



## Cisco IOS XE Everest 16.6.x (Catalyst 9500 スイッチ) キャンパス ファブリック コンフィギュレーションガイド

初版 : 2017 年 7 月 31 日

### シスコシステムズ合同会社

〒107-6227 東京都港区赤坂9-7-1 ミッドタウン・タワー

<http://www.cisco.com/jp>

お問い合わせ先 : シスコ コンタクトセンター

0120-092-255 (フリーコール、携帯・PHS含む)

電話受付時間 : 平日 10:00~12:00、13:00~17:00

<http://www.cisco.com/jp/go/contactcenter/>





## 目次

### 第 1 章

#### キャンパス ファブリック 1

キャンパス ファブリックについて 1

キャンパス ファブリック ネットワークのプロビジョニングの利点 2

ファブリック ドメイン要素について 2

キャンパス ファブリックの設定の注意事項と制限 3

キャンパス ファブリック：規模とパフォーマンス 4

Cisco IOS XE Everest 16.6.1 以降の CLI の変更点 4

キャンパス ファブリックの設定方法 5

ファブリック エッジデバイスの設定 5

エニーキャストスイッチ仮想インターフェイス (SVI) としてのファブリックエッジノードの設定 8

DHCP リレー エージェントとしてのファブリック エッジ ノードの設定 9

ファブリック ボーダー デバイスの設定 10

ファブリック コントロールプレーンの設定 11

LISP 設定をトラブルシューティングするための show コマンド 12

ファブリック エッジ ノードでの LISP 設定の設定例 12

キャンパス ファブリックのデータ プレーンセキュリティ 15

エッジデバイスでのデータ プレーンセキュリティの設定 16

コントロールプレーンデバイスでのデータ プレーンセキュリティの設定 17

ファブリック ボーダー デバイスの設定 18

キャンパス ファブリックでのセキュリティ グループ タグとポリシーの適用 19

キャンパス ファブリック オーバーレイを使用したマルチキャスト 19

LISP マルチキャストについて 20

IPv4 レイヤ 3 LISP マルチキャストの設定 20

レイヤ 2 オーバーレイ ブロードキャストの設定	22
LISP マルチキャスト設定をトラブルシューティングするための show コマンド	23
LISP マルチキャストの設定例	23
キャンパス ファブリックの機能履歴	25

---

**第 2 章**

<b>キャンパス ファブリック内での DHCP の設定</b>	<b>27</b>
キャンパス ファブリックの DHCP 設定	27
DHCP パケットフロー	28
キャンパス ファブリック ネットワーク内の DHCP クライアントへの IP アドレス割り当てに おける操作のシーケンス	29
DHCP クライアント/サーバの設定方法	30
DHCP リレー エージェントとしてのファブリック エッジノードの設定	30
エニーキャスト SVI としてのファブリック エッジノードの設定	31
ファブリック エッジノードでの LISP の設定	32
DHCP 設定例	33
キャンパス ファブリックでの DHCP ソリューションの機能履歴	36



# 第 1 章

## キャンパス ファブリック

- [キャンパス ファブリックについて \(1 ページ\)](#)
- [キャンパス ファブリックの設定方法 \(5 ページ\)](#)
- [LISP 設定をトラブルシューティングするための show コマンド \(12 ページ\)](#)
- [ファブリック エッジ ノードでの LISP 設定の設定例 \(12 ページ\)](#)
- [キャンパス ファブリックのデータプレーンセキュリティ \(15 ページ\)](#)
- [キャンパスファブリックでのセキュリティグループタグとポリシーの適用 \(19 ページ\)](#)
- [キャンパス ファブリック オーバーレイを使用したマルチキャスト \(19 ページ\)](#)
- [キャンパス ファブリックの機能履歴 \(25 ページ\)](#)

## キャンパス ファブリックについて

キャンパスファブリックはソフトウェア定義型アクセスとも呼ばれ、ポリシーベースのセグメンテーション構造上に仮想ネットワークを構築するための基本インフラストラクチャを提供します。任意のアンダーレイネットワークの上に構築された Locator ID Separator Protocol (LISP) オーバーレイ ネットワークに基づいています。

オーバーレイ ネットワークはすべてのアンダーレイ ネットワーク デバイスまたはそれらのデバイス サブネットにわたって実行できます。複数のオーバーレイ ネットワークがマルチテナント機能をサポートするように、同じアンダーレイ ネットワーク全体に拡散させることができます。

Cisco IOS XE Everest 16.6.1 は、レイヤ 2 およびレイヤ 3 のオーバーレイ ネットワークをサポートしています。

キャンパス ファブリック オーバーレイのプロビジョニングでは次の 3 つのコンポーネントを使用し、ユーザやデバイスのフレキシブルな接続や、ユーザベースおよびデバイスグループベースのポリシーを通じたセキュリティの強化を可能にします。

- コントロールプレーン
- データプレーン
- ポリシープレーン

キャンパスファブリック機能は、エンタープライズサービスとIPベースソフトウェアイメージでサポートされています。

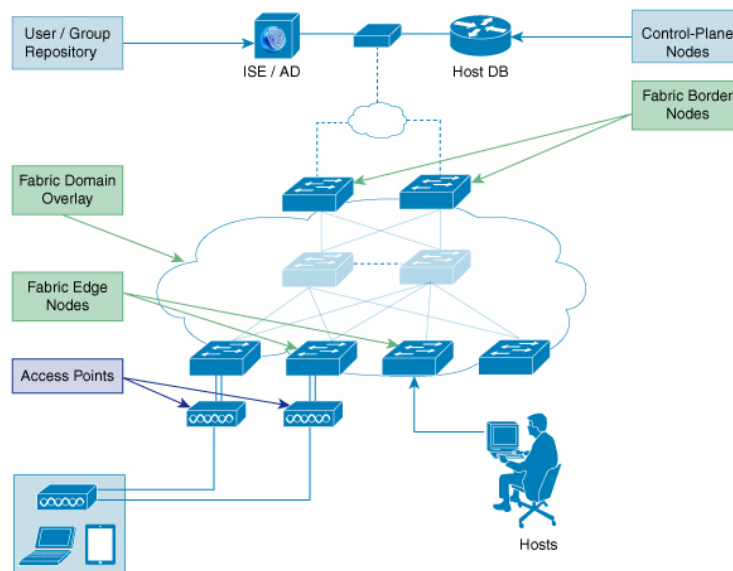
## キャンパス ファブリック ネットワークのプロビジョニングの利点

- ハイブリッドレイヤ2とレイヤ3オーバーレイは、それらの両方のサービスの長所を提供します。
- LISP 仮想化テクノロジーを使用したエンドツーエンドのセグメンテーションを提供します。この場合、ファブリック エッジ ノードとボーダー ノードのみが LISP 認識型になっている必要があります。残りのコンポーネントは IP フォワーダにすぎません。
- スパニングツリープロトコル (STP) を排除し、リンク使用率を向上させ、高速コンバージェンスと等コスト マルチパス (ECMP) のロード バランシングをもたらします。
- ファブリック ヘッダーはセキュリティグループタグ (SGT) の伝達をサポートして、ネットワーク全体で統一されたポリシー モデルが確立されるようにします。SGT ベースのポリシー構造はサブネットに依存しません。
- 有線とワイヤレスの両方のクライアントにホスト モビリティを提供します。
- LISP を使用すると、ホスト アドレスとその場所を分離し、ルーティング操作を簡略化して、スケーラビリティとサポートを向上させるのに役立ちます。

## ファブリック ドメイン要素について

図 1: ファブリック ドメインの要素 に、ファブリック ドメインを構成する要素を示します。

図 1: ファブリック ドメインの要素



次に、[図 1: ファブリック ドメインの要素](#)に示したファブリック ドメインの要素について説明します。

- **ファブリック エッジデバイス**：ファブリック ドメインに接続するユーザとデバイスに接続を提供します。ファブリック エッジデバイスは、エンドポイントを識別して認証し、エンドポイント ID 情報をファブリック ホストトラッキングデータベースに登録します。また、これらのデバイスは入力時にカプセル化を、出力時にはカプセル化解除を実行し、ファブリック ドメインに接続されたエンドポイント間でトラフィックを転送します。
- **ファブリック コントロールプレーンのデバイス**：オーバーレイ到達可能性情報とエンドポイントからルーティング ロケータへのマッピングをホストトラッキングデータベースで提供します。コントロールプレーンのデバイスは、ローカルエンドポイントを持つファブリック エッジデバイスから登録を受信し、エッジデバイスからのリモートエンドポイントを検索する要求を解決します。ネットワークに冗長性をもたせるために、内部（ファブリック ボーダー デバイス）および外部（Cisco CSR1000v などの指定されたコントロールプレーン デバイス）に最大 3 台のコントロールプレーン デバイスを設定できます。
- **ファブリック ボーダー デバイス**：従来のレイヤ 3 ネットワークまたは異なるファブリック ドメインをローカル ドメインに接続し、Virtual Routing and Forwarding (VRF) や SGT の情報などの到達可能性情報とポリシー情報を 1 つのドメインから別のドメインに変換します。
- **仮想コンテキスト**：レイヤ 3 ルーティングテーブルの複数のインスタンスを作成するために、デバイス レベルで仮想化を提供します。コンテキストまたは VRF は、IP アドレス全体のセグメンテーションを行い、オーバーラップしたアドレス空間とトラフィックの分離を可能にします。ファブリック ドメインに最大 32 のコンテキストを設定できます。
- **ホストプール**：ファブリック ドメイン内にあるエンドポイントを IP プールにグループ化し、VLAN ID および IP サブネットでそれらを識別します。

