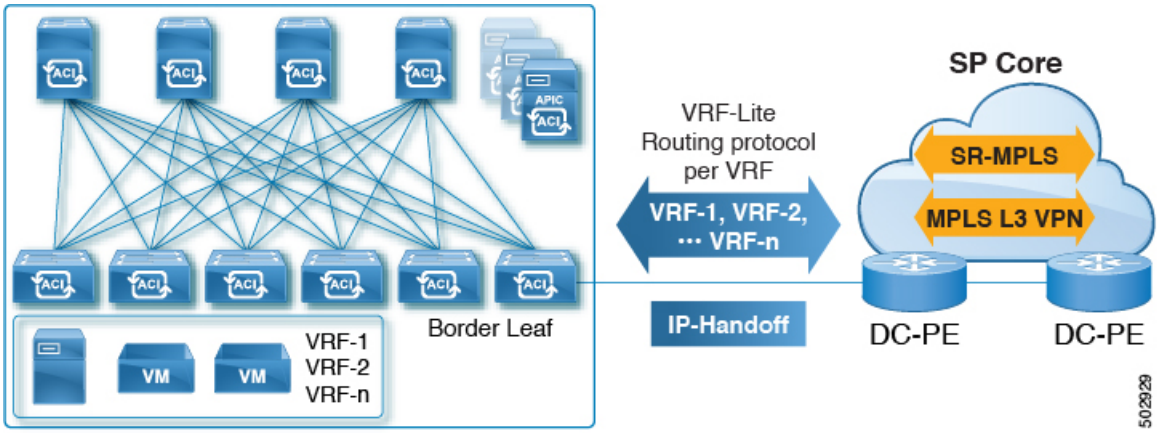
- iBGP is not supported between DC-PE and border/remote leaf.
- The router ID must be unique across all border leaf switches and the DC-PE.

Information About DC-PE Router in ACI to SR-MPLS Hand-off

SR/MPLS Handoff is an interconnection option that enables Cisco ACI fabric to WAN interconnect using Segment Routing (SR) MPLS underlay. SR/MPLS is a better solution than others known solution as it is much more common for an SP core. The solution brings the following benefits:

- Unified transport and policies between DC and SP

- Single Control Plane session for multiple VRFs
- Traffic engineering in the SP core controlled from the DC



For more information about Cisco ACI fabric and the underlying ACI to SR-MPLS hand-off interconnection, see the following publications:

- ACI SRMPLS Handoff Whitepaper
- ACI SRMPLS Architecture:
  1. Validated Design for Cisco ACI to SR-MPLS Handoff - Introduction
  2. Validated Design for Cisco ACI to SR-MPLS Handoff - Tenant Configuration
  3. ACI Fabric L3Out White Paper
- ACI SRMPLS Architecture/ Sample Use Cases

Supported Platforms

From Cisco IOS XE 17.14.1a, the following routers can be configured as DC-PE device in an ACI to SR-MPLS hand-off interconnection:

- Cisco ASR 1000 Series Aggregation Services Routers
- Cisco Catalyst 8500 Series Edge Platforms

How to Configure the DC-PE Router

Perform the following steps to configure the VRF and BGP on the DC-PE router.

Configuring VRF on the DC-PE Router

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vrf definition** *vrf-name*
4. **rd** *vpn-route-distinguisher*
5. **address-family ipv4** [**multicast** | **unicast**]
6. **route-target** {**export** | **import** | **both**} *route-target-ext-community*
7. **route-target** {**export** | **import** | **both**} *route-target-ext-community* **stitching**
8. **exit-address-family**
9. **address-family ipv6** [**multicast** | **unicast**]
10. **route-target** {**export** | **import** | **both**} *route-target-ext-community*
11. **route-target** {**export** | **import** | **both**} *route-target-ext-community* **stitching**
12. **exit-address-family**
13. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enters privileged EXEC mode. Enter password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**vrf definition** *vrf-name* **Example:** `Device(config)# vrf definition test`	Enters the VRF configuration mode for the specified VRF instance.
Step 4	**rd** *vpn-route-distinguisher* **Example:** `Device(config-vrf)#  rd 65000:1`	Specifies the route distinguisher for the VRF instance.
Step 5	**address-family ipv4** [**multicast**	**unicast**] **Example:** `Device(config-vrf)# address-family ipv4`	Enters the IPv4 address family configuration mode.
Step 6	**route-target** {**export**	**import**	**both**} *route-target-ext-community*	Creates a list of import, export, or both import and export route target communities for the specified VRF.

