セグメントルーティングMPLSを使用したレイヤ2 EVPN

レイヤ2 EVPN について

イーサネット VPN（EVPN）は、MPLS ネットワークを介してイーサネット マルチポイント サービスを提供する次世代のソリューションです。EVPN は、コアでコントロールプレーン ベースの MAC ラーニングを可能にする既存の仮想プライベート LAN サービス（VPLS）とは対照的に動作します。EVPN では、EVPN インスタンスに参加している PE が MP-BGP プロトコルを使用してコントロールプレーン内でカスタマー MAC ルートを学習します。コントロールプレーン MAC 学習には数多くの利点があり、フローごとのロードバランシングによるマルチホーミングのサポートなどにより、VPLS の弱点に EVPN で対処できるようにします。

EVPN コントロール プレーンでは、データセンター ネットワークにおいて、次のものを提供します。

- データセンター ネットワークの物理トポロジに制限されない、柔軟なワークロード配置。そのため、データセンター ファブリック内の任意の場所に仮想マシン（VM）を配置できます。
- データセンター内部およびデータセンター間における最適なサーバー間 East-West トラフィック。サーバ/仮想マシン間の East-West トラフィックは、ファースト ホップ ルータでのほぼ特定されたルーティングで達成されます。ファースト ホップ ルーティングはアクセス レイヤで行われます。ホスト ルートの交換は、サーバまたはホストへの流入と送出に関するルーティングがほぼ特定されるようにする必要があります。VM モビリティは、新しい MAC アドレスまたは IP アドレスがローカル スイッチに直接接続されている場合に、新しいエンドポイント接続を検出することでサポートされます。ローカル スイッチは、新しい MAC または IP アドレスを検出すると、ネットワークの残りの部分に新しいロケーションを通知します。
- レイヤ 2 およびレイヤ 3 トラフィックのセグメンテーション。トラフィック セグメンテーションは MPLS カプセル化を使用して実現され、ラベル（BD ごとのラベルおよび VRF ごとのラベル）はセグメント識別子として機能します。

セグメント ルーティング MPLS 上のレイヤ 2 EVPN の注意事項と制限事項

セグメント ルーティング MPLS 上のレイヤ 2 EVPN には、次の注意事項と制限事項があります。

- セグメント ルーティング レイヤ 2 EVPN フラッディングは、入力レプリケーション メカニズムに基づいています。MPLS コアはマルチキャストをサポートしていません。
- ARP 抑制はサポートされていません。
- vPC での整合性チェックはサポートされていません。
- 同じレイヤ 2 EVI とレイヤ 3 EVI を一緒に設定することはできません。
- Cisco NX-OS リリース 9.3(1) 以降、レイヤ 2 EVPN は Cisco Nexus 9300-FX2 プラットフォーム スイッチでサポートされます。
- Cisco NX-OS リリース 9.3(5) 以降、セグメント ルーティング MPLS 上のレイヤ 2 EVPN は、Cisco Nexus 9300-GX および Cisco Nexus 9300-FX3 プラットフォーム スイッチでサポートされます。

セグメント ルーティング MPLS 上のレイヤ 2 EVPNの設定

始める前に

次の手順を実行します。

- **install feature-set mpls** コマンドと **feature-set mpls** コマンドを使用して、MPLS 機能セットをインストールして有効にする必要があります。
- MPLS セグメント ルーティング機能を有効にする必要があります。
- **nv overlay** コマンドを使用して、nv オーバーレイ機能を有効にする必要があります。
- **nv overlay evpn** コマンドを使用して EVPN コントロール プレーンを有効にする必要があります。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		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 2	**feature bgp** 例： `switch(config)#feature bgp`	BGP 機能と構成を有効にします。
ステップ 3	**install feature-set mpls** 例： `switch(config)#install feature-set mpls`	MPLS 構成コマンドを有効にします。
ステップ 4	**feature-set mpls** 例： `switch(config)#install feature-set mpls`	MPLS 構成コマンドを有効にします。
ステップ 5	**feature mpls segment-routing** 例： `switch(config)#feature mpls segment-routing`	セグメント ルーティング構成コマンドを有効にします。

