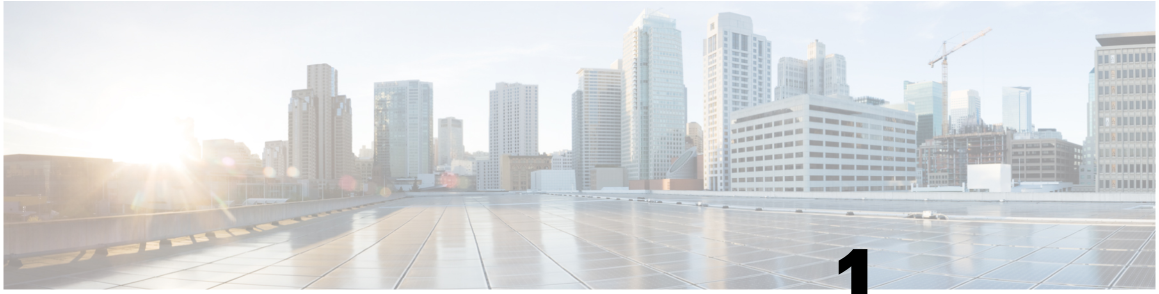
LISP VXLAN Fabric Configuration Guide, Cisco IOS XE Cupertino 17.9.x (Catalyst 9000 Series Switches)

First Published: 2023-03-31

Last Modified: 2023-08-01

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS



CHAPTER 1

LISP VXLAN Fabric Overview

LISP VXLAN Fabric is a wired and wireless connectivity solution offering scalable policy-based segmentation at the network edge.

Note This document describes the configurations required to deploy a LISP VXLAN fabric in a campus network. If you are not familiar with the LISP routing architecture and VXLAN networking, we recommend that you go over the fundamentals of LISP and VXLAN before you proceed with the configurations described below.

What is LISP VXLAN Fabric

A network fabric is made of network devices such as wireless access points, switches, and routers that are interconnected, to transport data to its destination. These physical devices form the underlay network that forwards the traffic. A virtual network is built over the underlay network using tunneling technologies such as VXLAN, and is called an overlay. Endpoints or users are logically connected to the overlay network, which transports the user data.

While there are several routing protocols that enable the transport of data in a fabric, this particular fabric uses a combination of Locator/ID Separation Protocol (LISP) and VXLAN.

The Locator/ID Separation Protocol (LISP) is an overlay routing technology that provides improved routing scalability and dynamic host mobility. LISP works with two separate IP address spaces: one to indicate routing locators (RLOCs) for routing traffic to the external network and a second address called endpoint identifier (EID), which is used to identify the endpoints.

VXLAN, a Layer 2 tunneling mechanism, forms the data plane in the overlay network and uses a MAC-in-IP encapsulation method to carry the data packets through the tunnel.

A LISP VXLAN fabric solution uses virtual networks (overlay networks) that run on a physical network (underlay network). The overlay network creates a logical topology to virtually connect the physical devices that are part of the underlay network. In the underlay network, IP connectivity is established among the physical devices through a routing protocol.

Three fundamental components work together to provision a LISP VXLAN fabric. These enable flexible attachment of devices, data transmission and enhanced security through segmentation and group-based policies:

- Control Plane: Uses LISP for mapping endpoint identity (IP addresses or MAC addresses) to their location within the fabric.
- Data Plane: Uses Virtual Extensible LAN (VXLAN) encapsulation method to transmit data packets.
- Policy Plane: (Optional) Uses Cisco Security Group Tags (SGTs) and Group-Based Policy for microsegmentation.

Benefits of Provisioning a LISP VXLAN Fabric

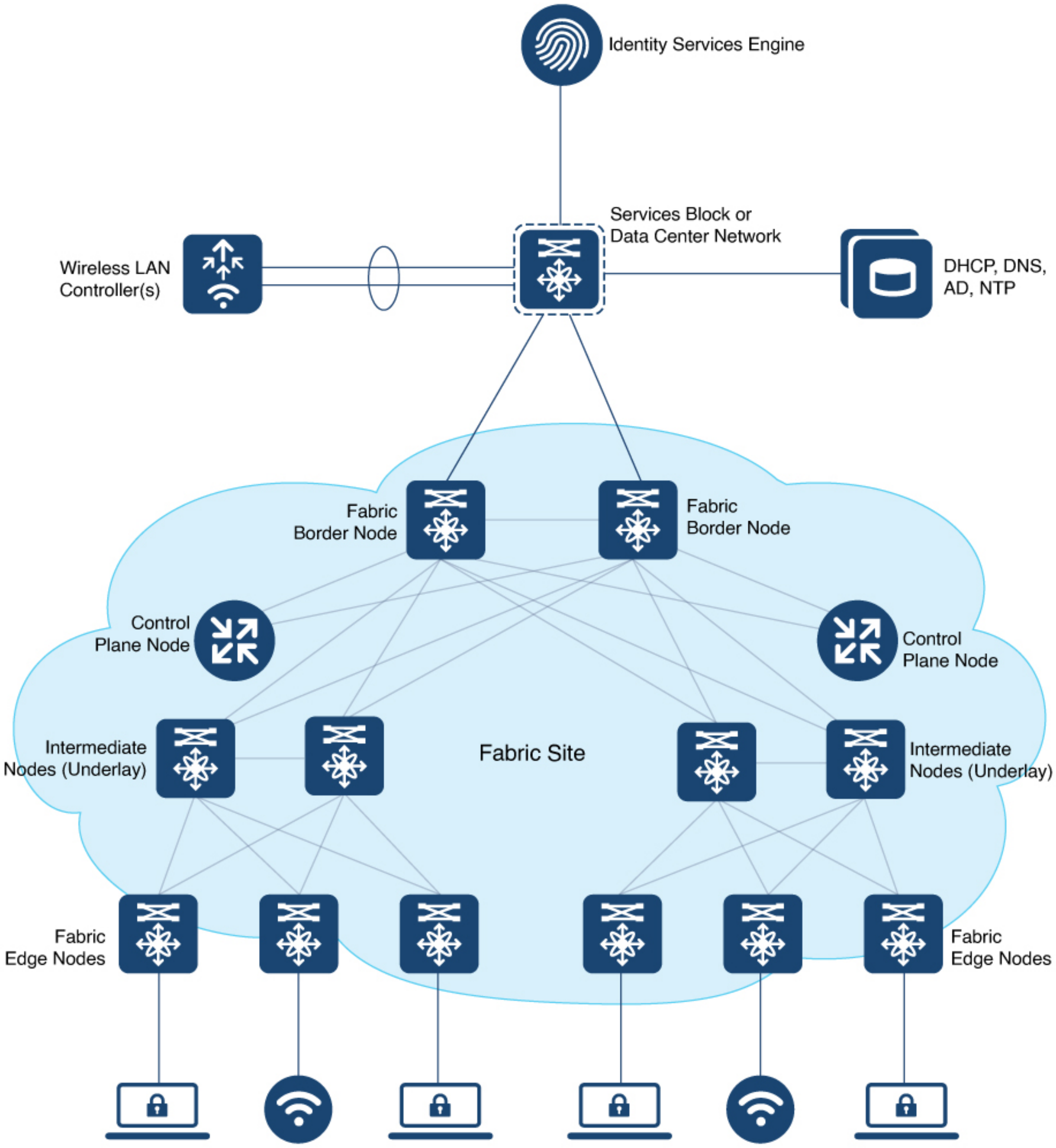
- Use of LISP helps decouple the host address and its location, simplifying the routing operations, and improving scalability.
- Provides end-to-end segmentation using LISP Virtualization technology wherein only the fabric edge and border nodes must be LISP-aware. The rest of the components are just IP forwarders.
- Eliminates Spanning Tree Protocol (STP), improves link utilization, and brings in faster convergence and equal cost multipath (ECMP) load balancing.
- Fabric header (VXLAN) supports Security Group Tag (SGT) propagation, which helps in having a uniform policy model across the network. SGT-based policy constructs are subnet independent.
- Provides host mobility for both wired and wireless clients.