	Command or Action	Purpose

	Example:
	`Device(config-vrf-af)# route-target import 1:1`
	Example:
	`Device(config-vrf-af)# route-target export 2:2`
		Enter either an autonomous system number and an arbitrary number (xxx:y), or an IP address and an arbitrary number (A.B.C.D:y).

Step 7	**route-target** {**export** | **import** | **both**} *route-target-ext-community* **stitching**
	Example:
	`Device(config-vrf-af)# route-target import 3:3 stitching`
	Example:
	`Device(config-vrf-af)# route-target export 4:4 stitching`
		Configures importing, exporting, or both importing and exporting of EVPN route target communities for the VRF.

Step 8	**exit-address-family** **Example:** `Device(config-vrf-af)# exit-address-family`	Exits VRF address family configuration mode and enters VRF configuration mode.
Step 9	**address-family ipv6** [**multicast** | **unicast**] **Example:** `Device(config-vrf)# address-family ipv6`	Enters the IPv6 address family configuration mode.

Step 10	**route-target** {**export** | **import** | **both**} *route-target-ext-community*
	Example:
	`Device(config-vrf-af)# route-target import 1:1`
	Example:
	`Device(config-vrf-af)# route-target export 2:2`
		Creates a list of import, export, or both import and export route target communities for the specified VRF.
		Enter either an autonomous system number and an arbitrary number (xxx:y), or an IP address and an arbitrary number (A.B.C.D:y).

Step 11	**route-target** {**export** | **import** | **both**} *route-target-ext-community* **stitching**
	Example:
	`Device(config-vrf-af)# route-target import 3:3 stitching`
	Example:
	`Device(config-vrf-af)# route-target export 4:4 stitching`
		Configures importing, exporting, or both importing and export of EVPN route target communities for the VRF.

Step 12	**exit-address-family** **Example:** `Device(config-vrf-af)# exit-address-family`	Exits VRF address family configuration mode and enters VRF configuration mode.
Step 13	**end** **Example:** `Device(config-vrf)# end`	Returns to privileged EXEC mode.

Example

The following example demonstrates the VRF configuration required for the DC-PE router:

```
vrf definition test
  rd 65000:1
  address-family ipv4
            route-target import 1:1
            route-target export 2:2
            route-target import 3:3 stitching
            route-target export 4:4 stitching
  exit
  address-family ipv6
            route-target import 1:1
            route-target export 2:2
            route-target import 3:3 stitching
            route-target export 4:4 stitching
  exit
```

Configuring BGP on the DC-PE router.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **routerbgp** *as-number*
4. **neighbor** *dc-border-leaf-address* **remote-as** *number*
5. **neighbor** *wan-router-address* **remote-as** *number*
6. **address-family l2vpn evpn**
7. **import vpnv4 unicast** [**re-originate**]
8. **import vpnv6 unicast** [**re-originate**]
9. **neighbor** *ip-address* **activate**
10. **neighbor** *ip-address* **send-community** [**both** | **extended** | **standard**]
11. **exit-address-family**
12. **address-family vpnv4**
13. **import l2vpn evpn** [**re-originate**]
14. **neighbor** *ip-address* **activate**
15. **neighbor** *ip-address* **send-community** [**both** | **extended** | **standard**]
16. **neighbor** {**ip-address** | **peer-group-name**} **next-hop-self** [**all**]
17. **exit-address-family**
18. **address-family vpnv6**
19. **import l2vpn evpn** [**re-originate**]
20. **neighbor** *ip-address* **activate**
21. **neighbor** *ip-address* **send-community** [**both** | **extended** | **standard**]
22. **neighbor** {**ip-address** | **peer-group-name**} **next-hop-self** [**all**]
23. **exit-address-family**
24. **address-family ipv4 vrf** *vrf-name*
25. **maximum-paths eibgp** *number*