	コマンドまたはアクション	目的
ステップ 6	**feature mpls evpn** 例： `switch(config)#feature mpls evpn`	EVPN over MPLS 構成コマンドを有効にします。このコマンドは **feature-nv** CLI コマンドとは相互に排他的です。
ステップ 7	**feature nv overlay** 例： `switch(config)#feature nv overlay`	セグメント ルーティング レイヤ 2 EVPN に使用される NVE 機能を有効にします。
ステップ 8	**nv overlay evpn** 例： `switch(config)#nv overlay evpn`	EVPN を有効にします。
ステップ 9	**interface loopback** *Interface_Number* 例： `switch(config)#interface loopback 1`	NVE のループバック インターフェイスを設定します。
ステップ 10	**ip address** *address* 例： `switch(config-if)#ip address 192.168.15.1`	IP アドレスを設定します。
ステップ 11	**exit** 例： `switch(config-if)#exit`	グローバルアドレス ファミリ コンフィギュレーション モードを終了します。
ステップ 12	**evpn** 例： `switch(config)#evpn`	EVPN 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	**evi** *number*
	例：
	`switch(config-evpn)#evi 1000`
	`switch(config-evpn-sr)#`
		レイヤ 2 EVIを設定します。必要であれば、自動生成された EVI に基づいて RT を手動で構成できます。

ステップ 14	**encapsulation mpls** 例： `switch(config-evpn)#encapsulation mpls`	MPLSカプセル化と入力レプリケーションを有効にします。
ステップ 15	**source-interface loopback** *Interface_Number* 例： `switch(config-evpn-nve-encap)#source-interface loopback 1`	NVE 送信元インターフェイスを指定します。

	コマンドまたはアクション	目的
ステップ 16	**exit** 例： `switch(config-evpn-nve-encap)#exit`	設定を終了します。
ステップ 17	**vrf context** *VRF_NAME* 例： `switch(config)#vrf context Tenant-A`	VRF を設定します。
ステップ 18	**evi** *EVI_ID* 例： `switch(config-vrf)#evi 30001`	L3 EVI を設定します。
ステップ 19	**exit** 例： `switch(config-vrf)#exit`	設定を終了します。
ステップ 20	**VLAN** *VLAN_ID* 例： `switch(config)#vlan 1001`	VLAN を設定します。
ステップ 21	**evi auto** 例： `switch(config-vlan)#evi auto`	L2 EVI を設定します。
ステップ 22	**exit** 例： `switch(config-vlan)#exit`	
ステップ 23	**router bgp** *autonomous-system-number* 例： `switch(config)#router bgp 1`	BGP コンフィギュレーション モードを開始します。
ステップ 24	**address-family l2vpn evpn** 例： `switch(config-router)#address-family l2vpn evpn`	EVPN アドレス ファミリをグローバルに有効にします。
ステップ 25	**neighbor address** *remote-as autonomous-system-number* 例： `switch(config-router)#neighbor 192.169.13.1 remote as 2`	BGP ネイバーを設定します。

	コマンドまたはアクション	目的
ステップ 26	**address-family l2vpn evpn** 例： `switch(config-router-neighbor)#address-family l2vpn evpn`	ネイバーの EVPN アドレス ファミリを有効にします。
ステップ 27	**encapsulation mpls** 例： `switch(config-router-neighbor)#encapsulation mpls`	MPLS カプセル化を有効にします。
ステップ 28	**send-community extended** 例： `switch(config-router-neighbor)#send-community extended`	BGP を設定し、拡張コミュニティ リストをアドバタイズします。
ステップ 29	**vrf** *VRF_NAME* 例： `switch(config-router)#vrf Tenant-A`	BGP VRF を設定します。
ステップ 30	**exit** 例： `switch(config-router)#exit`	設定を終了します。

EVI 用の VLAN の設定

手順

	コマンドまたはアクション	目的
ステップ 1	**vlan** *number*	VLAN を設定します。
ステップ 2	**evi** *[auto]*	VLAN の BD ラベルを作成します。このラベルは、セグメント ルーティング レイヤ 2 EVPN 全体で VLAN の識別子として使用されます。