## キャンパス ファブリックの設定の注意事項と制限

- 各ファブリック ドメインに設定するコントロールプレーン デバイスは 3 台までです。
- 各ファブリック ドメインに設定するボーダー デバイスは 2 台までです。
- 各ファブリック エッジ デバイスは、最大 2000 のホストをサポートします。
- 各コントロールプレーン デバイスは、最大 5000 のファブリック エッジ デバイス登録をサポートします。
- 各ファブリック ドメインに設定する仮想コンテキストは 64 個までです。
- レイヤ 2 (IPv4 ホスト) とレイヤ 3 (IPv6 ホスト) の LISP オーバーレイ機能は、Cisco IOS XE Everest 16.6.1 以降のリリースでサポートされています。
- エッジデバイス上で、Cisco TrustSec のリンクは、アンダーレイに接続されたアップリンク インターフェイスではサポートされません。

- レイヤ3送信元グループのタグは、アンダーレイに接続されたアップリンクインターフェイスには適用できません。
- Cisco IOS XE 16.6.1 はデンス モードまたは Bidirectional Protocol Independent Multicast (PIM) をサポートしていません。PIM スパース モード (SM) および PIM Source Specific Multicast (SSM) モードのみをサポートしています。
- マルチキャストは Group-to-Rendezvous Point (RP) マッピングの分散機能、オート RP、ブートストラップルータ (BSR) はサポートしていません。スタティック RP 設定のみをサポートしています。
- マルチキャスト RP の冗長性は、ファブリック ドメインではサポートされていません。



**重要** 仮想拡張 LAN (VXLAN) と LISP はキャンパス ファブリック ネットワークの一部として設定する必要があります。スタンドアロン機能としてはサポートされていません。

## キャンパス ファブリック：規模とパフォーマンス

- レイヤ2 EID VLAN の有効な最大数は 2048 です。
- 各ファブリック エッジのローカル ホストとリモート ホストの最大数は 32000 です。
- ファブリックに接続できるアクセス ポイントの最大数は 100 です。
- キャンパス ファブリックでオンボード可能なワイヤレス クライアントの最大数は 2000 です。

## Cisco IOS XE Everest 16.6.1 以降の CLI の変更点

Cisco IOS XE Everest 16.6.1 以降、L2 LISP 設定の CLI モデルは、設定フローをよりの確に反映させ、また、EID プレフィックスなどのレイヤ2 MAC アドレスのサポートなど、さまざまな機能に固有の LISP 動作を設定するように再設計されています。

次に、CLI の変更点のリストを示します。

- 新しい CLI では、2つのパスにおける2レベルの継承を行えます。
  - **router lisp > service** : グローバルサービス モードまたは上位サービス モードといいます。
  - **router lisp > instance-id > service** : インスタンスサービス モードといいます。
- エンドポイント識別子テーブル (**eid-table**) は **instance-id** から分離されています。このため、**instance-id** を指定せずに **eid-table** を設定できるようになりました。階層は **router lisp > instance-id > service > eid-table** になります。



- グローバル サービス モードでは共通設定、インスタンスサービス モードではインスタンス ID 固有の設定を行うことができます。
- 階層のグローバル レベルに設定した CLI は、明示的に上書きしない限り、下位レベルのすべてのインスタンス サービスの動作状態に影響します。
- すべての { ipv4 | ipv6 } ipv4 [proxy] { itr | etr } コマンドが、それらのアドレス ファミリ プレフィックスなしに、それぞれのサービス モードの下に表示されます。
- すべての LISP が **show lisp** プレフィックスでコマンドの開始を示します。
- グローバル レベルで設定された新しいコマンドの **locator default-set** はデフォルトとして設定したロケータ セットの 1 つをマークします。
- **service-ethernet** は、レイヤ 2 MAC ID を EID スペースとしてイネーブルにする新しいサブモードです。



(注) 変更後の設定形式でコマンドを入力すると、以前の CLI はサポートされません。以前の CLI に切り替えるには、システムをリロードします。

## キャンパス ファブリックの設定方法

キャンパス ファブリックの設定には次の段階があります。

- ネットワーク プロビジョニング：管理プレーンとアンダーレイ メカニズムのセットアップ
- オーバーレイ プロビジョニング：ファブリック エッジとファブリック ボーダーのデバイスを含めたファブリック オーバーレイのセットアップ
- ポリシー管理：仮想コンテキストまたは VRF、エンドポイント グループおよびポリシーのセットアップ

## ファブリック エッジ デバイスの設定

ファブリック エッジ デバイスを設定するには、次の手順を実行します。

### 始める前に

- デバイスが確実に到達できるように、各エッジデバイスに **loopback0** IP アドレスを設定します。 **ip lisp source-locator loopback0** コマンドをアップリンク インターフェイスで実行したことを確認します。
- アンダーレイ設定が設定されていることを確認します。

- ファブリック ドメイン内でコントロールプレーン デバイスとボーダー デバイスを設定します。

手順

	コマンドまたはアクション	目的
ステップ 1	<b>configureterminal</b> 例： Switch# 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 2	<b>router lisp</b> 例： Switch(config)# router lisp	LISP 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	<b>locator-table name {default   vrf vrf-name}</b> 例： Switch(config-router-lisp)# locator-table loc-table default	ルータがロケータアドレス空間に到達できるようにする仮想ルーティングおよび転送 (VRF) テーブルを関連付けます。
ステップ 4	<b>locator-set name {ip-address {priority priority_value   weight weight}   auto-discover-rlocs}</b> 例： Switch(config-router-lisp-locator-table)# locator-set rloc1 1.1.1.1 priority 1 weight 1	名前付きロケータ セットを指定します。
ステップ 5	<b>IPv4-interface loopback Loopback-address { priority priority_value   weight weight}</b> 例： Switch(config-router-lisp-locator-set)# IPv4-interface loopback0 priority 1 weight 1	ループバック IP アドレスを設定してデバイスが到達可能であることを確認します。
ステップ 6	<b>exit-locator-set</b> 例： Switch(config-router-lisp-locator-set)# exit-locator-set	ロケータ設定コンフィギュレーション モードを終了します。
ステップ 7	<b>instance-id instance</b> 例： Switch(config-router-lisp)# instance-id 3	複数のサービスをグループ化する LISP EID インスタンスを作成します。このインスタンス ID での設定は、下位のすべてのサービスに適用できます。

	コマンドまたはアクション	目的
ステップ 8	<b>dynamic-eid</b> <i>dynamic-EID</i> 例： Switch(config-router-lisp-instance)# dynamic-eid DEFAULT.EID.eng	ダイナミック EID ポリシーを作成し、ダイナミック EID 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	<b>database-mapping</b> <i>eid locator-set RLOC name</i> 例： Switch(config-router-lisp-instance-dynamic-eid)# database-mapping 10.1.1.0/24 locator-set set1	EID を RLOC マッピング関係に設定します。
ステップ 10	<b>exit-dynamic-eid</b> 例： Switch(config-router-lisp-instance-dynamic-eid)# exit-dynamic-eid	ダイナミック EID コンフィギュレーション モードを終了します。
ステップ 11	<b>service ipv4</b> 例： Switch(config-router-lisp-instance)# service ipv4	IPv4 アドレスファミリーに対してレイヤ 3 ネットワーク サービスをイネーブルにし、サービスサブモードを開始します。
ステップ 12	<b>eid-table vrf</b> <i>vrf-table</i> 例： Switch(config-router-lisp-instance-service)# eid-table vrf vrf2	エンドポイント識別子アドレス空間に到達できるようにする VRF テーブルと以前に設定した LISP インスタンス ID を関連付けます。
ステップ 13	<b>map-cache destination-eidmap-request</b> 例： Switch(config-router-lisp-instance-service)# map-cache 10.1.1.0/24 map-request	宛先 EID のスタティック マップ要求を生成します。
ステップ 14	<b>itr map-resolver</b> <i>map-resolver-address</i> 例： Switch(config-router-lisp-instance-service)# itr map-resolver 2.1.1.6	宛先 EID IP に対応する RLOC を照会する必要があるマップリゾルバ IP を設定します。
ステップ 15	<b>itr</b> 例： Switch(config-router-lisp-instance-service)# itr	このデバイスが入力トンネル ルータ (ITR) として動作することを指定します。
ステップ 16	<b>etr map-server</b> <i>map-server-addrkey {0   6} authentication key</i> 例：	IPv4 EID の登録時に出力トンネル ルータ (ETR) が使用する LISP マップサーバのロケータアドレスを設定します。