26. **exit-address-family**
27. **address-family ipv6 vrf** *vrf-name*
28. **maximum-paths eibgp** *number*
29. **exit-address-family**
30. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**routerbgp** *as-number* **Example:** `Device(config)# router bgp 1`	Configures a BGP routing process and enters router configuration mode.
Step 4	**neighbor** *dc-border-leaf-address* **remote-as** *number* **Example:** `Device(config-router)# neighbor 1.1.1.1 remote-as 2`	Defines multiprotocol-BGP neighbors in the EVPN network. Use the IP address of the spine switch as the neighbor IP address. This configures the spine switch as a BGP neighbor.
Step 5	**neighbor** *wan-router-address* **remote-as** *number* **Example:** `Device(config-router)# neighbor 2.2.2.2 remote-as 1`	Defines multiprotocol-BGP neighbors in the external MPLS network. Use the IP address of the external MPLS network peer as the neighbor IP address. This configures the external MPLS network peer as a BGP neighbor.
Step 6	**address-family l2vpn evpn** **Example:** `Device(config-router)# address-family l2vpn evpn`	Specifies the L2VPN address family and enters address family configuration mode.
Step 7	**import vpnv4 unicast** [**re-originate**] **Example:** `Device(config-router-af)# import vpnv4 unicast re-originate`	Reoriginates the VPNv4 routes imported from the external peer into the EVPN address family as EVPN routes, and distributes within the EVPN fabric.

	Command or Action	Purpose
Step 8	**import vpnv6 unicast** [**re-originate**] **Example:** `Device(config-router-af)# import vpnv6 unicast re-originate`	Reoriginates the VPNv6 routes imported from the external peer into the EVPN address family as EVPN routes, and distributes within the EVPN fabric.
Step 9	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 1.1.1.1 activate`	Enables the exchange information from a BGP neighbor. Use the IP address of the spine switch as the neighbor IP address.
Step 10	**neighbor** *ip-address* **send-community** [**both** | **extended** | **standard**] **Example:** `Device(config-router-af)# neighbor 1.1.1.1 send-community both`	Specifies the communities attribute sent to a BGP neighbor. Use the IP address of the spine switch as the neighbor IP address. **Note** Use either extended or both keywords. External connectivity cannot be established when you use the standard keyword.
Step 11	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits address family configuration mode and returns to router configuration mode.
Step 12	**address-family vpnv4** **Example:** `Device(config-router)# address-family vpnv4`	Specifies the VPNv4 address family and enters address family configuration mode.
Step 13	**import l2vpn evpn** [**re-originate**] **Example:** `Device(config-router-af)# import l2vpn evpn re-originate stitching-rt`	Reoriginates the EVPN routes imported from the EVPN fabric into the VPNv4 address family as VPNv4 routes and distributes them to the external network.
Step 14	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 2.2.2.2 active`	Enables the exchange information from a BGP neighbor. Use the IP address of the external MPLS network router as the neighbor IP address.
Step 15	**neighbor** *ip-address* **send-community** [**both** | **extended** | **standard**] **Example:** `Device(config-router-af)# neighbor 2.2.2.2 send-community both`	Specifies the communities attribute sent to a BGP neighbor. Use the IP address of the external MPLS network router as the neighbor IP address. **Note** Use either extended or both keywords. External connectivity cannot be established when you use the standard keyword.
Step 16	**neighbor** {**ip-address** | **peer-group-name**} **next-hop-self** [**all**]	Configures the router as the next hop for a BGP-speaking neighbor or peer group.