LISP VXLAN Fabric Constructs

The LISP VXLAN fabric comprises wired and wireless devices that make up the underlay and the overlay network. The wired and wireless devices perform different roles, providing end-to-end segmentation enabling efficient traffic movement within the fabric.

Use of Identity Services Engine (ISE) for access control and policy enforcement is optional.

Figure 1: Components of a LISP VXLAN Fabric



- **Fabric Edge Node**: Identifies and authenticates end points and registers end-point ID information in the fabric host-tracking database. These devices encapsulate at ingress and decapsulate at egress, to forward traffic to and from the end points connected to the fabric network.
- **Fabric Border Node**: Serves as the gateway between the fabric and networks external to the fabric. The border node device is physically connected to a transit or to a next-hop device that is connected to the external network. The border node helps translate the reachability and policy information, such as virtual routing and forwarding (VRF) and SGT.

  A fabric border node can be configured as an internal border node, or an external border node, or both internal and external border node.

An internal border node is used for known and registered routes for example, when the traffic needs to go to a datacenter, the LAN or the Shared Services. This internal-only border node advertises the endpoints to the external network and imports external routes into the fabric.

An external border is similar to a default gateway. It is used as a gateway for the traffic from the fabric to unknown destinations or unregistered routes for example, the internet. It advertises the fabric endpoints to the external network but does not import any external routes into the fabric domain.

A border can be both internal and external. An internal and external border is used to access registered and unregistered routes. It advertises the endpoints to the external network and imports external routes into the fabric. It also acts a default gateway for traffic to destinations that are unknown to the control plane database.

- **Fabric Control Plane Node**: Based on the LISP Map-Server and Map-Resolver (MSMR) functionality, a control plane node provides overlay reachability information and end points-to-routing locator (EID-to-RLOC) mapping. A control plane node is a Map Server that receive registrations from fabric edge devices with local end points. A control plane node is also a Map Resolver (MR) that resolves requests from edge devices to locate the remote end points.
- **Intermediate Nodes (Underlay Network)**: Part of the Layer 3 network that physically connects the devices operating in a certain fabric role, such as the interconnection between a border node and an edge node. For example, if a three-tier campus deployment provisions the core switches as the border nodes and the access switches as the edge nodes, the distribution switches are the intermediate nodes. Intermediate nodes simply route and transport IP traffic between the devices operating in fabric roles. The underlay network provides IP reachability, physical connectivity, and supports the additional MTU requirement to accommodate the larger-sized IP packets encapsulated with fabric VXLAN information.
- **Fabric Site**: A network that is composed of a unique set of devices operating in a fabric role (control plane node, border node, edge node) along with the intermediate nodes that are used to connect those devices.
- **Fabric In a Box**: Combines the roles of a border node, a control plane node, and an edge node on the same device. This may be a single switch, a switch with hardware stacking, or a StackWise Virtual deployment. In certain implementations, the same switch can also serve as a Wireless LAN Controller for Fabric-enabled Wireless designs.
- **Wireless LAN Controller**: Provides Access Point image and configuration management, client session management, and mobility services. Additionally, it registers the MAC address of wireless clients in the host tracking database at the time of client join events, as well as updates the location at the time of client roam events.
- **Virtual Network**: Network created in the policy application and provisioned to the fabric nodes as a VRF instance.
- **VXLAN Overlay**: Virtual network that is built over a Layer 3 network by forming a static or dynamic tunnel that runs on top of the physical network infrastructure.
- **Security Group Tag (SGT):** An attribute that is applied to the endpoint traffic to provide logical segmentation based on group membership. When an endpoint connects to a network, it is authenticated and based on the results of the authentication, the network assigns it a specific security group, with the help of SGT.