	コマンドまたはアクション	目的
	Switch(config-router-lisp-instance-service)# etr map-server 2.1.1.6 key foo	
ステップ 17	<b>etr</b> 例 : Switch(config-router-lisp-instance-service)# etr	このデバイスが ETR として動作することを指定します。
ステップ 18	<b>use-petr locator-address { priority priority_value   weight weight_value</b> 例 : Switch(config-router-lisp-instance-service)# use-petr 14.1.1.1	プロキシ出力トンネルルータ (PETR) を使用するようにデバイスを設定します。
ステップ 19	<b>exit-service-ipv4</b> 例 : Switch(config-router-lisp-instance-service)# exit-service-ipv4	サービス サブモードを終了します。
ステップ 20	<b>exit-instance-id</b> 例 : Switch(config-router-lisp-instance)# exit-instance-id	インスタンス サブモードを終了します。

## エニーキャストスイッチ仮想インターフェイス (SVI) としてのファブリック エッジノードの設定

エニーキャスト SVI としてファブリック エッジノードを設定するには、次の手順に従います。

### 手順

	コマンドまたはアクション	目的
ステップ 1	<b>configureterminal</b> 例 : switch# 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 2	<b>interface interface</b> 例 : switch(config)# interface vlan10	SVI 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	<b>ip vrf forwarding vrf-name</b> 例 : Switch(config-if)# ip vrf forwarding EMP	インターフェイス上で VRF を設定します。

	コマンドまたはアクション	目的
ステップ 4	<b>ip address</b> <i>ipv4-address</i> 例： Switch(config-if)# ip address 192.168.10.1/24	インターフェイスの IP アドレスを設定します。
ステップ 5	<b>ip helper-address</b> <i>ipaddress</i> 例： Switch(config-if)# ip helper-address 172.168.1.1	DHCP ブロードキャストは、ルータによってドロップされるのではなく、この特定のヘルパアドレスにユニキャストとして転送されます。
ステップ 6	<b>lisp mobility</b> 例： Switch(config-if)# lisp mobility	LISP 仮想マシンのモビリティであるダイナミック EID ローミングに参加するようにインターフェイスを設定します。

## DHCP リレー エージェントとしてのファブリック エッジノードの設定

次に、DHCP リレー エージェントとしてファブリック エッジを設定する手順について説明します。キャンパス ファブリックでの DHCP クライアント/サーバの設定の詳細については、『Cisco IOS XE 16.6.1 Configure DHCP for Campus Fabric』のドキュメントを参照してください。

### 手順

	コマンドまたはアクション	目的
ステップ 1	<b>configureterminal</b> 例： Switch# 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 2	<b>ip dhcp snooping</b> 例： Switch(config)# ip dhcp snooping	DHCP スヌーピングをグローバルにイネーブル化します。
ステップ 3	<b>ip dhcp snooping vlan</b> 例： Switch(config-if)# ip dhcp snooping vlan	指定した VLAN で DHCP スヌーピングをイネーブルにします。
ステップ 4	<b>ip dhcp relay information option</b> 例： Switch(config-if)# ip dhcp relay information option	DHCP サーバへ転送したメッセージに、システムが DHCP リレー エージェント情報オプション (Option 82 フィールド) を挿入できるようにします。

## ファブリック ボーダー デバイスの設定

ファブリック ボーダー デバイスを設定するには、次の手順に従います。

### 手順

	コマンドまたはアクション	目的
ステップ 1	<b>configureterminal</b> 例： Switch# 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 2	<b>router lisp</b> 例： Switch(config)# router lisp	LISP 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	<b>service ipv4</b> 例： Switch(config-router-lisp)# service ipv4	IPv4 アドレス ファミリに対してレイヤ 3 ネットワーク サービスをイネーブルにし、サービスサブモードを開始します。
ステップ 4	<b>map-cache destination-eidmap-request</b> 例： Switch(config-router-lisp-service)# map-cache 10.1.0.0/16 map-request	map-request が送信される宛先 EID を指定します。
ステップ 5	<b>encapsulation vxlan</b> 例： Switch(config-router-lisp-service)# encapsulation vxlan	VXLAN ベースのカプセル化を指定します。
ステップ 6	<b>itr map-resolver ip-address</b> 例： Switch(config-router-lisp-service)# itr map-resolver 2.1.1.6	このデバイスが IPv4 EID-to-RLOC マッピング解決のための Map Request メッセージを送信する LISP Map Resolver のロケータ アドレスを設定します。
ステップ 7	<b>proxy-itr locator-address</b> 例： Switch(config-router-lisp-service)# proxy-itr 7.7.7.7	LISP ITR 機能をイネーブルにします。
ステップ 8	<b>proxy-etr</b> 例： Switch(config-router-lisp-service)# proxy-etr	デバイス上で PETR 機能をイネーブルにします。

	コマンドまたはアクション	目的
ステップ 9	<b>exit-service-ipv4</b> 例： Switch(config-router-lisp-service)# exit-service-ipv4	サービス サブモードを終了します。
ステップ 10	<b>exit-router-lisp</b> 例： Switch(config-router-lisp)# exit-router-lisp	LISP コンフィギュレーションモードを終了します。

## ファブリック コントロール プレーンの設定

### 手順

	コマンドまたはアクション	目的
ステップ 1	<b>configureterminal</b> 例： Switch# 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 2	<b>router lisp</b> 例： Switch(config)# router lisp	LISP 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	<b>site site-name</b> 例： Switch(config-router-lisp)# site fabric	コントロールプレーンデバイスで LISP サイトを設定し、LISP サイト 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	<b>authentication-key key</b> 例： Switch(config-router-lisp-site)# authentication-key lisp	コントロールプレーンデバイスへの登録時にエッジデバイスが送信した <b>map-register</b> メッセージを認証するためのハッシュメッセージ認証コード (HMAC) セキュアハッシュアルゴリズム (SHA-1) ハッシュを作成するために使用するパスワードを設定します。
ステップ 5	<b>eid-record [instance-id instance-id] record [route-tag tag] [accept-more-specifics]</b> 例： Switch(config-router-lisp-site)# eid-record instance-id 30 10.1.0.0/16	コントロールプレーンデバイスへの登録時にエッジデバイスが送信した <b>map-register</b> メッセージで許可されるエンドポイント識別子 (EID) のプレフィックスのホストプールまたはリストを設定します。

	コマンドまたはアクション	目的
ステップ 6	<b>exit-site</b> 例 : Switch(config-router-lisp-site)# exit-site	LISP サイト コンフィギュレーション モードを終了し、LISP コンフィギュレーション モードに戻ります。
ステップ 7	<b>exit-router-lisp</b> 例 : Switch(config-router-lisp)# exit-router-lisp	LISP コンフィギュレーション モードを終了します。

## LISP 設定をトラブルシューティングするための show コマンド

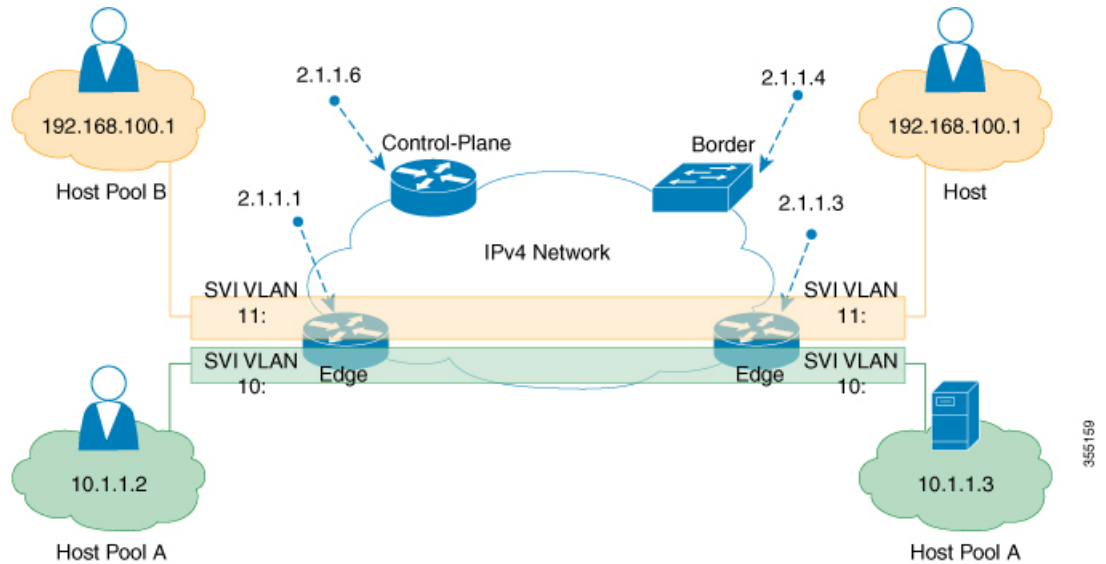
- **show lisp** [*router-lisp-id*] {*instance\_id id* | *eid-table table*} {*ipv4* | *ipv6* | *ethernet*} {*database* | *map-cache* | *server* [*address-resolution*]}
- **show lisp instance-id** *id* *ipv4* *database*
- **show lisp instance-id** *id* *ipv4* *map-cache*
- **show lisp service** *ipv4* *summary*
- **show lisp instance-id** *id* { *ipv4* | *ipv6* | *ethernet*}
- **show lisp instance-id** *id* *dynamic-eid*