	Command or Action	Purpose
	Example: `Device(config-router-af)# neighbor 2.2.2.2 next-hop-self all`	The all keyword is mandatory when implementing external connectivity through iBGP, where the EVPN fabric and the MPLS network are in the same BGP autonomous system number. The all keyword is optional when implementing external connectivity through eBGP, where the EVPN fabric and the MPLS network are in different BGP autonomous system numbers
Step 17	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits address family configuration mode and returns to router configuration mode.
Step 18	**address-family vpnv6** **Example:** `Device(config-router)# address-family vpnv6`	Specifies the VPNv6 address family and enters address family configuration mode.
Step 19	**import l2vpn evpn** [**re-originate**] **Example:** `Device(config-router-af)# import l2vpn evpn re-originate stitching-rt`	Reoriginates the EVPN routes imported from the EVPN fabric into the VPNv6 address family as VPNv6 routes and distributes them to the external network.
Step 20	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 2.2.2.2 active`	Enables the exchange information from a BGP neighbor. Use the IP address of the spine switch as the neighbor IP address.
Step 21	**neighbor** *ip-address* **send-community** [**both** | **extended** | **standard**] **Example:** `Device(config-router-af)# neighbor 2.2.2.2 send-community both`	Specifies the communities attribute sent to a BGP neighbor. Use the IP address of the spine switch as the neighbor IP address. **Note** Use either extended or both keywords. External connectivity cannot be established when you use the standard keyword.
Step 22	**neighbor** {**ip-address** | **peer-group-name**} **next-hop-self** [**all**] **Example:** `Device(config-router-af)# neighbor 2.2.2.2 next-hop-self all`	Configures the router as the next hop for a BGP-speaking neighbor or peer group. The all keyword is mandatory when implementing external connectivity through iBGP, where the EVPN fabric and the MPLS network are in the same BGP autonomous system number. The all keyword is optional when implementing external connectivity through eBGP, where the EVPN fabric and the MPLS network are in different BGP autonomous system numbers

	Command or Action	Purpose
Step 23	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits address family configuration mode and returns to router configuration mode.
Step 24	**address-family ipv4 vrf** *vrf-name* **Example:** `Device(config-router)# address-family ipv4 vrf test`	Places the router in address family configuration mode. Separate VRF multipath configurations are isolated by unique route distinguisher.
Step 25	**maximum-paths eibgp** *number* **Example:** `Device(config-router-af)# maximum-paths eibgp 16`	Configures the number of parallel iBGP and eBGP routes that can be installed into a routing table. **Note** You can configure the **maximum-paths eibgp** command only under the IPv4 VRF address family configuration mode
Step 26	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits address family configuration mode and returns to router configuration mode.
Step 27	**address-family ipv6 vrf** *vrf-name* **Example:** `Device(config-router)# address-family ipv6 vrf test`	Places the router in address family configuration mode. Separate VRF multipath configurations are isolated by unique route distinguisher.
Step 28	**maximum-paths eibgp** *number* **Example:** `Device(config-router-af)# maximum-paths eibgp 16`	Configures the number of parallel iBGP and eBGP routes that can be installed into a routing table. **Note** You can configure the **maximum-paths eibgp** command only under the IPv6 VRF address family configuration mode
Step 29	**exit-address-family** **Example:** `Device(config-vrf-af)# exit-address-family`	Exits VRF address family configuration mode and returns to VRF configuration mode.
Step 30	**end** **Example:** `Device(config-vrf)# end`	Returns to privileged EXEC mode.

Example

The following example demonstrates the VRF configuration required for the DC-PE router:

```
router bgp 1
      neighbor 1.1.1.1 remote-as 2
      neighbor 2.2.2.2 remote-as 1
      address-family l2vpn evpn
                import vpnv4 unicast re-originate
                import vpnv6 unicast re-originate
                neighbor 1.1.1.1 active
                neighbor 1.1.1.1 send-community both
      exit
      address-family vpnv4
                import l2vpn evpn re-originate stitching-rt
                neighbor 2.2.2.2 active
                neighbor 2.2.2.2 send-community both
                neighbor 2.2.2.2 next-hop-self all
      exit
      address-family vpnv6
                import l2vpn evpn re-originate stitching-rt
                neighbor 2.2.2.2 active
                neighbor 2.2.2.2 send-community both
                neighbor 2.2.2.2 next-hop-self all
      exit
      address-family ipv4 vrf test
        maximum-paths eibgp 16
      exit
      address-family ipv6 vrf test
        maximum-paths eibgp 16
      exit
```

Verifying DC-PE Router Configuration

This section provides the show commands that can be used to verify the DC-PE router configuration.