Fabric Roles Supported by Cisco Catalyst 9000 Series Switches

Platform Family	Fabric Role Support			
	Edge Node	Control Plane Node	Border Node	Embedded 9800 Wireless Controller
Cisco Catalyst 9300 Series	✓	✓	✓	✓
Cisco Catalyst 9400 Series	✓	✓	✓	✓
Cisco Catalyst 9500 Series	✓	✓	✓	✓
Cisco Catalyst 9600 Series	–	✓	✓	–

Deployment Options for a LISP VXLAN Fabric

LISP VXLAN fabric supports the following deployment models:

- A fabric site with multiple control plane nodes and border nodes. The control plane and border nodes are dedicated devices, usually deployed as redundant pairs.
- A fabric site with colocated border and control plane nodes, usually deployed in pairs for redundancy.
- A fabric site with a single device that performs all the fabric roles (control plane, border node, fabric edge node, and a wireless controller). This type of deployment is called a Configuring Fabric In a Box for Wired Devices and is suitable for small deployments such as a branch office.

Prerequisites for Configuring a LISP VXLAN Fabric

- All fabric nodes must have a Loopback interface with an IPv4 address.

  We recommend that the /32 routes of these Loopbacks be propagated by the underlay Interior Gateway Protocol (IGP) throughout the fabric site (without summarization). This is important to quickly detect the fabric edges that are going down.
- All switches in the network including fabric edge, border, control plane, and intermediate nodes should support jumbo MTU. VXLAN header adds 50 bytes of encapsulation to a data packet that is sourced from an endpoint. We recommend an MTU of 9100 to support packet forwarding without fragmentation.
- Ensure that the underlay has routed access network configured.
- Ensure that there is IP reachability between all fabric nodes.

- There should be specific subnet reachability in the underlay (global routing table) for the wireless controller subnet at the access layer. This is required for the access points to connect to the wireless controller irrespective of fabric-enabled wireless or centralized wireless.
- Ensure that all the Cisco Catalyst 9000 Series switches in the fabric operate Cisco IOS XE 17.9.3 or later releases.

  Cisco Identity Services Engine (ISE) operates ISE 3.1 Patch 1 or later releases.

Restrictions for Configuring LISP VXLAN Fabric

- LISP VXLAN fabric solution is supported only on the Cisco Catalyst 9000 Series switches.
- LISP VXLAN fabric underlay network supports only IPv4 addressing. LISP VXLAN overlay network supports both IPv4 and IPv6 addressing. Only the Border Gateway Protocol (BGP) is supported for handoff to external networks.
- Endpoints cannot be assigned to a default instance. (A default instance is an overlay virtual network which connects the infrastructure elements like access points, and Layer 2 switches to the fabric access layer.) Ensure that the endpoint subnets are all assigned to overlay VRFs.
- LISP VXLAN fabric does not support In-Service Software Upgrade (ISSU).
- LISP VXLAN fabric supports only those configurations that are described in this document.

How to Configure LISP VXLAN Fabric

Before you start configuring a LISP VXLAN fabric, ensure that the underlay physical network with the wired devices is configured with routed access.

Configuring a LISP VXLAN Fabric involves the following stages:

1. Configuring a Configuring Control Plane Node node to map the endpoint IDs to their routing locators. A control plane is LISP-based and serves as the Map Server and Map Resolver.
2. Configuring a Configuring Border Node to provide an exchange point for the traffic. A border node is LISP-based and performs the function of the Proxy Tunnel Router.

Note We recommend that you configure both the border and control plane nodes on a single fabric device.

3. Configuring Configuring Fabric Edge Node that are LISP-based and act as ingress and egress tunnel routers for endpoint traffic.
4. Configuring support for Configuring Wireless Support in a LISP VXLAN Fabric infrastructure and endpoints.
5. Configuring Configuring Multicast in LISP VXLAN Fabric in the overlay.

6. Configuring fabric security to provide secure fabric access to the wired and wireless endpoints that connect to the fabric. This involves Configuring Authentication Authorization and Accounting Services and Configuring Group-based Policy on a Fabric Edge on the fabric edge.

Troubleshooting LISP VXLAN Fabric

See Troubleshooting LISP VXLAN Fabric on Cisco Catalyst 9000 Series Switches document to learn how to troubleshoot issues in a LISP VXLAN fabric.