## ファブリック エッジ ノードでの LISP 設定の設定例

次のキャンパス ファブリック テクノロジーを検討します。



図 2: キャンパス ファブリック トポロジの例



次に、のファブリック エッジノードでの **show running-configuration** コマンドの出力を示します。 [図 2: キャンパス ファブリック トポロジの例 \(13 ページ\)](#)

```
interface Loopback0
ip address 2.1.1.1 255.255.255.255
!
interface Vlan10
mac-address ba25.cdf4.ad38
ip address 10.1.1.1 255.255.255.0
lisp mobility DEFAULT.EID.eng
end
!
interface Vlan11
mac-address ba25.cdf4.bd38
ip address 192.168.101.1 255.255.255.0
end
!
router lisp
locator-table default
locator-set set1
IPv4-interface Loopback0 priority 1 weight 1
exit-locator-set
!
locator default-set set1
service ipv4
proxy-itr 2.1.1.6
map-cache 0.0.0.0/0 map-request
itr map-resolver 2.1.1.6
etr map-server 2.1.1.6 key foo
etr map-server 2.1.1.6 proxy-reply
etr
use-petr 14.1.1.1
exit-service-ipv4
!
service ethernet
proxy-itr 2.1.1.6
map-cache 0.0.0.0/0 map-request
itr map-resolver 2.1.1.6
```

```

etr map-server 2.1.1.6 key foo
etr map-server 2.1.1.6 proxy-reply
etr
exit-service-ethernet
!
instance-id 30
dynamic-eid DEFAULT.EID.eng
database-mapping 10.1.1.0/24 locator-set set1
exit-dynamic-eid
!
service ipv4
eid-table default
exit-service-ipv4
!
exit-instance-id
!
instance-id 101
service ethernet
eid-table vlan 10
database-mapping mac locator-set set1
map-cache-limit 1000
database-mapping limit dynamic 2000
proxy-itr 2.1.1.6
map-cache 0.0.0.0/0 map-request
itr map-resolver 2.1.1.6
etr map-server 2.1.1.6 key foo
etr map-cache-ttl 10000
etr
exit-service-ethernet
!
exit-instance-id
!
instance-id 102
service ethernet
eid-table vlan 11
database-mapping mac locator-set set1
map-cache-limit 1000
database-mapping limit dynamic 2000

proxy-itr 2.1.1.6
map-cache 0.0.0.0/0 map-request
itr map-resolver 2.1.1.6
etr map-server 2.1.1.6 key foo
etr map-cache-ttl 10000
etr
exit-service-ethernet
!
exit-instance-id
exit-router-lisp
!

```

次に、[図 2: キャンパス ファブリック トポロジの例 \(13 ページ\)](#) のコントロールプレーンでの **show running-configuration** コマンドの出力を示します。

```

interface Loopback0
ip address 2.1.1.6 255.255.255.255
!
router lisp
locator-set WLC // enables wireless and access points to be registered.
3.3.3.20
exit-locator-set
!
service ipv4
map-server

```

```
map-resolver
exit-service-ipv4
!
service Ethernet // enables service ethernet on the map-server
map-server
map-resolver
exit-service-ethernet
!
map-server session passive-open WLC
site Shire
authentication-key cisco123
eid-record 10.1.1.0/24 accept-more-specifics
eid-record 20.1.1.0/24 accept-more-specifics
eid-record instance-id 1 any-mac
exit
!
exit-router-lisp
```

次に、のファブリック ボーダー ノードでの **show running-configuration** コマンドの出力を示します。 [図 2: キャンパス ファブリック トポロジの例 \(13 ページ\)](#)

```
router lisp
locator-set default.RLOC
IPv4-Interface Loopback0 priority 10 weight 10
exit
!
service ipv4
sgt
itr map-resolver 2.1.1.6
proxy-etr
proxy-itr 2.1.1.4
exit-service-ipv4
!
instance-id 0
service ipv4
eid-table default
map-cache 10.1.1.0/24 map-request
map-cache 20.1.1.0/24 map-request
exit-service-ipv4
!
exit-instance-id
!
instance-id 100
service ipv4
eid-table vrf guest
map-cache 192.168.100.0/24 map-request
exit-service-ipv4
!
exit-instance-id
exit-router-lisp
```

## キャンパスファブリックのデータプレーンセキュリティ

キャンパスファブリックデータプレーンセキュリティにより、ファブリックドメイン内からのトラフィックのみを宛先のエッジデバイスによってカプセル化解除できます。ファブリックドメイン内のエッジデバイスと境界デバイスは、データパケットによって伝送される送信元のルーティングロケータ (RLOC)、すなわちアップリンクインターフェイスアドレスが、ファブリックドメインのメンバーであることを確認します。

データプレーンセキュリティにより、カプセル化されたデータパケット内のエッジデバイスの送信元アドレスがスプーフィングされることはありません。ファブリックドメイン以外からのパケットは送信元 RLOC が無効であり、エッジデバイスと境界デバイスによるカプセル化解除時にブロックされます。

## エッジデバイスでのデータプレーンセキュリティの設定

### 始める前に

- デバイスが確実に到達できるように、各エッジデバイスに loopback0 IP アドレスを設定します。 **ip lisp source-locator loopback0** コマンドをアップリンク インターフェイスに適用したことを確認します。
- アンダーレイ設定が設定されていることを確認します。
- エッジデバイス、コントロールプレーンデバイス、およびボーダーデバイスが設定済みであることを確認します。

### 手順

	コマンドまたはアクション	目的
ステップ 1	<b>router lisp</b> 例： Switch(config)# router lisp	LISP 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	<b>instance-id instance-id</b> 例： Switch(config-router-lisp)# instance-id 3	複数のサービスをグループ化する LISP EID インスタンスを作成します。このインスタンス ID の下の設定が、その下位のすべてのサービスに適用されます。
ステップ 3	<b>decapsulation filter rloc source member</b> 例： Switch(config-router-lisp-instance)# decapsulation filter rloc source member	ファブリックドメイン内のカプセル化されたパケットの送信元 RLOC (アップリンク インターフェイス) アドレスの検証を有効にします。
ステップ 4	<b>exit</b> 例： Switch(config-router-lisp-instance)# exit	LISP インスタンス コンフィギュレーション モードを終了し、LISP コンフィギュレーション モードに戻ります。
ステップ 5	<b>exit</b> 例： Switch(config-router-lisp)# exit	LISP コンフィギュレーション モードを終了し、グローバル コンフィギュレーション モードに戻ります。

# コントロールプレーンデバイスでのデータプレーンセキュリティの設定

## 始める前に

- デバイスが確実に到達できるように、各コントロールプレーンデバイスに loopback0 IP アドレスを設定します。Ensure
- **ip lisp source-locator loopback0** コマンドをアップリンク インターフェイスに適用していることを確認します。
- アンダーレイ設定が設定されていることを確認します。
- エッジデバイス、コントロールプレーンデバイス、および境界デバイスを設定済みであることを確認します。

## 手順

	コマンドまたはアクション	目的
ステップ 1	<b>router lisp</b> 例： Switch(config)# router lisp	LISP 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	<b>map-server rloc members distribute</b> 例： Switch(config-router-lisp)# map-server rloc members distribute	ファブリック ドメイン内のエッジデバイスへの、EIDプレフィックスのリストの配布を有効にします。
ステップ 3	<b>exit</b> 例： Switch(config-router-lisp)# exit	LISP コンフィギュレーション モードを終了します。
ステップ 4	<b>show lisp [session [established]   vrf [vrf-name [session [peer-address]]]]</b> 例： Switch# show lisp session	信頼性の高い転送セッションの情報を表示します。複数の転送セッションがある場合は、対応する情報が表示されます。
ステップ 5	<b>show lisp decapsulation filter [IPv4-rloc-address   IPv6-rloc-address] [eid-table eid-table-vrf   instance-id iid]</b> 例： show lisp decapsulation filter	手動で設定または検出されたアップリンク インターフェイス アドレス設定の詳細を表示します。