Verifying IPv4 and IPv6 Route from ACI

Use the following commands to verify IPv4 route from ACI:

```
Router#show bgp l2vpn evpn route-type 5 0 99.1.2.0 24
BGP routing table entry for [5][2:2][0][24][99.1.2.0]/17, version 2
Paths: (1 available, best #1, table EVPN-BGP-Table)
  Not advertised to any peer
  Refresh Epoch 1
  65000 65001
   2.2.2.2 (via default) from 5.5.5.5 (5.5.5.5)
    Origin incomplete, localpref 100, valid, external, best
    EVPN ESI: 00000000000000000000, Gateway Address: 0.0.0.0, VNI  Label 0, MPLS VPN Label
 19
    Extended Community: RT:2:2 Color:10
    rx pathid: 0, tx pathid: 0x0
    Updated on Feb 27 2024 15:46:31 PST


Router#show bgp vpnv4 uni all 99.1.2.0
BGP routing table entry for 6:6:99.1.2.0/24, version 2
Paths: (1 available, best #1, table red)
  Advertised to update-groups:
   1
  Refresh Epoch 1
  65000 65001, imported path from [5][2:2][0][24][99.1.2.0]/17   (global)
```

```
    2.2.2.2 (via default) from 5.5.5.5 (5.5.5.5)
     Origin incomplete, localpref 100, valid, external, best
     Extended Community: RT:2:2 Color:10
     mpls labels in/out IPv4 VRF Aggr:19/19
     rx pathid: 0, tx pathid: 0x0
     Updated on Feb 27 2024 15:46:31 PST
```

```
Router#show ip route vrf red 99.1.2.0
Routing Table: red
Routing entry for 99.1.2.0/24
Known via "bgp 65100", distance 20, metric 0
Tag 65000, type external
Last update from 2.2.2.2 00:07:23 ago
Routing Descriptor Blocks:
* 2.2.2.2 (default), from 5.5.5.5, 00:07:23 ago
opaque_ptr 0x7F055237F160
Route metric is 0, traffic share count is 1
AS Hops 2
Route tag 65000
MPLS label: 19
```

Use the following commands to verify IPv6 route from ACI:

```
Router#show bgp l2vpn evpn route-type 5 0 2001::99:1:2:0 112
BGP routing table entry for [5][2:2][0][112][2001::99:1:2:0]/29, version 4
Paths: (1 available, best #1, table EVPN-BGP-Table)
  Not advertised to any peer
  Refresh Epoch 1
  65000 65001
   2.2.2.2 (via default) from 5.5.5.5 (5.5.5.5)
    Origin incomplete, localpref 100, valid, external, best
    EVPN ESI: 00000000000000000000, Gateway Address:::,VNI Label 0,MPLS VPN Label 21
    Extended Community: RT:2:2 Color:10
    rx pathid: 0, tx pathid: 0x0
    Updated on Feb 27 2024 15:46:31 PST
```

```
Router#show bgp vpnv6 uni all 2001::99:1:2:0/112
BGP routing table entry for [6:6]2001::99:1:2:0/112, version 2
Paths: (1 available, best #1, table red)
  Advertised to update-groups:
   1
  Refresh Epoch 1
  65000 65001, imported path from [5][2:2][0][112][2001::99:1:2:0]/29 (global)
   ::FFFF:2.2.2.2 (via default) from 5.5.5.5 (5.5.5.5)
     Origin incomplete, localpref 100, valid, external, best
     Extended Community: RT:2:2 Color:10
     mpls labels in/out IPv6 VRF Aggr:20/21
     rx pathid: 0, tx pathid: 0x0
     Updated on Feb 27 2024 15:46:31 PST
```

```
Router#show ipv6 route vrf red 2001::99:1:2:0/112
Routing entry for 2001::99:1:2:0/112
Known via "bgp 65100", distance 20, metric 0
Tag 65000, type external
Route count is 1/1, share count 0
Routing paths:
2.2.2.2%default indirectly connected
Route metric is 0, traffic share count is 1
MPLS label: 21
From ::FFFF:5.5.5.5
opaque_ptr 0x7F05523C42C8
Last updated 00:10:33 ago
```