PART I

LISP VXLAN Fabric in a Campus Network

CHAPTER 2

Configuring LISP VXLAN Fabric in a Campus Network

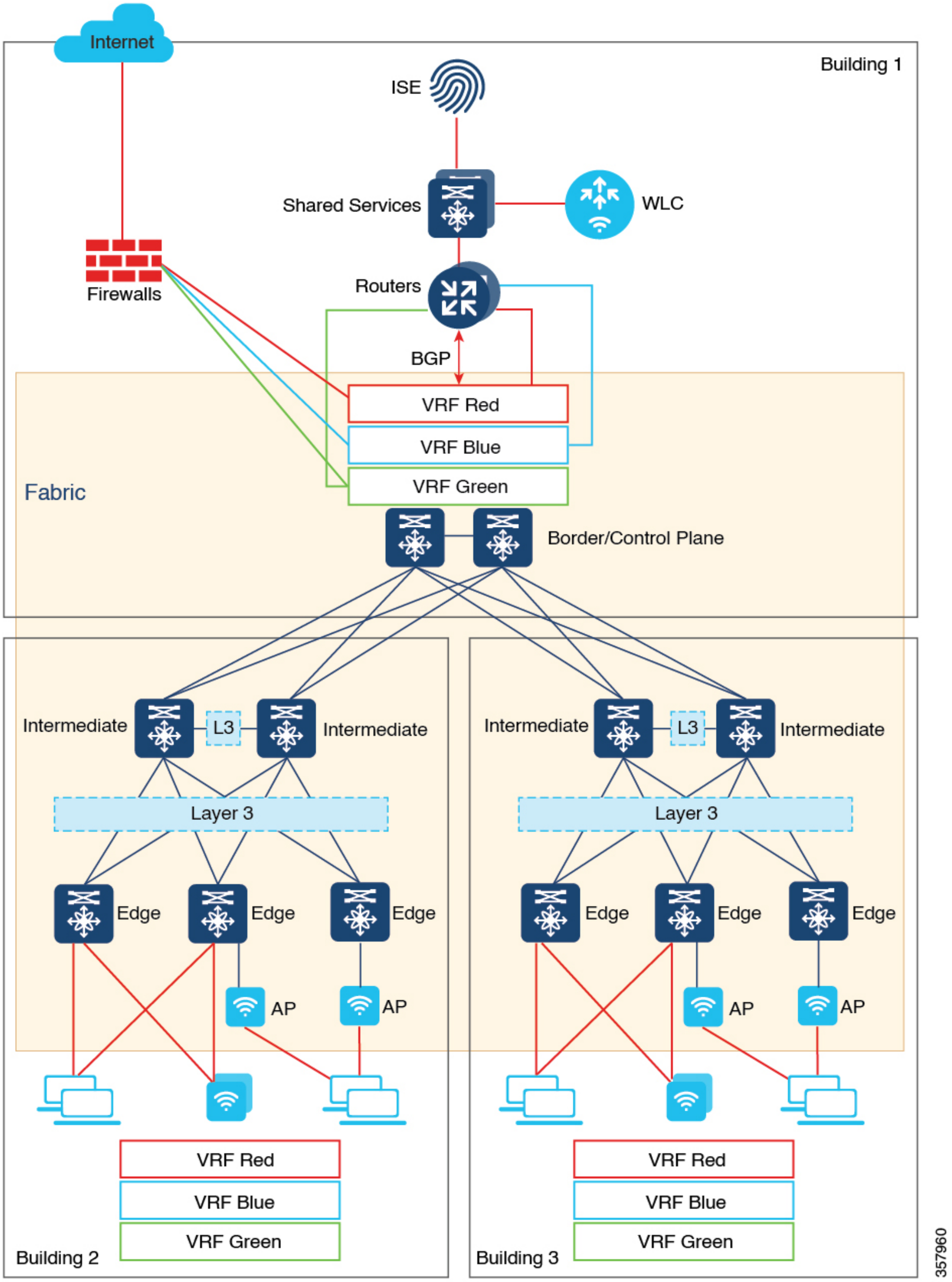
This section describes the configuration of a large fabric site with dedicated devices for control plane node, border node, and edge nodes that connect wired endpoints. All devices in the fabric are a part of the Cisco Catalyst 9000 Series switch family.

LISP VXLAN Fabric Topology for a Campus Network

A campus network could be a building with a three-tier network or a group of buildings comprising multiple distribution blocks. The building blocks of a campus network are a set of interconnected Local Area Networks (LANs).

A LISP VXLAN-based fabric site could span a single large campus or multiple fabric sites within a campus.

Figure 2: LISP VXLAN Topology for Campus Deployment



This topology shows three buildings within a campus. The campus core switches operate as the fabric border and control plane nodes, creating the boundary of the fabric site. The intermediate nodes connect the fabric edge, border, and control plane nodes and provide the Layer 3 underlay for fabric overlay traffic.

Wired clients directly connect to the fabric edge nodes at the access layer. The shared services such as DNS, DHCP, IPAM, and so on are external to the fabric but reside in the global routing table of the campus network. For the endpoints that reside in the overlay virtual network, an inter-VRF route leaking is required to access the shared services in the global routing space. An upstream router provides the inter-VRF route leaking by importing and exporting the routes in different VRF tables to merge them. To maintain the isolation between the different overlay networks, VRF-lite extends from the fabric border nodes to the upstream routers. BGP is the protocol that is used between the fabric border and the upstream routers.

The Shared Services block provides a centralized unit for server and services management in the campus network. End user applications and services such as DNS, DHCP, and so on, are all managed within this Shared Services block.

A wireless controller is located external to the fabric and is connected to the Shared Services unit to manage the wireless clients. The wireless controller also provides Access Point (AP) image and configuration management, client session management, and mobility services.

An AP connects to a fabric edge node and is located in the default instance of the overlay. The AP establishes a CAPWAP control plane tunnel to the wireless controller and joins as local-mode AP. Wireless clients that successfully connect (authenticated and authorised) to an AP are placed in the overlay virtual network.

How to Configure a LISP VXLAN Fabric for Campus Deployment

1. Configure the underlay network with point-to-point routed links between the devices using an Interior Gateway Protocol (IGP). Assign Loopback0 IP addresses to all the fabric nodes. The loopback addresses of the underlay devices need to propagate outside of the fabric to establish connectivity to infrastructure services and, so on.

2. Configuring Control Plane Node to have a mapping system that maps the endpoint IDs to their locators, a Map Server and Map Resolver to accept and respond to queries about the endpoints location, from the network devices.

3. Configuring Border Node to connect to other fabric sites and to the external network.

4. Configuring Fabric Edge Node node to accept endpoint registrations, encapsulate or decapsulate the traffic to and from the fabric, and act as an anycast gateway.

5. Configure support for wireless network:

   A LISP VXLAN fabric supports wireless clients in the following ways:

   - Workflow to Integrate Wireless in a LISP VXLAN Fabric The wireless controller is integrated with the fabric control plane to provide a centralized service for the wired and wireless users. This is the preferred method because it provides the same benefits of a fabric to both the wired and wireless users. Fabric-Enabled Wireless is the recommended deployment model for a large campus network.

   - Configuring OTT Centralized Wireless The control plane traffic and data plane traffic, both traverse using a Control and Provisioning of Wireless Access Points (CAPWAP) tunnel between APs and the wireless controller. The CAPWAP tunnel between wireless controller and an AP traverses the campus backbone network, using the wired fabric as a transport medium.

6. Configure Multicast:

- Configure Configure Layer 2 Overlay Broadcast, Unknown Unicast, and Multicast traffic to be transported over IP multicast in the underlay.
- Configure Configure Layer 3 Overlay Multicast in a LISP VXLAN Fabric.

7. Configure Fabric Security.

   Configuring Authentication Authorization and Accounting Services for the fabric to ensure secure fabric access to the endpoints. The AAA policies are enforced at the fabric edge node where the endpoints connect.