## ファブリック ボーダー デバイスの設定

ファブリック ボーダー デバイスを設定するには、次の手順に従います。

### 手順

	コマンドまたはアクション	目的
ステップ 1	<b>configureterminal</b> 例： Switch# 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 2	<b>router lisp</b> 例： Switch(config)# router lisp	LISP 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	<b>service ipv4</b> 例： Switch(config-router-lisp)# service ipv4	IPv4 アドレス ファミリに対してレイヤ 3 ネットワーク サービスをイネーブルにし、サービスサブモードを開始します。
ステップ 4	<b>map-cache destination-eidmap-request</b> 例： Switch(config-router-lisp-service)# map-cache 10.1.0.0/16 map-request	map-request が送信される宛先 EID を指定します。
ステップ 5	<b>encapsulation vxlan</b> 例： Switch(config-router-lisp-service)# encapsulation vxlan	VXLAN ベースのカプセル化を指定します。
ステップ 6	<b>itr map-resolver ip-address</b> 例： Switch(config-router-lisp-service)# itr map-resolver 2.1.1.6	このデバイスが IPv4 EID-to-RLOC マッピング解決のための Map Request メッセージを送信する LISP Map Resolver のロケーター アドレスを設定します。
ステップ 7	<b>proxy-itr locator-address</b> 例： Switch(config-router-lisp-service)# proxy-itr 7.7.7.7	LISP ITR 機能をイネーブルにします。
ステップ 8	<b>proxy-etr</b> 例： Switch(config-router-lisp-service)# proxy-etr	デバイス上で PETR 機能をイネーブルにします。

	コマンドまたはアクション	目的
ステップ 9	<b>exit-service-ipv4</b> 例： Switch(config-router-lisp-service)# exit-service-ipv4	サービス サブモードを終了します。
ステップ 10	<b>exit-router-lisp</b> 例： Switch(config-router-lisp)# exit-router-lisp	LISP コンフィギュレーションモードを終了します。

## キャンパス ファブリックでのセキュリティ グループ タグとポリシーの適用

キャンパス ファブリック オーバーレイは、ファブリック ドメイン内のデバイス間で送信元グループタグ (SGT) を伝達します。パケットは、仮想拡張LAN (VXLAN) を使用してカプセル化され、ヘッダーに SGT 情報を伝えます。エッジデバイスの IP アドレスにマッピングされた SGT は、カプセル化されたパケット内に運ばれ、宛先デバイスに伝達されます。このデバイスでは、パケットがカプセル化解除され、送信元グループ アクセス コントロール リスト (SGACL) のポリシーが適用されます。

Cisco TrustSec と送信元グループ タグの詳細については、『[Cisco TrustSec Switch Configuration Guide](#)』を参照してください。

## キャンパス ファブリック オーバーレイを使用したマルチキャスト

キャンパス ファブリック オーバーレイを使用して、ネイティブのマルチキャスト機能のないコア ネットワークを介してマルチキャストトラフィックを伝送することができます。キャンパス ファブリック オーバーレイによって、エッジデバイスにヘッドエンドを複製してマルチキャストトラフィックのユニキャスト転送が可能になります。



(注) キャンパス ファブリックでサポートされるのは、Protocol Independent Multicast (PIM) スパースモードおよび PIM Source Specific Multicast (SSM) のみです。デンスモードはサポートされません。

## LISP マルチキャストについて

LISPL マルチキャストには、次の機能があります。

- LISP EID としてのマルチキャスト送信元アドレスのマッピング（宛先グループアドレスはトポロジに依存しないため）。
- LISP オーバーレイでマルチキャスト配信ツリーを生成。
- ルートの入力トンネル ルータ サイト内の送信元から受信者の出力トンネル ルートへのマルチキャスト データ パケットのユニキャスト ヘッドエンド レプリケーション。
- ユニキャスト レプリケーションでは Any Source Multicast (ASM) と Source Specific Multicast (SSM) のサービス モデルをサポート。マルチキャスト レプリケーションではコア ツリーでの SSM のみをサポート。
- LISP および非 LISP 対応の送信元と受信者サイトのさまざまな組み合わせをサポート。
- ヘッドエンド レプリケーション マルチキャスト モードで IPv6 EID をサポート。
- IPv6 マルチキャスト ルーティングは、デフォルトの VRF でのみサポート。
- デフォルトでは、IPv6 マルチキャストは IPv6 インターフェイスでイネーブルになっています。したがって、EID 側インターフェイスには明示的な IPv6 マルチキャスト設定は必要ありません。



- (注) LISP xTR が、PIM ファースト ホップ ルータ (FH) または ランデブー ポイント (RP) であり、そのデバイスがトラフィックを受信のみしている場合、デバイスの少なくとも 1 つのインターフェイスがローカルの LISP データベース マッピングによりカバーされていることを確認します。正しいアドレスが選択されていることを確認するための追加設定は必要ありません。

## IPv4 レイヤ 3 LISP マルチキャストの設定

手順

	コマンドまたはアクション	目的
ステップ 1	<b>configureterminal</b> 例： Switch# 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 2	<b>ip multicast-routing</b> 例： Switch(config)# ip multicast-routing	IP マルチキャスト ルーティングをイネーブルにします。



	コマンドまたはアクション	目的
ステップ 3	次のいずれか 1 つを入力します。 <ul style="list-style-type: none"> <li>• <b>ip pim rp-address</b> <i>rp-address</i></li> <li>• <b>ip pim ssm</b> {<b>default</b>   <b>range</b> {<i>access-list-name</i>   <i>access-list-number</i>}}</li> </ul> 例： <pre>Switch(config-if)# ip pim rp-address 66.66.66.66</pre>	マルチキャストグループの PIMRP のアドレスを静的に設定します。  IP マルチキャストアドレスの Source Specific Multicast (SSM) 範囲を定義します。
ステップ 4	<b>interface</b> <i>LISP-interface number</i> 例： <pre>Switch(config-if)# interface lisp0</pre>	PIM スパース モードをイネーブルにする LISP インターフェイスおよびサブインターフェイスを指定します。
ステップ 5	次のいずれか 1 つを入力します。 <ul style="list-style-type: none"> <li>• <b>ip pim sparse-mode</b></li> <li>• <b>ip pim transport multicast</b></li> </ul> 例： <pre>Switch(config-if)# ip pim sparse-mode</pre>	スパース モード動作のインターフェイスで PIM をイネーブルにします。  スパース モード動作のインターフェイスで PIM をイネーブルにします。コア ネットワークにネイティブのマルチキャスト機能がある場合は、 <b>ip pim transport multicast</b> コマンドを使用します。
ステップ 6	<b>exit</b> 例： <pre>Swtich(config-if)# exit</pre>	インターフェイスコンフィギュレーション モードを終了し、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 7	<b>interface</b> <i>interface-type interface-number</i> 例： <pre>Switch(config)# interface GigabitEthernet0/1</pre>	エンドポイント側のインターフェイスを設定し、インターフェイスコンフィギュレーション モードを開始します。
ステップ 8	<b>ip pim sparse-mode</b> 例： <pre>Switch(config-if)# ip pim sparse-mode</pre>	スパース モード動作のインターフェイスで PIM をイネーブルにします。
ステップ 9	<b>end</b> 例： <pre>Swtich(config-if)# end</pre>	現在のコンフィギュレーションセッションを終了して、特権 EXEC モードに戻ります。

## レイヤ2オーバーレイ ブロードキャストの設定

### 手順

	コマンドまたはアクション	目的
ステップ 1	<b>configureterminal</b> 例： Switch# 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 2	<b>router lisp</b> 例： Switch(config)# router lisp	LISP 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	<b>instance-id instance</b> 例： Switch(config-router-lisp)# instance-id 0	複数のサービスをグループ化する LISP EID インスタンスを作成します。この instance-id での設定が、下位のすべてのサービスに適用されます。
ステップ 4	<b>service ethernet</b> 例： Switch(config-router-lisp-instance)# service ethernet	レイヤ2ネットワークサービスをイネーブルにし、サービス サブモードを開始します。
ステップ 5	<b>eid-table vlan vlan-number</b> 例： Switch(config-router-lisp-instance-service)# eid-table vlan 3	以前に設定した LISP インスタンス ID を、エンドポイント識別子のアドレス空間の VLAN に関連付けます。
ステップ 6	<b>broadcast-underlay multicast-group</b> 例： Switch(config-router-lisp-instance-service)# broadcast-underlay 225.1.1.1	オーバーレイ レイヤ2ブロードキャストトラフィックを伝送するように、アンダーレイが使用するマルチキャストグループを指定します。
ステップ 7	<b>exit-service-ethernet</b> 例： Switch(config-router-lisp-instance-service)# exit-service-ethernet	サービス サブモードを終了します。
ステップ 8	<b>exit-instance-id</b> 例： Switch(config-router-lisp-instance)# exit-instance-id	インスタンス モードを終了します。