Verifying IPv4 and IPv6 Route from WAN

Use the following commands to verify IPv4 route from WAN:

```
Router#show bgp vpnv4 uni vrf red 13.13.13.13
BGP routing table entry for 6:6:13.13.13.13/32, version 19
Paths: (1 available, best #1, table red)
Flag: 0x100
  Not advertised to any peer
  Refresh Epoch 1
  65013, imported path from 12:12:13.13.13.13/32 (global)
    12.12.12.12 (metric 30) (via default) from 7.7.7.7 (7.7.7.7)
     Origin incomplete, metric 0, localpref 100, valid, internal,  best
     Extended Community: RT:12:12 Color:10
     Originator: 12.12.12.12, Cluster list: 7.7.7.7
     mpls labels in/out nolabel/18
     binding SID: 22 (color - 10) (state - UP)
     rx pathid: 0, tx pathid: 0x0
     Updated on Feb 27 2024 15:46:32 PST


Router#show bgp l2vpn evpn route-type 5 0 13.13.13.13 32
BGP routing table entry for [5][6:6][0][32][13.13.13.13]/17, version 18
Paths: (1 available, best #1, table EVPN-BGP-Table)
Advertised to update-groups:
   1
  Refresh Epoch 1
  65013, imported path from base
   12.12.12.12 (metric 30) (via default) from 7.7.7.7 (7.7.7.7)
   Origin incomplete, metric 0, localpref 100, valid, internal, best
   EVPN ESI: 00000000000000000000, Gateway Address: 0.0.0.0, local   vtep: 0.0.0.0, VNI
Label 0, MPLS VPN Label 18, MPLS VPN Local Label  19
   Extended Community: RT:2:2 RT:4:4 Color:10
   Originator: 12.12.12.12, Cluster list: 7.7.7.7
   rx pathid: 0, tx pathid: 0x0
   Updated on Feb 27 2024 15:46:32 PST


Router#show ip route vrf red 13.13.13.13
Routing Table: red
Routing entry for 13.13.13.13/32
Known via "bgp 65100", distance 200, metric 0
Tag 65013, type internal
Routing Descriptor Blocks:
* Binding Label: 22, from 7.7.7.7, 00:07:48 ago
opaque_ptr 0x7F055237ED70
Route metric is 0, traffic share count is 1
AS Hops 1
Route tag 65013
MPLS label: 18
MPLS Flags: MPLS Required
```

Use the following commands to verify IPv6 route from WAN:

```
Router#show bgp vpnv6 uni vrf red 2001::13:13:13:13/128
BGP routing table entry for [6:6]2001::13:13:13:13/128, version 19
Paths: (1 available, best #1, table red)
Flag: 0x100
  Not advertised to any peer
  Refresh Epoch 1
  65013, imported path from [12:12]2001::13:13:13:13/128 (global)
   ::FFFF:12.12.12.12 (metric 30) (via default) from 7.7.7.7 (7.7.7.7)
   Origin incomplete, metric 0, localpref 100, valid, internal, best
```

```
   Extended Community: RT:12:12 Color:10
   Originator: 12.12.12.12, Cluster list: 7.7.7.7
   mpls labels in/out nolabel/20
   binding SID: 22 (color - 10) (state - UP)
   rx pathid: 0, tx pathid: 0x0
   Updated on Feb 27 2024 15:46:32 PST
```

```
Router#show bgp l2vpn evpn route-type 5 0 2001::13:13:13:13 128
BGP routing table entry for [5][6:6][0][128][2001::13:13:13:13]/29, version 12
Paths: (1 available, best #1, table EVPN-BGP-Table)
   Advertised to update-groups:
     1
   Refresh Epoch 1
   65013, imported path from base
   ::FFFF:12.12.12.12 (metric 30) (via default) from 7.7.7.7 (7.7.7.7)
     Origin incomplete, metric 0, localpref 100, valid, internal, best
     EVPN ESI: 00000000000000000000, Gateway Address: ::, local vtep: 0.0.0.0, VNI Label
0, MPLS VPN Label 20, MPLS VPN Local Label 20
     Extended Community: RT:2:2 RT:4:4 Color:10
     Originator: 12.12.12.12, Cluster list: 7.7.7.7
     rx pathid: 0, tx pathid: 0x0
     Updated on Feb 27 2024 15:46:32 PST
```

```
Router#show ipv6 route vrf red 2001::13:13:13:13/128
Routing entry for 2001::13:13:13:13/128
Known via "bgp 65100", distance 200, metric 0
Tag 65013, type internal
Route count is 1/1, share count 0
Routing paths:
Bind Label: 22 indirectly connected
Route metric is 0, traffic share count is 1
MPLS label: 20
From ::FFFF:7.7.7.7
opaque_ptr 0x7F05523C3ED8
Last updated 00:10:03 ago
```