NVE インターフェイスの設定

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		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 2	**interface loopback** *loopback_number* 例： `switch(config)# interface loopback 1`	IPアドレスをこのループバック インターフェイスに関連付け、この IP アドレスをセグメント ルーティング設定に使用します。
ステップ 3	**ip address** 例： `switch(config-if)#ip address 192.169.15.1/32`	IPv4 アドレス ファミリを指定し、ルータ 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 4	**evpn** 例： `switch(config)#evpn`	EVPN 設定モードを開始します。
ステップ 5	**encapsulation mpls** 例： `switch(config-evpn)# encapsulation mpls`	MPLS カプセル化と入力レプリケーションを有効にします。
ステップ 6	**source-interface** *loopback_number* 例： `switch(config-evpn-nve-encap)#source-interface loopback 1`	NVE 送信元インターフェイスを指定します。
ステップ 7	**exit** 例： `switch(config)# exit`	セグメント ルーティング モードを終了し、コンフィギュレーション端末モードに戻ります。

VRF 下での EVI の設定

手順

	コマンドまたはアクション	目的
ステップ 1	**vrf context** テナント	VRF テナントを作成します。
ステップ 2	**evi** *number*	VRF 下でレイヤ 3 EVI を設定します。

エニーキャスト ゲートウェイの設定

ファブリック転送の設定は、SVI がエニーキャスト モードで設定されている場合にのみ必要です。

手順

	コマンドまたはアクション	目的
ステップ 1	**fabric forwarding anycast-gateway-mac 0000.aabb.ccdd**	分散ゲートウェイの仮想 MAC アドレスを設定します。
ステップ 2	**fabric forwarding mode anycast-gateway**	インターフェイス コンフィギュレーション モードで SVI をエニーキャスト ゲートウェイと関連付けます。

ループバック インターフェイスのラベル付きパスのアドバタイズ

レイヤ 2 EVPN エンドポイントとしてアドバタイズされるループバック インターフェイスは、ラベル インデックスにマッピングする必要があります。これにより、BGP は、同じものに対応する MPLS ラベル付きパスをアドバタイズします。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		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

	コマンドまたはアクション	目的
ステップ 2	[**no**]**router ospf** *process* 例： `switch(config)# router ospf test`	OSPF モードを有効にします。
ステップ 3	**segment-routing** 例： `switch(config-router)# segment-routing mpls`	OSPF でのセグメント ルーティング機能を設定します。
ステップ 4	**connected-prefix-sid-map** 例： `switch(config-sr-mpls)# connected-prefix-sid-map`	ローカル プレフィックスと SID のアドレス ファミリ固有のマッピングを設定できるサブモードを開始します。
ステップ 5	**address-family ipv4** 例： `switch(config-sr-mpls-conn)# address-family ipv4`	IPv4 アドレス プレフィックスを指定します。
ステップ 6	**1.1.1.1/32 index 100** 例： `switch(config-sr-mpls-conn-af)# 1.1.1.1/32 100`	SID 100 にアドレス 1.1.1.1/32 を関連付けます。
ステップ 7	**exit-address-family** 例： `switch(config-sr-mpls-conn-af)# exit-address-family`	アドレス ファミリを終了します。

SRv6 静的プレフィックス単位 TE について

SRv6 静的プレフィックス単位 TE 機能を使用すると、デフォルト以外の VRF にマッピングされたプレフィックスをマッピングおよびアドバタイズできます。この機能により、一致する VRF ルート ターゲットを使用して単一のインスタンスで複数のプレフィックスをアドバタイズでき、各プレフィックスを手動で入力する必要がなくなります。

Cisco NX-OS リリース 9.3(5) では、1 つの VNF だけが VM にサービスを提供できます。

SRv6 の静的なプレフィックスごとの TE の設定

始める前に

次の手順を実行します。

- **install feature-set mpls** コマンドと **feature-set mpls** コマンドを使用して、MPLS 機能セットをインストールして有効にする必要があります。
- MPLS セグメント ルーティング機能を有効にする必要があります。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		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 2	**vrf context** *VRF_Name* 例： `switch(config)# vrf context vrf_2_7_8`	VRF を定義し、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 3	**rd** *rd_format* 例： `switch(config-vrf)# rd 2.2.2.0:2`	RD を VRF に割り当てます。
ステップ 4	**address-family** {**ipv4** | **ipv6** } 例： `switch(config-vrf)# address-family ipv4 unicast`	VRF インスタンス用に IPv4 または IPv6 アドレスファミリを指定し、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 5	**route-target import** *route-target-id* 例： `switch(config-vrf)# route-target import 1:2`	VRF へのルートのインポートを設定します。
ステップ 6	**route-target import** *route-target-id* **evpn** 例： `switch(config-vrf)# route-target import 1:2 evpn`	一致するルート ターゲット値を持つ、レイヤ3 EVPN から VRF へのルートのインポートを設定します。
ステップ 7	**route-target export** *route-target-id* 例：	VRF からのルートのエクスポートを設定します。

