



Cisco ACI から SR-MPLS ハンドオフにおける DC-PE ルータ

SR-MPLS ハンドオフは、セグメントルーティング (SR) MPLS アンダーレイを使用して Cisco ACI から WAN への相互接続を可能にする相互接続オプションです。

Cisco IOS XE 17.14.1a 以降、Cisco ASR 1000 シリーズ アグリゲーション サービス ルータおよび Cisco Catalyst 8500 シリーズ エッジプラットフォームは、ACI から SR-MPLS ハンドオフ相互接続の中間 DC-PE デバイスとして使用できます。

- [前提条件 \(1 ページ\)](#)
- [機能制限 \(1 ページ\)](#)
- [ACI から SR-MPLS ハンドオフにおける DC-PE ルータに関する情報 \(2 ページ\)](#)
- [サポートされるプラットフォーム \(3 ページ\)](#)
- [DC-PE ルータの設定方法 \(3 ページ\)](#)
- [DC-PE ルータ設定の確認 \(11 ページ\)](#)
- [トラブルシューティングとデバッグ \(14 ページ\)](#)
- [Cisco ACI から SR-MPLS ハンドオフにおける DC-PE ルータの機能情報 \(15 ページ\)](#)

前提条件

ACI から SR-MPLS ハンドオフにおける DC-PE ルータに関する特定の前提条件はありません。

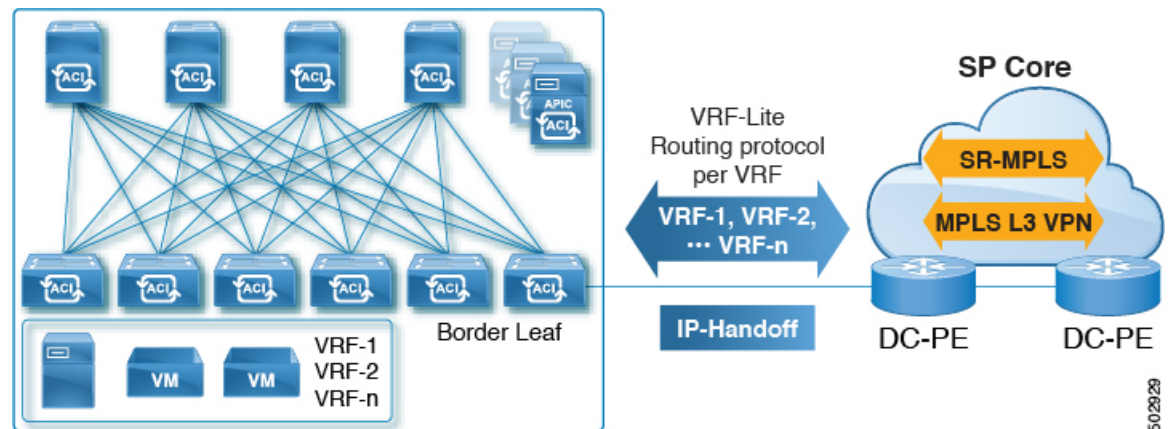
機能制限

- iBGP は、DC-PE とボーダー/リモートリーフ間ではサポートされていません。
- ルータ ID は、すべての境界リーフスイッチと DC-PE で一意である必要があります。

ACI から SR-MPLS ハンドオフにおける DC-PE ルータに関する情報

SR/MPLS ハンドオフは、セグメントルーティング（SR）MPLS アンダーレイを使用して Cisco ACI ファブリックから WAN への相互接続を可能にする相互接続オプションです。SR/MPLS は、SP コアでははるかに一般的であるため、他の既知のソリューションよりも優れたソリューションです。このソリューションには、次の利点があります。

- DC と SP 間の統合トランスポートとポリシー
- 複数の VRF の単一コントロールプレーンセッション
- DC から制御される SP コアのトラフィック エンジニアリング