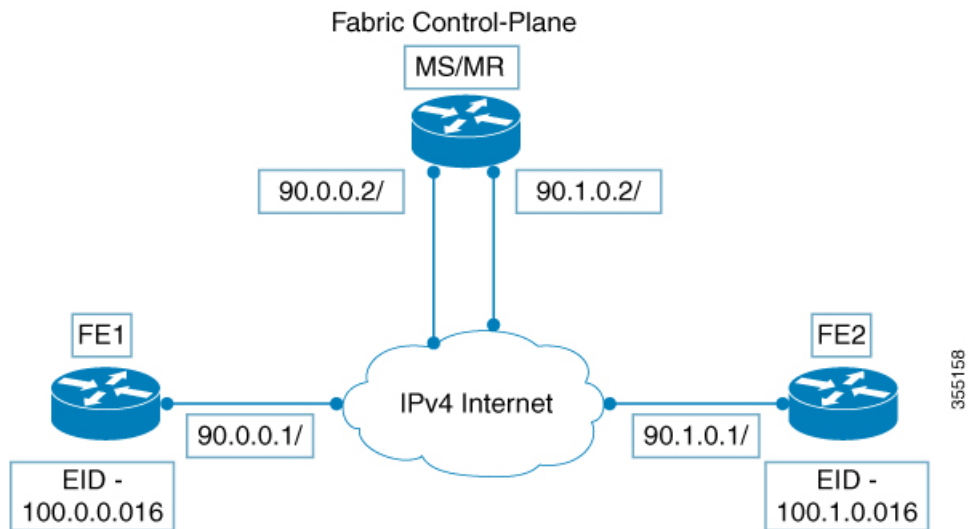
## LISP マルチキャスト設定をトラブルシューティングするための show コマンド

- `show ip pim vrf vrf_name rp mapping`
- `show ip pim vrf vrf_name neighbor`
- `show ip pim vrf vrf_name tunnel`
- `show ip mroute vrf vrf_name`
- `show ip mfib vrf vrf_name`
- `show ip mfib vrf vrf_name count`
- `show ip multicast interface`

## LISP マルチキャストの設定例

キャンパス ファブリックの次のトポロジを検討します。

図 3: マルチキャストを設定するキャンパス ファブリック トポロジ



次に、図のファブリック エッジ ノード FE1 での LISP マルチキャストの設定例を示します。

図 3: マルチキャストを設定するキャンパス ファブリック トポロジ (23 ページ)

```
ip multicast-routing
ip pim ssm default
!
interface Loopback0
ip address 11.1.1.1 255.0.0.0
!
interface Loopback100
ip address 66.66.66.66 255.255.255.255
ip pim sparse-mode
!
interface GigabitEthernet0/1
```

```

ip address 90.0.0.1 255.255.255.0
ip pim sparse-mode
!
Interface Vlan100
ip address 100.0.0.1 255.255.0.0
no ip redirects
ip local-proxy-arp
ip pim sparse-mode
ip route-cache same-interface
no lisp mobility liveness test
lisp mobility vl_100
ip pim sparse-mode
!
interface GigabitEthernet1/0/1
switchport access vlan 100
switchport mode access
!
!
interface LISP0
ip pim sparse-mode
ip pim lisp transport multicast

!
router lisp
locator-table default
locator-set rloc_1
  IPv4-interface Loopback0 priority 1 weight 1
  exit-locator-set
!
instance-id 0
  dynamic-eid vl_100
  database-mapping 100.0.0.0/16 locator-set rloc_1
  exit-dynamic-eid
!
  service ipv4
  eid-table default
  database-mapping 66.66.66.66/32 locator-set rloc_1
  itr map-resolver 30.3.1.1
  itr
  etr map-server 30.3.1.1 key lisp
  etr
  use-petr 14.1.1.1
  exit-service-ipv4
!
  exit-instance-id
!
encapsulation vxlan
exit-router-lisp
!
ip pim rp-address 66.66.66.66

```

次に、のコントロールプレーン (MS/MR) の設定例を示します。 [図 3: マルチキャストを設定するキャンパス ファブリック トポロジ \(23 ページ\)](#)

```

interface Loopback0
ip address 30.3.1.1 255.255.255.255
!
interface GigabitEthernet0/1
ip address 90.0.0.2 255.255.255.0
Ip pim sparse-mode
!
interface GigabitEthernet0/2
ip address 90.1.0.2 255.255.255.0
Ip pim sparse-mode

```

```

!
router lisp
site Fabric
authentication-key lisp
eid-record 100.0.0.0/16 accept-more-specifics
eid-record 66.66.66.66/32 accept-more-specifics
eid-record 77.77.77.77/32 accept-more-specifics
eid-record 88.88.88.88/32 accept-more-specifics
exit
!
ipv4 map-server
ipv4 map-resolver
exit
    
```

## キャンパス ファブリックの機能履歴

リリース	変更箇所
Cisco IOS XE Everest 16.6.1	<b>auto</b> コマンドのサポートを削除しました。 新しいモードの CLI が導入されました。





## 第 2 章

# キャンパス ファブリック内での DHCP の設定

- [キャンパス ファブリックの DHCP 設定 \(27 ページ\)](#)
- [DHCP パケットフロー \(28 ページ\)](#)
- [キャンパス ファブリック ネットワーク内の DHCP クライアントへの IP アドレス割り当てにおける操作のシーケンス \(29 ページ\)](#)
- [DHCP クライアント/サーバの設定方法 \(30 ページ\)](#)
- [DHCP 設定例 \(33 ページ\)](#)
- [キャンパス ファブリックでの DHCP ソリューションの機能履歴 \(36 ページ\)](#)

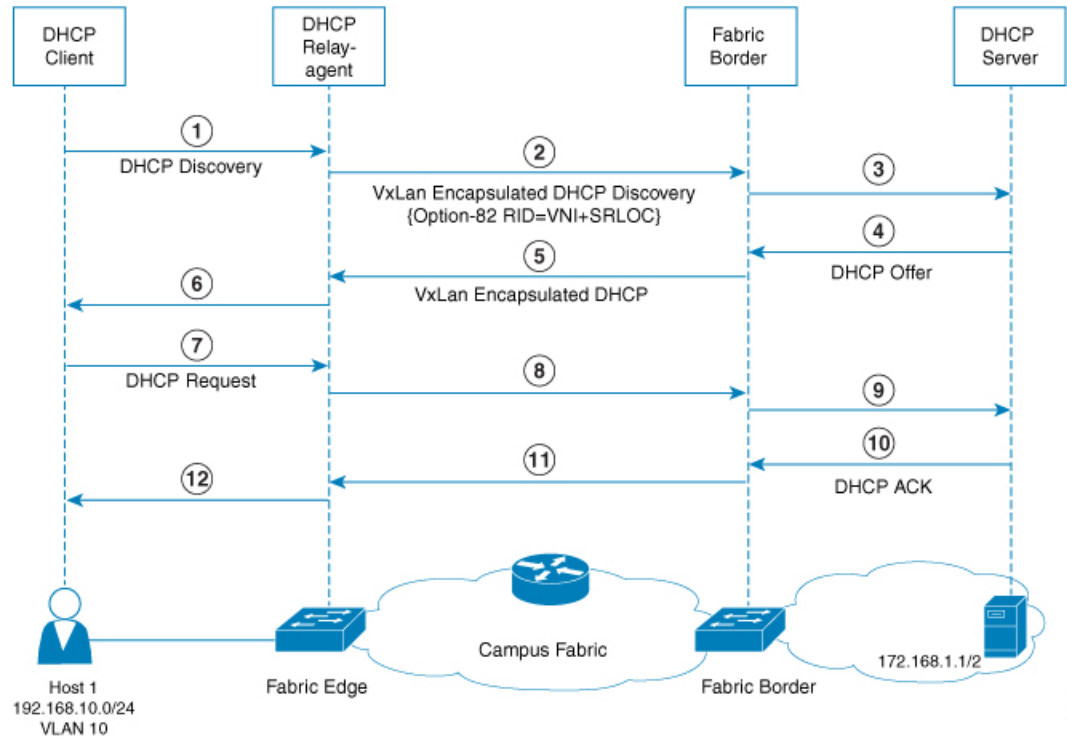
## キャンパス ファブリックの DHCP 設定

キャンパス ファブリック ネットワークでは、ファブリック エンドポイントとは異なるネットワーク内に位置する共有サービスとして DHCP サーバが導入されます。どのファブリック エッジも、ファブリック エンドポイントと DHCP サーバ間で DHCP トラフィックをリレーする DHCP リレーエージェントとして設定されます。DHCP サーバはエンタープライズファブリック ネットワーク内の EID 以外のスペースに位置します。また、ファブリック エッジ ノードはファブリック ボーダーをプロキシトンネルルータ (PxTR) として使用して DHCP サーバと通信します。