CHAPTER 3

Configuring Control Plane Node

A LISP VXLAN control plane node controls and manages the routing information between the devices in the network. It maintains a host tracking database to identify and map the endpoints' identity with their location information.

The following devices can be configured as control plane nodes:

- Cisco Catalyst 9300 Series Switches
- Cisco Catalyst 9400 Series Switches
- Cisco Catalyst 9500 Series Switches
- Cisco Catalyst 9600 Series Switches

Functions of a Control Plane Node

A fabric control plane node performs the following functions in the fabric:

- **Host Tracking Database (HTDB)**: HTDB is a repository that contains the mapping of an endpoint ID to its routing locator (EID-to-RLOC). Routing locator is the IP address of the loopback interface of the fabric device to which the endpoint is connected. The control plane builds and maintains the HTDB.
- **Endpoint Identifier (EID)**: An EID is an address used for identifying an endpoint device in the network. The endpoint information that is registered by a fabric edge node is updated in the HTDB. HTDB supports IPv4, IPv6, and MAC addresses as endpoint IDs.
- **LISP Map-Server**: The control plane receives endpoint ID map registrations from the edge and border nodes. This information is used to populate the HTDB.
- **LISP Map-Resolver**: The control plane resolves the lookup requests from edge and border nodes, to locate destination endpoint IDs. This tells the requesting device to which fabric node an endpoint is connected and thus where to direct traffic.

How to Configure a Control Plane Node

Note Before you begin, ensure that the underlay network links are configured for routed access connectivity.

Task	Purpose
Configure LISP to build the endpoint identifier (EID) namespace and the routing information table.	• Configure a LISP site to maintain the endpoint ID namespace. A control plane node builds the HTDB using the endpoint information that it receives from the fabric edge nodes. • Configure a Map Server to receive and store the endpoint registrations. • Configure a Map Resolver to resolve a lookup request for route to destination endpoints. Map Resolver tells the requesting device to which fabric node an endpoint is connected and directs the traffic flow from one endpoint to another.

Configure LISP

To configure LISP on a control plane node, perform this task:

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	**router lisp** **Example:** `Device(config)# router lisp`	Enters LISP configuration mode.

Step 4	**locator-table default**
	Example:
	`Device(config-router-lisp)#`
	`locator-table default`
		Selects the default (global) routing table for association with the routing locator address space.

	Command or Action	Purpose

Step 5	**service** {**ipv4** | **ipv6**}
	Example:
	`Device(config-router-lisp)# `**`service ipv4`**
	`Device(config-router-lisp)# `**`service ipv6`**
		Enables network services for the default instance.
		service ipv4: Enables Layer 3 network services for the IPv4 address family.
		service ipv6: Enables Layer 3 network services for the IPv6 address family.

Step 6	**encapsulation vxlan**
	Example:
	`Device(config-router-lisp-serv-ipv4)#`
	`encapsulation vxlan`
	`Device(config-router-lisp-serv-ipv6)#`
	`encapsulation vxlan`
		Specifies VXLAN-based encapsulation for the configured IP address family.

Step 7	**sgt**
	Example:
	`Device(config-router-lisp-serv-ipv4)#`
	`sgt`
	`Device(config-router-lisp-serv-ipv6)#`
	`sgt`
		(Optional) Enables the Security Group Tag (SGT) function for SGT tag propagation, for the configured IP address family. Configure this command only if you need SGT propagation in your fabric network.

Step 8	**map-server**
	Example:
	`Device(config-router-lisp-serv-ipv4)#`
	`map-server`
	`Device(config-router-lisp-serv-ipv6)#`
	`map-server`
		Configures a LISP map server (MS).

Step 9	**map-resolver**
	Example:
	`Device(config-router-lisp-serv-ipv4)#`
	`map-resolver`
	`Device(config-router-lisp-serv-ipv6)#`
	`map-resolver`
		Configures a LISP map resolver (MR).

Step 10	Do one of the following:
	• **exit-service-ipv4**
	• **exit-service-ipv6**
	Example:
	`Device(config-router-lisp-serv-ipv4)#`
	`exit-service-ipv4`
	`Device(config-router-lisp-serv-ipv6)#`
	`exit-service-ipv6`
		Exits service configuration mode, and enters LISP configuration mode.
		Use the appropriate command, depending on which service mode you are exiting from (IPv4 or IPv6 service mode).

Step 11	**service ethernet** **Example:**	Enables Layer 2 network services.

	Command or Action	Purpose
	`Device(config-router-lisp)#` **`service ethernet`**	
Step 12	**map-server** **Example:** `Device(config-router-lisp-serv-eth)#` **`map-server`**	Configures a LISP map server (MS).
Step 13	**map-resolver** **Example:** `Device(config-router-lisp-serv-eth)#` **`map-resolver`**	Configures a LISP map resolver (MR).
Step 14	**exit-service-ethernet** **Example:** `Device(config-router-lisp-serv-eth)#` **`exit-service-ethernet`**	Exits service configuration mode, and enters LISP configuration mode.
Step 15	**site** *site-name* **Example:** `Device(config-router-lisp)#` **`site site_uci`**	Specifies a LISP site and enters LISP site configuration mode. A LISP site name is locally significant to the map server on which it is configured. It has no relevance anywhere else. This name is used solely as an administrative means of associating one or more EID prefixes with an authentication key and other site-related mechanisms.
Step 16	**description** *description* **Example:** `Device(config-router-lisp-site)#` **`description map-server`**	Provides a description for the LISP site.
Step 17	**authentication-key** {*key-type*} *authentication-key* **Example:** `Device(config-router-lisp-site)#` **`authentication-key some-key`**	Configures the password used to create the Hashed Message Authentication Code (HMAC) Secure Hash Algorithm (SHA-1) hash for authenticating the map-register messages sent by edge nodes when registering with the control plane node. Use the following values for *key-type*, depending on the type of authentication desired: • 0: Specifies that an unencrypted password follows • 6: Specifies that an encrypted (AES) password follows

	Command or Action	Purpose
		• 7: Specifies that an encrypted (weak) password follows • <any word>: the unencrypted (cleartext) password **Note** Ensure that you have the same authentication key configured on all the fabric nodes in your network.