Cisco ACI ファブリックおよび基盤となる ACI から SR-MPLS ハンドオフ相互接続の詳細については、次の資料を参照してください。

- [ACI SRMPLS ハンドオフホワイトペーパー](#)
- ACI SRMPLS アーキテクチャ：
 1. [Cisco ACI から SR-MPLS ハンドオフの検証済み設計：概要](#)
 2. [Cisco ACI から SR-MPLS ハンドオフの検証済み設計：テナント設定](#)
 3. [ACI ファブリック L3Out ホワイトペーパー](#)
- [ACI SRMPLS アーキテクチャ/サンプルユースケース](#)

サポートされるプラットフォーム

Cisco IOS XE 17.14.1a 以降、次のルータは、ACI から SR-MPLS ハンドオフ相互接続の DC-PE デバイスとして設定できます。

- Cisco ASR 1000 シリーズ アグリゲーション サービス ルータ
- Cisco Catalyst 8500 シリーズ エッジプラットフォーム

DC-PE ルータの設定方法

DC-PE ルータで VRF と BGP を設定するには、次の手順を実行します。

DC-PE ルータでの VRF の設定

手順の概要

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**

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例 : Device> enable	特権 EXEC モードを開始します。パスワードの入力を求められたら、入力します。

	コマンドまたはアクション	目的
ステップ 2	configure terminal 例： Device# configure terminal	グローバル コンフィギュレーション モードを開始します。
ステップ 3	vrf definition vrf-name 例： Device(config)# vrf definition test	指定した VRF インスタンスの VRF コンフィギュレーション モードを開始します。
ステップ 4	rd vpn-route-distinguisher 例： Device(config-vrf)# rd 65000:1	VRF インスタンスのルート識別子を指定します。
ステップ 5	address-family ipv4 [multicast unicast] 例： Device(config-vrf)# address-family ipv4	IPv4 アドレス ファミリ コンフィギュレーション モードを開始します。
ステップ 6	route-target {export import both} route-target-ext-community 例： Device(config-vrf-af)# route-target import 1:1 例： Device(config-vrf-af)# route-target export 2:2	指定した VRF のインポートまたはエクスポート、あるいはインポートとエクスポートの両方のルートターゲットコミュニティのリストを作成します。 自律システム番号および任意の数 (xxx:y) 、または IP アドレスおよび任意の数 (A.B.C.D:y) のいずれかを入力します。
ステップ 7	route-target {export import both} route-target-ext-community stitching 例： Device(config-vrf-af)# route-target import 3:3 stitching 例： Device(config-vrf-af)# route-target export 4:4 stitching	VRF の EVPN ルートターゲットコミュニティのインポートまたはエクスポート、あるいはインポートとエクスポートの両方を設定します。
ステップ 8	exit-address-family 例： Device(config-vrf-af)# exit-address-family	VRF アドレス ファミリ コンフィギュレーション モードを終了し、VRF コンフィギュレーションモードを開始します。
ステップ 9	address-family ipv6 [multicast unicast] 例： Device(config-vrf)# address-family ipv6	IPv6 アドレス ファミリ コンフィギュレーション モードを開始します。
ステップ 10	route-target {export import both} route-target-ext-community 例：	指定した VRF のインポートまたはエクスポート、あるいはインポートとエクスポートの両方のルートターゲットコミュニティのリストを作成します。

	コマンドまたはアクション	目的
	Device(config-vrf-af)# route-target import 1:1 例 : Device(config-vrf-af)# route-target export 2:2	自律システム番号および任意の数 (xxx:y) 、または IP アドレスおよび任意の数 (A.B.C.D:y) のいずれかを入力します。
ステップ 11	route-target {export import both} route-target-ext-community stitching 例 : Device(config-vrf-af)# route-target import 3:3 stitching 例 : Device(config-vrf-af)# route-target export 4:4 stitching	VRF の EVPN ルートターゲットコミュニティのインポートまたはエクスポート、あるいはインポートとエクスポートの両方を設定します。
ステップ 12	exit-address-family 例 : Device(config-vrf-af)# exit-address-family	VRF アドレス ファミリ コンフィギュレーション モードを終了し、VRF コンフィギュレーションモードを開始します。
ステップ 13	end 例 : Device(config-vrf)# end	特権 EXEC モードに戻ります。