キャンパス ファブリック内での DHCP ソリューションの導入は、ファブリック エニーキャスト ゲートウェイ モデルに基づいて行われ、クライアントのゲートウェイ IP はすべてのファブリック エッジ ノードに設定されたエニーキャスト スイッチ仮想インターフェイス (SVI) の IP アドレスです。DHCP はエニーキャスト アドレス サポートとネットワーク アドレス透過性を備えたレイヤ 3 オーバーレイに実装されます。

# DHCP パケットフロー

図 4:



DHCP 向けに Option 82 リモート ID サブオプションを実装するこのトポロジでは、次のようになります。

- ファブリック エッジ ノードは、LISP 入力または出力トンネル ルータ (xTR) としてロケータ アドレス 1.1.1.1 を使用して設定されます。
- ファブリック ボーダー ノードは、LISP プロキシ トンネル ルータ (PxTR) として設定されます。
- ホスト 1 はプレフィックスが 192.168.10.0/24 のファブリック エッジ VLAN 10 に接続された DHCP クライアントです。
- レイヤ3 インターフェイス (SVI) はモビリティ サブネットである インターフェイス VLAN 10 に接続します。
- DHCP リレー エージェントは、ファブリック エッジ ノード上で SVI VLAN 10 用に設定されます。
- DHCP サーバはネイティブ ネットワークに接続され、そのアドレスは 172.168.1.1/24 で、ファブリック ボーダー ノードを介して到達できます。



# キャンパス ファブリック ネットワーク内の DHCP クライアントへの IP アドレス割り当てにおける操作のシーケンス

## DHCP クライアント：（ホスト 1）

1.ホスト 1 は DHCP 検出メッセージを生成し、ネットワーク上でブロードキャストします。

## DHCP リレー エージェント

2.DHCP リレー エージェント（ファブリック エッジ ノード）はパケットを傍受して、パケット内に次のフィールドを設定します。

- GIADDR：着信ユニキャスト SVI インターフェイス IP アドレス（192.168.10.1）に設定します。

- Option 82 リモート ID サブオプション：クライアントセグメントに関連付けられた「SRLOC IPv4 アドレス」と「VxLAN L3 VNI ID」としてエンコードされた文字列。

ロケータ アドレスは 1.1.1.1 に設定されます。

L3 VNI ID は 20 に設定されます。

- 回路 ID サブオプション：VLAN-ポート-モジュール形式でエンコードされ、VLAN は 10、ポート/モジュールは着信ポートとスイッチ番号に設定されます。

3.内部 DHCP 送信元アドレス、内部 VXLAN Mac ヘッダー、VXLAN ヘッダー、UDP ヘッダー、外部 IP ヘッダー、および外部 L2 ヘッダーを書き換えて DHCP メッセージを構築します。次に、この VXLAN カプセル化 DHCP ユニキャスト パケットをファブリック ボーダー ノードに転送します。

## ファブリック ボーダー ノード：

4.ファブリック ボーダー デバイスは VXLAN カプセル化された DHCP パケットのカプセル化を解除し、DHCP サーバアドレス宛てのそのパケットをネクストホップ ルータにネイティブに転送します。

## DHCP サーバ：

5.DHCP リレー エージェントから DHCP パケットを受信した後、次のプロセスが DHCP サーバ上で実行されます。

- DHCP サーバは着信メッセージに設定された GIADDR の値（192.168.10.1）に基づいて IP プール（192.168.10.0/24）を選択します。

- IP プールから IP アドレス（192.168.10.2）を割り当てます。

- 受信した GIADDR の値に設定された宛先アドレスで DHCP オファー メッセージを生成します。これは、回路 ID とリモート ID を含むオプション 82 サブオプションでピギーバックされます。

6.DHCPサーバは、ファブリック ボーダーを通じて DHCP リレー エージェントに対して DHCP 応答パケットをルーティングします（ファブリック ボーダーは、ファブリックへのすべてのインバウンドトラフィックのエントリ ポイントです）。

**ファブリック ボーダー ノード :**

7.LISP PxTRとして設定されたファブリック ボーダー ノードは、ファブリック サブネット宛てのすべてのパケットの入力 LISP トンネルルータとして動作します。DHCP リレー エージェント アドレス宛ての DHCP 応答メッセージを受信すると、ファブリック ボーダー デバイスは DHCP オファーメッセージ VXLAN をオプション 82 リモート ID フィールド（Src RLOC IP フィールドと VNI フィールド）を使用してカプセル化し、DHCP リレー エージェントに転送します。

**DHCP リレー エージェント :**

8.DHCP リレー エージェントは DHCP オファー パケットを受信して処理し、クライアントに転送します。

**DHCP クライアント :**

9.DHCP クライアントは DHCP オファー パケットを受信し、DHCP 要求パケットを開始して、IP アドレス（192.168.10.2）を要求します。

次に DHCP 要求パケットは、DHCP サーバに到達するまで、手順 2～4 で説明したとおりに処理されます。

DHCP サーバ DHCP 要求パケットの通常処理を実行し、DHCP ACK を DHCP リレー エージェントに送信します。DHCP ACK は、手順 5～9 に示したとおりの転送手順に従います。

## DHCP クライアント/サーバの設定方法

次の設定は、任意の順序で実行できます。ホストをオンボーディングする前に、必ずデバイスを設定してください。

## DHCP リレー エージェントとしてのファブリック エッジノードの設定

次に、DHCP リレー エージェントとしてファブリック エッジを設定する手順について説明します。キャンパス ファブリックでの DHCP クライアント/サーバの設定の詳細については、『Cisco IOS XE 16.6.1 Configure DHCP for Campus Fabric』のドキュメントを参照してください。

手順

	コマンドまたはアクション	目的
ステップ 1	<b>configureterminal</b> 例： Switch# 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 2	<b>ip dhcp snooping</b> 例： Switch(config)# ip dhcp snooping	DHCP スヌーピングをグローバルにイネーブル化します。
ステップ 3	<b>ip dhcp snooping vlan</b> 例： Switch(config-if)# ip dhcp snooping vlan	指定した VLAN で DHCP スヌーピングをイネーブルにします。
ステップ 4	<b>ip dhcp relay information option</b> 例： Switch(config-if)# ip dhcp relay information option	DHCP サーバへ転送したメッセージに、システムが DHCP リレー エージェント情報オプション (Option 82 フィールド) を挿入できるようにします。

## エニーキャスト SVI としてのファブリック エッジノードの設定

手順

	コマンドまたはアクション	目的
ステップ 1	Switch# <b>configure terminal</b>	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	Switch(config)# <b>interface interface</b>	SVI 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	Switch(config-if)# <b>ip vrf forwarding vrf-name</b>	インターフェイス上で VRF を設定します。
ステップ 4	Switch(config-if)# <b>ip address ip address</b>	インターフェイスの IP アドレスを設定します。
ステップ 5	Switch(config-if)# <b>ip helper-address ipaddress</b>	DHCP ブロードキャストは、ルータによってドロップされるのではなく、この特定のヘルパアドレスにユニキャストとして転送されます。

	コマンドまたはアクション	目的
ステップ 6	Switch(config-if)# <b>lisp mobility dynamic-EID</b>	LISP 仮想マシンのモビリティであるダイナミック EID ローミングに参加するようにインターフェイスを設定します。
ステップ 7	Switch(config-if)# <b>no lisp mobility liveness test</b>	インターフェイスで活性テストをディセーブルにします。