Step 18	**eid-record instance-id** *instance-id* [*eid-prefix*] [**accept-more-specifics**]
	Example:
	`Device(config-router-lisp-site)#` **`eid-record instance-id 4099 10.50.1.0/24 accept-more-specifics`**
	`Device(config-router-lisp-site)#` **`eid-record instance-id 8197 any-mac`**
		Configures EID prefixes that are associated with this LISP instance ID. A LISP instance ID is a unique identifier for LISP instance and is associated with a routing table (VRF) or a switching table (VLAN).
		eid-prefix can be IPv4 or IPv6 or MAC EID prefixes.
		accept-more-specifics allows the site to accept registrations for more EID prefixes
		Use this command to configure the EID prefixes that are allowed in a map-register message sent by the edge device when registering with the control plane node. Configure 0.0.0.0/0 as *eid-prefix* for a default instance, if you have to import unregistered prefixes into the LISP database.
		• Repeat this step as necessary to configure additional EID prefixes under the LISP instance.

Step 19	**allow-locator-default-etr instance-id** *instance-id* {**ipv4** | **ipv6**}
	Example:
	`Device(config-router-lisp-site)#` **`allow-locator-default-etr instance-id 4099 ipv4`**
	`Device(config-router-lisp-site)#` **`allow-locator-default-etr instance-id 4099 ipv6`**
		Configures the LISP site to accept default egress tunnel router (ETR) registrations for a particular instance-id and a given service level (IPv4 or IPv6) within that instance-id.
		A default ETR handles the unknown EID prefixes, which are the EID prefixes that are not present in the control plane database. A border node that registers with the control plane node as a default ETR tracks the unknown EID prefixes in each of their VRF tables (a given service level within an instance ID).

Step 20	**exit-site** **Example:**	Exits the LISP Site configuration mode, and enters LISP configuration mode.

	Command or Action	Purpose
	`Device(config-router-lisp-site)# exit-site`	
Step 21	**ipv4 source-locator Loopback** *loopback-interface-number* **Example:** `Device(config-router-lisp)# ipv4 source-locator Loopback0`	Specifies the interface whose IPv4 address should be used as the source locator address for outbound LISP encapsulated packets.
Step 22	**exit-router-lisp** **Example:** `Device(config-router-lisp)# exit-router-lisp`	Exits LISP configuration mode, and enters global configuration mode.

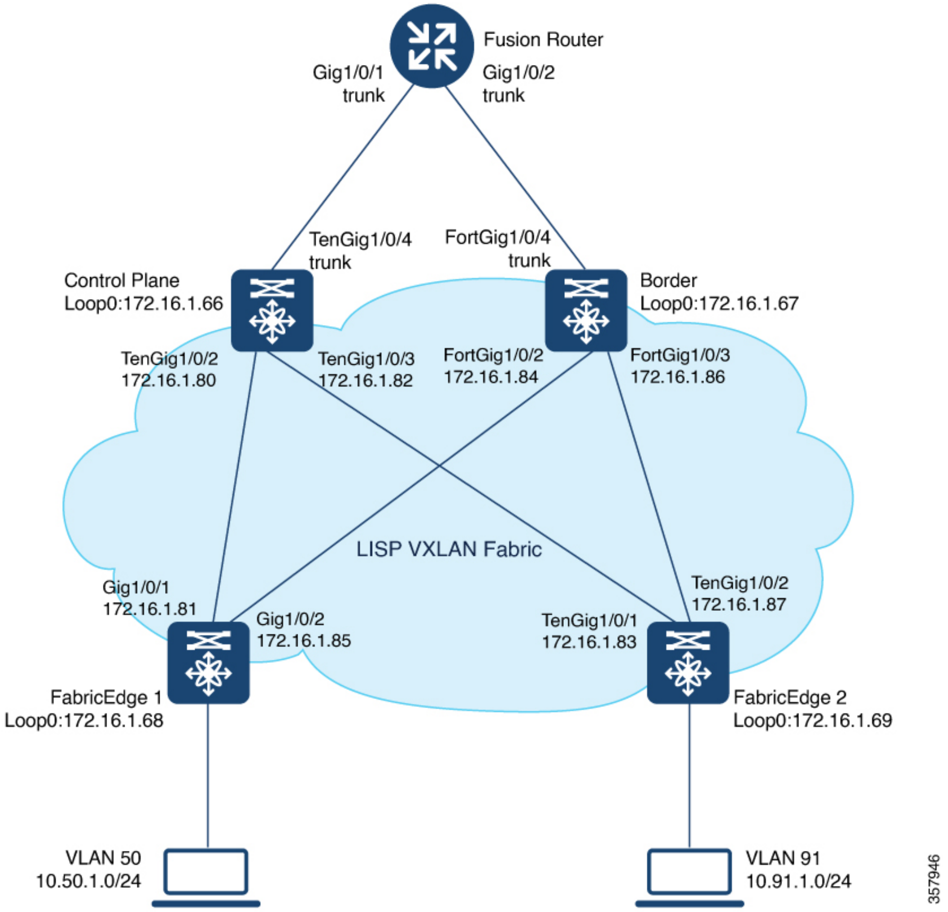
Step 23	**end**
	Example:
	`Device(config)# end`
		Returns to privileged EXEC mode.

Configuration Example for a Control Plane Node

This example shows a sample configuration for a control plane node in a LISP VXLAN-based fabric with two border nodes, two control plane nodes, and two fabric edge nodes. VLAN50 is configured on Fabric Edge 1 and VLAN91 is configured on Fabric Edge 2.

This example only shows the configuration of a control plane node. It does not show any other prior configuration such as that of an underlay.