例

次に、DC-PE ルータに必要な VRF 設定の例を示します。

```
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
```

DC-PE ルータでの BGP の設定

手順の概要

1. **enable**
2. **configure terminal**

3. **routerbgpas-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 vpv4 unicast [re-originate]**
8. **import vpv6 unicast [re-originate]**
9. **neighbor ip-address activate**
10. **neighbor ip-address send-community [both | extended | standard]**
11. **exit-address-family**
12. **address-family vpv4**
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 vpv6**
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**

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： Device# configure terminal	グローバル コンフィギュレーション モードを開始します。
ステップ 3	routerbgpas-number 例：	BGP ルーティングプロセスを設定し、ルータ コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
	Device(config)# router bgp 1	
ステップ 4	neighbor dc-border-leaf-address remote-as number 例： Device(config-router)# neighbor 1.1.1.1 remote-as 2	EVPN ネットワーク内のマルチプロトコル BGP ネイバーを定義します。 スパインスイッチの IP アドレスをネイバー IP アドレスとして使用します。これにより、スパインスイッチが BGP ネイバーとして設定されます。
ステップ 5	neighbor wan-router-address remote-as number 例： Device(config-router)# neighbor 2.2.2.2 remote-as 1	外部 MPLS ネットワークのマルチプロトコル BGP ネイバーを定義します。 外部 MPLS ネットワークピアの IP アドレスをネイバー IP アドレスとして使用します。これにより、外部 MPLS ネットワークピアが BGP ネイバーとして設定されます。
ステップ 6	address-family l2vpn evpn 例： Device(config-router)# address-family l2vpn evpn	L2VPN アドレスファミリーを指定し、アドレスファミリーコンフィギュレーションモードを開始します。
ステップ 7	import vpnv4 unicast [re-originate] 例： Device(config-router-af)# import vpnv4 unicast re-originate	外部ピアから EVPN アドレスファミリーにインポートされた VPNv4 ルートを EVPN ルートとして再生成し、EVPN ファブリック内で配布します。
ステップ 8	import vpnv6 unicast [re-originate] 例： Device(config-router-af)# import vpnv6 unicast re-originate	外部ピアから EVPN アドレスファミリーにインポートされた VPNv6 ルートを EVPN ルートとして再生成し、EVPN ファブリック内で配布します。
ステップ 9	neighbor ip-address activate 例： Device(config-router-af)# neighbor 1.1.1.1 activate	BGP ネイバーからの情報交換を有効にします。 スパインスイッチの IP アドレスをネイバー IP アドレスとして使用します。
ステップ 10	neighbor ip-address send-community [both extended standard] 例： Device(config-router-af)# neighbor 1.1.1.1 send-community both	BGP ネイバーに送信したコミュニティ属性を指定します。 スパインスイッチの IP アドレスをネイバー IP アドレスとして使用します。 (注) extended キーワードまたは both キーワードを使用します。standard キーワードを使用すると、外部接続を確立できません。