Troubleshooting and Debugging

The following debug commands can be used to enable the debugs required for debugging BGP Label Manager:

```
debug bgp lmm address-family vpnv4
debug bgp lmm address-family vpnv6
```

The following example shows the output of the **debug bgp lmm address-family vpnv4/6** command:

```
*Jul 18 21:32:09.835: BGP_LMM (VPNv4): Add update info for 1:1:3.3.3.0/24, neighbor 1.1.1.3,
 NH  unchanged (no), topology neighbor labeled (yes)
*Jul 18 21:34:48.577: BGP_LMM (VPNv6): Add update info for [1:1]3333::/120, neighbor 1.1.1.3,
 NH unchanged (no), topology neighbor labeled (yes)
*Jul 18 21:32:09.835: BGP_LMM (VPNv4): Allocated and installed a per VRF aggregate label
10 for vrf  red, address family ipv4"
*Jul 18 21:32:09.835: BGP_LMM (VPNv4): Allocated and installed a per VRF aggregate label
11 for vrf  red, address family ipv6"
```

The following debug commands can be used to debug BGP EVPN to L3VPN import/re-origination:

```
debug bgp all import updates
debug bgp all import events
```

The following example shows the output of the **debug bp all import** command:

```
*Jul 21 14:31:22.693: BGP VPN-IMP: red:VPNv4 Unicast:base 1:1:3.3.3.0/24 Exporting doing
PATHS.
*Jul 21 14:31:22.693: BGP VPN-IMP: VPNv4 Unicast:base Building ETL from VPN
*Jul 21 14:31:22.693: BGP VPN-IMP: VPNv4 Unicast:base GBL Building ETL.
*Jul 21 14:31:22.693: BGP VPN-IMP: VPNv4 Unicast:base -> global:IPv4 Unicast:base Creating
 Import Topo.
*Jul 21 14:31:22.693: BGP VPN-IMP: VPNv4 Unicast:base -> global:IPv4 Unicast:base GBL Adding
 topology IPv4 Unicast to ETL.
*Jul 21 14:31:22.693: BGP VPN-IMP: VPNv4 Unicast:base -> global:IPv4 Multicast:base Creating
 Import Topo.
*Jul 21 14:31:22.693: BGP VPN-IMP: VPNv4 Unicast:base -> global:IPv4 Multicast:base GBL
Adding to ETL.
*Jul 21 14:31:22.693: BGP VPN-IMP: VPNv4 Unicast:base Building GBL ETL done.
*Jul 21 14:31:22.693: BGP VPN-IMP: VPNv4 Unicast:base L2VPN E-VPN AF_PRIV Building ETL.
```

Feature Information for DC-PE Router in Cisco ACI to SR-MPLS Hand-off

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.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 1: Feature Information for DC-PE Router in Cisco ACI to SR-MPLS Hand-off

Feature Name	Releases	Feature Information
DC-PE Router in Cisco ACI to SR-MPLS Hand-off	Cisco IOS XE 17.14.1a	From Cisco IOS XE 17.14.1a, Cisco ASR 1000 Series Aggregation Services Routers and Cisco Catalyst 8500 Series Edge Platforms can be used as intermediate DC-PE devices in Cisco ACI to SR-MPLS hand-off interconnection. SR-MPLS hand-off is an interconnection option that enables Cisco ACI to WAN interconnect using Segment Routing (SR) MPLS underlay.