Figure 3: LISP VXLAN Fabric Topology



CP

```
router lisp
 locator-table default
 service ipv4
  encapsulation vxlan
  sgt
  map-server
  map-resolver
  exit-service-ipv4
 !
 service ipv6
  encapsulation vxlan
  sgt
  map-server
  map-resolver
  exit-service-ipv6
 !
 service ethernet
  map-server
  map-resolver
  exit-service-ethernet
 !
 !
 site site_uci
  description map-server
  authentication-key some-key
  eid-record instance-id 4097 0.0.0.0/0 accept-more-specifics     //to import routes from
external network
  eid-record instance-id 4097 10.91.1.0/24 accept-more-specifics  //10.91.1.0/24 is a fabric
 prefix
  eid-record instance-id 4099 0.0.0.0/0 accept-more-specifics     //to import routes from
external network
  eid-record instance-id 4099 10.50.1.0/24 accept-more-specifics  //10.50.1.0/24 is fabric
 prefix
  eid-record instance-id 4099 ::/0 accept-more-specifics          //to import routes from
external network
  eid-record instance-id 4099 2001:DB8:2050::/64 accept-more-specifics  //fabric prefix
  eid-record instance-id 8194 any-mac
  eid-record instance-id 8197 any-mac
  allow-locator-default-etr instance-id 4097 ipv4
  allow-locator-default-etr instance-id 4099 ipv4
  allow-locator-default-etr instance-id 4099 ipv6
  exit-site
 !
 ipv4 source-locator Loopback0
 exit-router-lisp
!
```

Note Configure the 0.0.0.0/0 and ::/0 EID prefixes if you have to import routes from external network into the LISP database. A typical case would be if your fabric is connected to a Data Center. The Data Center pushes EID prefixes that are not known in the LISP database and that are imported into the fabric through BGP.

CHAPTER 4

Configuring Border Node

A LISP VXLAN fabric border node serves as a gateway between the fabric site and the sites external to the fabric. Traffic entering or leaving the fabric is encapsulated or decapsulated (respectively) by the border node.

The following devices can be configured as border nodes:

- Cisco Catalyst 9300 Series Switches
- Cisco Catalyst 9400 Series Switches
- Cisco Catalyst 9500 Series Switches
- Cisco Catalyst 9600 Series Switches

A fabric border node can be configured as an internal border node, or an external border node, or both internal and external border node.

An **internal border node** is used when traffic originating from within the fabric should follow a non-default route to reach an external destination. The Internal Border Node advertises endpoint reachability to the external network and imports external non-default routes into the fabric control plane.

An **external border node** is a default gateway for a Fabric Site. It is used as a gateway for traffic originating from within the fabric that is following a default route, such as traffic destined for the internet. It advertises endpoint reachability to the external network but does not import any external routes into the fabric control plane.

An **internal and external border node** both imports non-default routes into the fabric control plane and functions a default gateway for a fabric site. It advertises endpoint reachability to the external network and imports external non-default routes into the fabric.

Note In a border node configuration, each LISP instance-id should be associated with a routing table (global routing table or the VRF). A default border should have default routes configured in the routing table for each VRF, to dynamically register with the control plane node as a default border.

Functions of a Border Node

A fabric border node performs the following functions in the fabric:

- **Advertise EID subnets**: A border node exports the endpoint prefix space as an aggregate to the external networks, using the Border Gateway Protocol (BGP). This helps to direct the traffic from outside of the fabric destined for endpoints within the fabric.
- **Gateway between the Fabric and an external network**: A border node is an egress point for traffic to all those destinations that are outside the fabric.

  An external border acts like a default gateway. It handles the traffic destined to locations that are not known to the control plane. Internal border advertises external destinations into the fabric and should be used for traffic to known destinations outside the fabric.
- **Network virtualization extension to the external world**: A border node can extend network virtualization from inside the fabric to outside the fabric by using VRF-lite and VRF-aware routing protocols to preserve the segmentation.
- **Policy mapping**: A border node maps the SGT information from within the fabric to be appropriately maintained when the traffic exits that fabric. When a fabric packet is decapsulated at the border node, the SGT information can be directly mapped into the Cisco metadata field of packet, using inline tagging.
- **VXLAN encapsulation/decapsulation**: A border node encapsulates the packets received from external network, which are destined to the endpoints within the fabric. It decapsulates the packets that are sourced from the fabric endpoints and destined to locations outside the fabric.

How to Configure an External Border Node

Note Before you begin, ensure that routed access design is used to configure the underlay network.

Step	Task	Purpose
Step 1	Configure VRF	Configure a VRF to support IPv4 and IPv6 address routing tables. VRF maintains the routing and forwarding information for devices within a virtual network. A VRF instance has its own IP routing table, a forwarding table, and one or more interfaces assigned to it. The VRF tables help the routing device reach the locator address space.
Step 2	Configure Layer 3 Handoff SVI	Configure the SVI for Layer 3 handoff.

Step	Task	Purpose
Step 3	Configure the Interface that Connects to an Upstream Router	Configure a VLAN trunk port interface to connect to an upstream router. An upstream router is located external to the fabric and provides inter-VRF forwarding that is necessary for communication between the virtual networks (segments). It also provides access to shared services for the endpoints in the fabric.
Step 4	Configure Loopback for Overlay Segment in User-Defined VRF	• Configure a loopback interface for a overlay segment. This loopback is used to advertise the overlay subnet prefixes to the external network. • Configure a loopback interface for the default instance in LISP (Global Routing Table). The default instance is used to connect the network infrastructure elements like Access Points and Layer 2 switches to the fabric access layer.
Step 5	Configure LISP	• Set up the Proxy Ingress Tunnel Router (PITR) functionality for both IPv4 and IPv6 address families. A PITR encapsulates and forwards the incoming packets to provide non-LISP-to-LISP interworking. • Set up the Proxy Egress Tunnel Router (PETR) functionality for both IPv4 and IPv6 address families. A PETR decapsulates the LISP VXLAN encapsulated packets to the provide LISP-to-non-LISP interworking. • Define this border node as a default ETR and map the default route for each VRF.
Step 6	Configure Layer 3 Instance ID: • Create Layer 3 Instance ID for Default Instance • Create Layer 3 Instance ID for User-Defined VRF - External Border	• Configure a Layer 3 instance ID for the default instance. • Configure Layer 3 instance IDs for the VRFs that you define.
Step 7	Configure a BGP Routing Process	Configure Border Gateway Protocol (BGP) for route exchange with the external network.