	コマンドまたはアクション	目的
ステップ 11	exit-address-family 例： Device(config-router-af)# exit-address-family	アドレスファミリー コンフィギュレーションモードを終了し、ルータ コンフィギュレーションモードに戻ります。
ステップ 12	address-family vpnv4 例： Device(config-router)# address-family vpnv4	VPNv4 アドレス ファミリーを指定し、アドレスファミリー コンフィギュレーションモードを開始します。
ステップ 13	import l2vpn evpn [re-originate] 例： Device(config-router-af)# import l2vpn evpn re-originate stitching-rt	EVPN ファブリックから VPNv4 アドレスファミリーにインポートされた EVPN ルートを VPNv4 ルートとして再生成し、それらを外部ネットワークに配布します。
ステップ 14	neighbor ip-address activate 例： Device(config-router-af)# neighbor 2.2.2.2 activate	BGP ネイバーからの情報交換を有効にします。 外部 MPLS ネットワークルータの IP アドレスをネイバー IP アドレスとして使用します。
ステップ 15	neighbor ip-address send-community [both extended standard] 例： Device(config-router-af)# neighbor 2.2.2.2 send-community both	BGP ネイバーに送信したコミュニティ属性を指定します。 外部 MPLS ネットワークルータの IP アドレスをネイバー IP アドレスとして使用します。 (注) extended キーワードまたは both キーワードを使用します。standard キーワードを使用すると、外部接続を確立できません。
ステップ 16	neighbor {ip-address peer-group-name} next-hop-self [all] 例： Device(config-router-af)# neighbor 2.2.2.2 next-hop-self all	ルータを BGP スピーキングネイバーまたはピアグループのネクストホップとして設定します。 all キーワードは、EVPN ファブリックと MPLS ネットワークが同じ BGP 自律システム番号にある、iBGP を介した外部接続を実装する場合に必須です。 EVPN ファブリックと MPLS ネットワークが異なる BGP 自律システム番号にある、eBGP を介した外部接続を実装する場合は、all キーワードは任意です。
ステップ 17	exit-address-family 例： Device(config-router-af)# exit-address-family	アドレスファミリー コンフィギュレーションモードを終了し、ルータ コンフィギュレーションモードに戻ります。
ステップ 18	address-family vpnv6 例： Device(config-router)# address-family vpnv6	VPNv6 アドレス ファミリーを指定し、アドレスファミリー コンフィギュレーションモードを開始します。

	コマンドまたはアクション	目的
ステップ 19	import l2vpn evpn [re-originate] 例： Device(config-router-af)# import l2vpn evpn re-originate stitching-rt	EVPN ファブリックから VPNv6 アドレスファミリにインポートされた EVPN ルートを VPNv6 ルートとして再生成し、それらを外部ネットワークに配布します。
ステップ 20	neighbor ip-address activate 例： Device(config-router-af)# neighbor 2.2.2.2 activate	BGP ネイバーからの情報交換を有効にします。 スパインスイッチの IP アドレスをネイバー IP アドレスとして使用します。
ステップ 21	neighbor ip-address send-community [both extended standard] 例： Device(config-router-af)# neighbor 2.2.2.2 send-community both	BGP ネイバーに送信したコミュニティ属性を指定します。 スパインスイッチの IP アドレスをネイバー IP アドレスとして使用します。 (注) extended キーワードまたは both キーワードを使用します。 standard キーワードを使用すると、外部接続を確立できません。
ステップ 22	neighbor {ip-address peer-group-name} next-hop-self [all] 例： Device(config-router-af)# neighbor 2.2.2.2 next-hop-self all	ルータを BGP スピーキングネイバーまたはピアグループのネクストホップとして設定します。 all キーワードは、EVPN ファブリックと MPLS ネットワークが同じ BGP 自律システム番号にある、iBGP を介した外部接続を実装する場合に必須です。 EVPN ファブリックと MPLS ネットワークが異なる BGP 自律システム番号にある、eBGP を介した外部接続を実装する場合は、 all キーワードは任意です。
ステップ 23	exit-address-family 例： Device(config-router-af)# exit-address-family	アドレスファミリ コンフィギュレーションモードを終了し、ルータ コンフィギュレーションモードに戻ります。
ステップ 24	address-family ipv4 vrf vrf-name 例： Device(config-router)# address-family ipv4 vrf test	ルータをアドレスファミリ コンフィギュレーションモードにします。 別々の VRF マルチパス設定は、固有ルート識別子によって分離されます。
ステップ 25	maximum-paths eibgp number 例： Device(config-router-af)# maximum-paths eibgp 16	ルーティングテーブルにインストールできるパラレルの iBGP ルートおよび eBGP ルートの数を設定します。 (注) maximum-paths eibgp コマンドは、IPv4 VRF アドレスファミリ コンフィギュレーションモードでのみ設定できます。