## ファブリック エッジノードでの LISP の設定

ファブリック エッジデバイスを設定するには、次の手順を実行します。

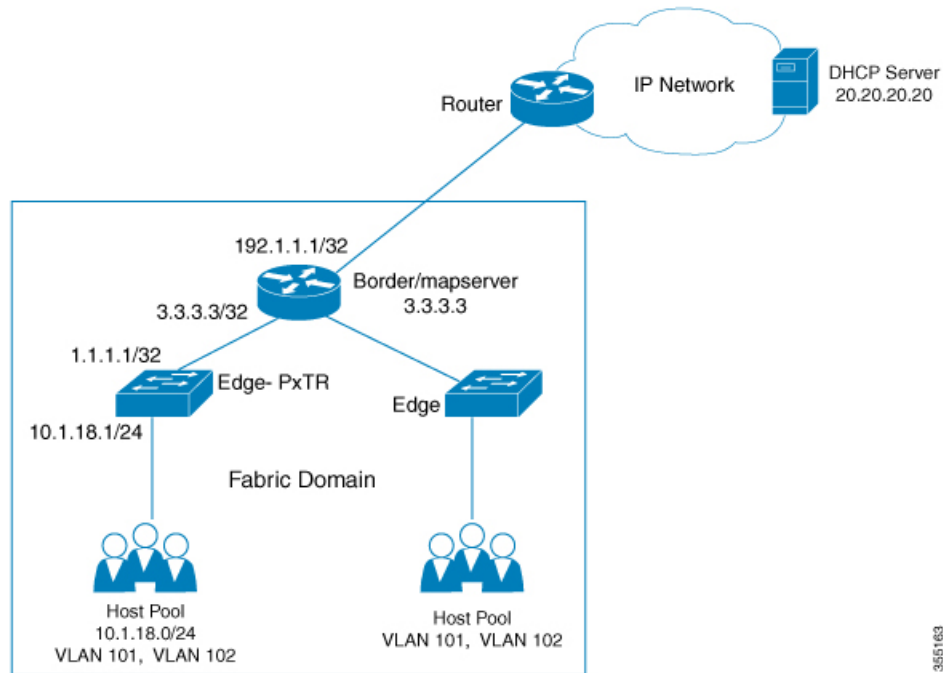
### 手順

	コマンドまたはアクション	目的
ステップ 1	switch# <b>configureterminal</b>	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	switch(config)# <b>router lisp</b>	LISP 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	Switch(config-router-lisp)# <b>locator-set name</b>	名前付きロケータ セットを指定します。
ステップ 4	Switch(config-router-lisp-locator-set)# <b>IPv4-interface loopback</b> <i>Loopback-address</i> { <b>priority</b> <i>priority_value</i>   <b>weight</b> <i>weight</i> }	ループバック IP アドレスを設定して、デバイスが到達可能であることを確認します。
ステップ 5	Switch(config-router-lisp-locator-set)# <b>exit-locator-set</b>	ロケータ設定コンフィギュレーションモードを終了します。
ステップ 6	Switch(config-router-lisp)# <b>instance-id instance</b>	複数のサービスをグループ化する LISP EID インスタンスを作成します。この <b>instance-id</b> での設定が、下位のすべてのサービスに適用されます。
ステップ 7	Switch(config-router-lisp-instance)# <b>dynamic-eid dynamic-EID</b>	ダイナミック EID ポリシーを作成し、ダイナミック EID コンフィギュレーションモードを開始します。
ステップ 8	Switch(config-router-lisp-instance-dynamic-eid)# <b>base-mapping eid locator-set RLOC name</b>	EID を RLOC マッピング関係に設定します。
ステップ 9	Switch(config-router-lisp-instance-dynamic-eid)# <b>exit-dynamic-eid</b>	ダイナミック EID コンフィギュレーションモードを終了します。

	コマンドまたはアクション	目的
ステップ 10	Switch(config-router-lisp-instance)# <b>service ipv4</b>	IPv4 アドレス ファミリに対してレイヤ 3 ネットワーク サービスをイネーブルにし、サービスサブモードを開始します。
ステップ 11	Switch(config-router-lisp-instance-service)# <b>eid-table vrf vrf-table</b>	以前に設定した LISP インスタンス ID と、エンドポイント識別子アドレス空間に到達できるようにする仮想ルーティングおよび転送 (VRF) テーブルを関連付けます。
ステップ 12	Switch(config-router-lisp-instance-service)# <b>map-cache destination-eidmap-request</b>	宛先 EID にスタティック マップ要求を生成します。
ステップ 13	Switch(config-router-lisp-instance-service)# <b>itr map-resolver map-resolver-address</b>	宛先エンドポイント識別子 (EID) の IP に対応する RLOC を照会する必要がある マップリゾルバ IP を設定します。
ステップ 14	Switch(config-router-lisp-instance-service)# <b>itr</b>	このデバイスが入力トンネルルータ (ITR) として動作することを指定します。
ステップ 15	Switch(config-router-lisp-instance-service)# <b>etr map-server map-server-addrkey {0   6} authentication key</b>	IPv4 エンドポイント識別子の登録時に出力トンネルルータ (ETR) が使用する LISP マップサーバのロケータ アドレスを設定します。
ステップ 16	Switch(config-router-lisp-instance-service)# <b>etr</b>	このデバイスが出力トンネルルータ (ETR) として動作することを指定します。
ステップ 17	Switch(config-router-lisp-instance-service)# <b>use-petr locator-address { priority priority_value   weight weight_value}</b>	プロキシ出力トンネルルータ (PETR) を使用するようにデバイスを設定します。
ステップ 18	Switch(config-router-lisp-instance-service)# <b>exit service ipv4</b>	サービス サブモードを終了します。
ステップ 19	Switch(config-router-lisp-instance)# <b>exit-instance-id</b>	インスタンス サブモードを終了します。

## DHCP 設定例

次のトポロジを検討します。



355163

### ファブリック エッジ ノードでのループバック 0 の設定

```
Configure terminal
interface loopback 0
ip address 1.1.1.1/32
exit
```

オーバーレイ内で送信する DHCP 要求に 0/0 マップキャッシュを使用してプロキシ ITR としてのファブリック エッジを設定します。

```
router lisp
 locator-set edged1
 IPv4-interface loopback 0
 exit-locator-set
 !
 instance-id 4098
 dynamic-eid user
 database-mapping 10.1.18.0/24 locator-set edged1
 exit-dynamic-eid
 !
 service ipv4
 eid-table vrf User
 map-cache 0.0.0.0/0 map-request
 itr map-resolver 3.3.3.3
 proxy-itr 1.1.1.1
 etr map-server 3.3.3.3 key uci
 etr
 use-petr 3.3.3.3
 exit-service-ipv4
 !
 exit-instance-id
 !
 exit-router-lisp
```

ファブリック内のすべての VLAN で DHCP スヌーピングをイネーブルにします。

```
ip dhcp relay information option
ip dhcp snooping
ip dhcp snooping vlan 101
```

検出/要求パケットは 20.20.20.20 (DHCP サーバ IP) 宛の VRF 「dhcp」のオーバーレイを介して送信されます。ゲートウェイである SVI の下に DHCP サーバヘルパアドレスを設定します。

```
interface Vlan101
 ip vrf forwarding User
 ip address 10.1.18.1 255.255.255.0
 ip helper-address 20.20.20.20
 no lisp mobility liveness test
 lisp mobility user
end
```

ファブリック エッジでホスト側のポートを設定します。

```
interface GigabitEthernet1/0/38
 description conn_IX_0104
 switchport access vlan 101
 switchport mode access
 spanning-tree portfast
end
```

DHCP サーバが位置するネットワークに接続したマップサーバルータでもあるファブリック ボーダーを設定します。

```
router lisp
 locator-table default
 locator-set border
   IPv4-interface Loopback0 priority 10 weight 10
 !
instance-id 4098
 service ipv4
   eid-table vrf PACAF
   route-export site-registrations
   distance site-registrations 250
   map-cache site-registration
   exit-service-ipv4
 !
exit-instance-id

router bgp 65002
 bgp log-neighbor-changes
 !
address-family ipv4 vrf USER
 aggregate-address 10.1.18.0 255.255.255.0 summary-only
 redistribute lisp metric 10
 neighbor 30.1.1.1 remote-as 200
 exit-address-family
```

ボーダーの VNI ごとにエニーキャスト SVI IP アドレスのループバック インターフェイスを作成し、DHCP サーバから受信した CPU への DHCP パケットのパンティングを促進します。

```
interface Loopback3000
 vrf forwarding User
 ip address 10.1.18.1 255.255.255.255
end
```

エニーキャスト SVI アドレスを BGP ピアにアドバタイズします。

```
router bgp 100
address-family ipv4 vrf User
  bgp router-id 23.1.1.1
  network 10.1.18.1 mask 255.255.255.255
  aggregate-address 10.1.18.0 255.255.0.0 summary-only
  redistribute lisp metric 10
  neighbor 23.1.1.2 remote-as 200
  neighbor 23.1.1.2 ebgp-multihop 3
  neighbor 23.1.1.2 activate
exit-address-family
```

DHCP プールを作成します。DHCP サーバで、デフォルトのルータ IP アドレスが LISP 内の SVI ゲートウェイであることを確認します。

```
ip dhcp excluded-address 10.1.18.1
ip dhcp excluded-address 10.1.18.202 10.1.18.255
!

ip dhcp pool User
  network 10.1.18.0 255.255.255.0
  default-router 10.1.18.1
!
```

## キャンパス ファブリックでの DHCP ソリューションの機能履歴

リリース	変更箇所
Cisco IOS XE Everest 16.6.1	この機能が導入されました。