	コマンドまたはアクション	目的
	`switch(config-vrf)# route-target export 1:2`	
ステップ 8	**route-target export** *route-target-id* **evpn** 例： `switch(config-vrf)# route-target export 1:2 evpn`	一致するルートターゲット値を持つ、VPN から レイヤ3 EVPN からへのルートのエクスポートを設定します。
ステップ 9	**router bgp** *autonomous-system-number* 例： `switch(config)# router bgp 65000`	BGP を有効にして、ローカル BGP スピーカに AS 番号を割り当てます。
ステップ 10	**router-id** *id* 例： `switch(config-router)# router-id 2.2.2.0`	ルータ ID を設定します。
ステップ 11	**address-family l2vpn evpn** 例： `switch(config-router-af)# address-family l2vpn evpn`	レイヤ 2 VPN EVPN のグローバル 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 12	**neighbor** *ipv4-address* **remote-as**
	例：
	`switch(config-router)# neighbor 7.7.7.0 remote-as 65000`
	`switch(config-router-neighbor)#`
		リモート BGP ピアの IPv4 アドレスおよび AS 番号を設定します。

ステップ 13	**update-source loopback** *number* 例： `switch(config-router-neighbor)# update-source loopback0`	ループバック番号を指定します
ステップ 14	**address-family l2vpn evpn** 例： `switch(config-router-neighbor)#address-family l2vpn evpn`	ネイバーの EVPN アドレス ファミリを有効にします。
ステップ 15	**send-community extended** 例： `switch(config-router-neighbor)#send-community extended`	BGP を設定し、拡張コミュニティ リストをアドバタイズします。
ステップ 16	**encapsulation mpls** 例： `switch(config-router-neighbor)#encapsulation mpls`	MPLS カプセル化を有効にします。

	コマンドまたはアクション	目的
ステップ 17	**exit** 例： `switch(config-router-neighbor)#exit`	設定を終了します。

例

次の例は、VRF VT を定義するために RPM 構成を設定する方法を示しています。

```
rf context vrf_2_7_8
          rd 2.2.2.0:2
          address-family ipv4 unicast
            route-target import 0.0.1.1:2
            route-target import 0.0.1.1:2 evpn
            route-target export 0.0.1.1:2
            route-target export 0.0.1.1:2 evpn
ip extcommunity-list standard vrf_2_7_8-test permit rt 0.0.1.1:2
            route-map Node-2 permit 4
            match extcommunity vrf_2_7_8-test
            set extcommunity color 204
```

Route-Target Auto について

自動派生Route-Target（route-target import/export/both auto）は、IETF RFC 4364 セクション 4.2（https://tools.ietf.org/html/rfc4364#section-4.2）で説明されているタイプ 0 エンコーディング形式に基づいています。IETF RFC 4364 セクション 4.2 ではルート識別子形式について説明し、IETF RFC 4364 セクション 4.3.1では、Route-Target に同様の形式を使用することが望ましいとしています。タイプ 0 エンコーディングでは、2 バイトの管理フィールドと 4 バイトの番号フィールドを使用できます。Cisco NX-OS 内では、自動派生 Route-Target は、2 バイトの管理フィールドとして自律システム番号（ASN）、4 バイトの番号フィールドのサービス識別子（EVI）で構成されます。

2 バイト ASN

タイプ 0 エンコーディングでは、2 バイトの管理フィールドと 4 バイトの番号フィールドを使用できます。Cisco NX-OS 内では、自動派生 Route-Target は、2 バイトの管理フィールドとしての自律システム番号（ASN）と、4 バイトの番号フィールドのサービス識別子（EVI）で構成されます。

自動派生 Route-Target（RT）の例：

- ASN 65001 と L3EVI 50001 内の IP-VRF： Route-Target 65001:50001
- ASN 65001 と L2VNI 30001 内の MAC-VRF：Route-Target 65001:30001