	コマンドまたはアクション	目的
ステップ 26	exit-address-family 例： Device(config-router-af)# exit-address-family	アドレスファミリー コンフィギュレーション モードを終了し、ルータ コンフィギュレーション モードに戻ります。
ステップ 27	address-family ipv6 vrf vrf-name 例： Device(config-router)# address-family ipv6 vrf test	ルータをアドレスファミリー コンフィギュレーション モードにします。 別々の VRF マルチパス設定は、固有ルート識別子によって分離されます。
ステップ 28	maximum-paths eibgp number 例： Device(config-router-af)# maximum-paths eibgp 16	ルーティング テーブルにインストールできるパラレルの iBGP ルートおよび eBGP ルートの数を設定します。 (注) maximum-paths eibgp コマンドは、IPv6 VRF アドレスファミリー コンフィギュレーション モードでのみ設定できます。
ステップ 29	exit-address-family 例： Device(config-vrf-af)# exit-address-family	VRF アドレスファミリー コンフィギュレーション モードを終了し、VRF コンフィギュレーション モードに戻ります。
ステップ 30	end 例： Device(config-vrf)# end	特権 EXEC モードに戻ります。

例

次に、DC-PE ルータに必要な VRF 設定の例を示します。

```
router bgp 1
  neighbor 1.1.1.1 remote-as 2
  neighbor 2.2.2.2 remote-as 1
  address-family l2vpn evpn
    import vpv4 unicast re-originate
    import vpv6 unicast re-originate
    neighbor 1.1.1.1 active
    neighbor 1.1.1.1 send-community both
  exit
  address-family vpv4
    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 vpv6
    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
```

DC-PE ルータ設定の確認

ここでは、DC-PE ルータの設定を確認するために使用できる show コマンドについて説明します。

ACI からの IPv4 および IPv6 ルートの確認

次のコマンドを使用して、ACI からの IPv4 ルートを確認します。

```
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
```

次のコマンドを使用して、ACI からの IPv6 ルートを確認します。

```
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: 0000000000000000000000, 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
```

WAN からの IPv4 および IPv6 ルートの確認

次のコマンドを使用して、WAN からの IPv4 ルートを確認します。

```
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
```

次のコマンドを使用して、WAN からの IPv6 ルートを確認します。

```
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

```

トラブルシューティングとデバッグ

次のデバッグコマンドを使用すると、BGP ラベルマネージャのデバッグに必要なデバッグを有効にできます。

```

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

```

次に、**debug bgp lmm address-family vpnv4/6** コマンドの出力例を示します。

```

*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"

```

次の **debug** コマンドを使用して、BGP EVPN から L3VPN へのインポート/再発信をデバッグできます。

```

debug bgp all import updates
debug bgp all import events

```

次に、**debug bp all import** コマンドの出力例を示します。

```

*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.
```

Cisco ACI から SR-MPLS ハンドオフにおける DC-PE ルータの機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

プラットフォームのサポートおよびシスコソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

表 1: Cisco ACI から SR-MPLS ハンドオフにおける DC-PE ルータの機能情報

機能名	リリース	機能情報
Cisco ACI から SR-MPLS ハンドオフにおける DC-PE ルータ	Cisco IOS XE 17.14.1a	Cisco IOS XE 17.14.1a 以降、Cisco ASR 1000 シリーズ アグリゲーションサービス ルータ および Cisco Catalyst 8500 シリーズ エッジプラットフォームは、Cisco ACI から SR-MPLS ハンドオフ相互接続の中間 DC-PE デバイスとして使用できます。SR-MPLS ハンドオフは、セグメントルーティング (SR) MPLS アンダーレイを使用して Cisco ACI から WAN への相互接続を可能にする相互接続オプションです。

翻訳について

このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。