Multi-AS 環境では、Route-Target を静的に定義するか、Route-Target の ASN 部分と一致するように書き換える必要があります。

（注） 4 バイト ASN の自動派生 Route-Target はサポートされていません。

4 バイト ASN

タイプ 0 エンコーディングでは、2 バイトの管理フィールドと 4 バイトの番号フィールドを使用できます。Cisco NX-OS 内では、自動派生 Route-Target は、2 バイトの管理フィールドとしての自律システム番号（ASN）と、4 バイトの番号フィールドのサービス識別子（EVI）で構成されます。4 バイト長の ASN 要求と 24 ビット（3 バイト）を必要とする EVI では、拡張コミュニティ内のサブフィールド長が使い果たされます（2 バイト タイプと 6 バイト サブフィールド）。長さと形式の制約、およびサービス識別子（EVI）の一意性の重要性の結果、4 バイトの ASN は、IETF RFC 6793 セクション 9（https://tools.ietf.org/html/rfc6793#section-9）で説明されているように、AS_TRANS という名前の 2 バイトの ASN で表されます。2 バイトの ASN 23456 は、4 バイトの ASN をエイリアスする特別な目的の AS 番号であるAS_TRANSとして IANA（https://www.iana.org/assignments/iana-as-numbers-special-registry/iana-as-numbers-special-registry.xhtml）によって登録されます。

4 バイトの ASN（AS_TRANS）を使用した自動派生 Route-Target（RT）の例：

- ASN 65656 と L3VNI 50001 内の IP-VR：Route-Target 23456:50001
- ASN 65656 とL2VNI 30001 内の MAC-VRF：Route-Target 23456:30001

BD 用の RD およびルート ターゲットの設定

VLAN で evi auto を設定すると、ブリッジ ドメイン (BD) RD およびルート ターゲットが自動的に生成されます。BD RD およびルート ターゲットを手動で設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		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 2	**evpn** 例： `switch(config)# evpn`	EVPN 設定モードを開始します。
ステップ 3	**evi** *VLAN_ID* 例： `switch(config-evpn)# evi 1001`	RD/ルート ターゲットを設定するための L2 EVI を指定します。

	コマンドまたはアクション	目的
ステップ 4	**rd** *rd_format* 例： `switch(config-evpn-evi-sr)# rd 192.1.1.1:33768`	RD を設定します。
ステップ 5	**route-target both** *rt_format* 例： `switch(config-evpn-evi-sr)# route-target both 1:20001`	ルートターゲットを設定します。

VRF用 の RD およびルート ターゲットの設定

VRF で **evi** *evi_ID* を設定すると、VRF RD およびルート ターゲットが自動的に生成されます。VRF RD およびルート ターゲットを手動で設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `switch# configure terminal switch(config)#`	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 2	**vrf context** *VRF_NAME* 例： `switch(config)# vrf context A`	VRF を設定します。
ステップ 3	**rd auto** または **rd_format** 例： `switch(config-vrf)# rd auto`	RD を設定します。
ステップ 4	**address-family ipv4 unicast** 例： `switch(config-vrf)# address-family ipv4 unicast`	IPv4アドレスファミリを有効にします。
ステップ 5	**route-target both** *rt_format* **evpn** 例： `switch(config-vrf-af-ipv4)# route-target both 1:30001 evpn`	ルートターゲットを設定します。

セグメントルーティング MPLS 上のレイヤ 2 EVPN の設定例

次の例は、セグメント ルーティング MPLS を介したレイヤ 2 EVPN の設定を示しています。

```
install feature-set mpls
feature-set mpls
nv overlay evpn
feature bgp
feature mpls segment-routing
feature mpls evpn
feature interface-vlan
feature nv overlay

fabric forwarding anycast-gateway-mac 0000.1111.2222

vlan 1001
  evi auto

vrf context Tenant-A
  evi 30001

interface loopback 1
  ip address 192.168.15.1/32

interface vlan 1001
  no shutdown
  vrf member Tenant-A
  ip address 111.1.0.1/16
  fabric forwarding mode anycast-gateway

router bgp 1
  address-family l2vpn evpn
    neighbor 192.169.13.1
      remote-as 2
      address-family l2vpn evpn
        send-community extended
        encapsulation mpls
    vrf Tenant-A

evpn
  encapsulation mpls
    source-interface loopback 1
```

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。