Step	Task	Purpose
Step 8	(Optional) Redistribute Routing Information through External Border, on page 51	If your deployment has a scenario where the fabric site has an internal border that accepts prefixes to be routed to an external network through an external border, perform this step. This step redistributes LISP routes to BGP through an external border.

Step	Task	Purpose
Step 9	Verify the configurations on the border node using these **show** commands:	
	show lisp session	Displays the details of the LISP sessions that are established on the border node.
	show lisp locator-set	Displays the locator set information.
	show ip interface brief	Displays the usability status of all the interfaces that are configured on the device. Filter the output to view the dynamically created LISP interfaces, using the **show ip interface brief | i LISP** command.
	show lisp instance-id * ipv4 **show lisp instance-id * ipv6**	Displays the details of each of the LISP IPv4 or IPv6 instances that are configured on the border node. Use this command to view the operational status of the IPv4 or the IPv6 address family under each instance-id. This includes the status of the database, map-cache, publication entries, site registration entries, and so on.
	show ip route vrf *vrf*	Displays the route table that is created on the border node for a given VRF.
	show lisp service ipv4 summary **show lisp service ipv6 summary**	Displays a summary of the LISP IPv4 or IPv6 services on the border node. Use this command to check the number of EID tables and database entries, the total number of map-cache entries, and information about each VRF.
	show lisp service ipv4 statistics **show lisp service ipv6 statistics**	Displays the LISP IPv4 or IPv6 packet statistics for all EID prefixes. Use this command to check the total number of packet encapsulations, decapsulations, map requests, map replies, map registers, and other LISP-related packet information, for the IPv4 or IPv6 service.
	show lisp service ipv4 forwarding eid remote detail **show lisp service ipv6 forwarding eid remote detail**	Displays the forwarding information for the destination EID prefixes. Use this command to view the EID prefix, associated locator status bits, and total encapsulated packets and bytes for each destination EID-prefix.
	show lisp platform	

Step	Task	Purpose
		Displays the limits of the given platform or the device. This command shows the LISP instance limits, Layer 3 limits, Layer 2 limits, and the supported configuration style on the device. Use this command to understand the limits of the device and plan its usage and role in the fabric.

To see a sample configuration for an external border node, go to Configuration Example for an External Border Node.

To see the sample outputs of show commands on the border node, go to Verify Distributed Border and Control Plane Node, on page 59.

How to Configure an Internal Border Node

Note Before you begin, ensure that routed access design is used to configure the underlay network.

Step	Task	Purpose
Step 1	Configure VRF	Configure a VRF to support IPv4 and IPv6 address routing tables. VRF maintains the routing and forwarding information for devices within a virtual network. A VRF instance has its own IP routing table, a forwarding table, and one or more interfaces assigned to it. The VRF tables help the routing device reach the locator address space.
Step 2	Configure Layer 3 Handoff SVI	Configure the SVI for Layer 3 handoff.
Step 3	Configure the Interface that Connects to an Upstream Router	Configure a VLAN trunk port interface to connect to an upstream router. An upstream router is located external to the fabric and provides inter-VRF forwarding that is necessary for communication between the virtual networks (segments). It also provides access to shared services for the endpoints in the fabric.

Step	Task	Purpose
Step 4	Configure Loopback for Overlay Segment in User-Defined VRF	• Configure a loopback interface for a overlay segment. This loopback is used to advertise the overlay subnet prefixes to the external network. • Configure a loopback interface for the default instance in LISP (Global Routing Table). The default instance is used to connect the network infrastructure elements like Access Points and Layer 2 switches to the fabric access layer.
Step 5	Configure LISP	• Set up the Proxy Ingress Tunnel Router (PITR) functionality for both IPv4 and IPv6 address families. A PITR encapsulates and forwards the incoming packets to provide non-LISP-to-LISP interworking. • Set up the Proxy Egress Tunnel Router (PETR) functionality for both IPv4 and IPv6 address families. A PETR decapsulates the LISP VXLAN encapsulated packets to the provide LISP-to-non-LISP interworking. • Set up the route-import functionality to import external routes into each VRF that is configured.
Step 6	Configure Layer 3 Instance ID: • Create Layer 3 Instance ID for Default Instance • Create Layer 3 Instance ID for User-Defined VRF - Internal Border	• Configure a Layer 3 instance ID for the default instance. • Configure Layer 3 instance IDs for the VRFs that you define. Use the **route-import database** command to register the imported routes to the control plane. The routes that are learnt are filtered according to the **route-map** option specified, to prevent routing loops.
Step 7	Configure a BGP Routing Process	Configure Border Gateway Protocol (BGP) for route exchange with the external network.
Step 8	Configure Prefix-List and Route-Map	Define route maps with prefix lists to filter the routes that are imported into the fabric.

Step	Task	Purpose
Step 9	Verify the configurations on the border node using these **show** commands:	
	show lisp session	Displays the details of the LISP sessions that are established on the border node.
	show lisp locator-set	Displays the locator set information.
	show ip interface brief	Displays the usability status of all the interfaces that are configured on the device. Filter the output to view the dynamically created LISP interfaces, using the **show ip interface brief | i LISP** command.
	show lisp instance-id * ipv4 **show lisp instance-id * ipv6**	Displays the details of each of the LISP IPv4 or IPv6 instances that are configured on the border node. Use this command to view the operational status of the IPv4 address family under each instance-id. This includes the status of IPv4 database, map-cache, publication entries, site registration entries, and so on.
	show ip route vrf *vrf*	Displays the route table that is created on the border node for a given VRF.
	show lisp service ipv4 summary **show lisp service ipv6 summary**	Displays a summary of the LISP IPv4 or IPv6 services on the border node. Use this command to check the number of EID tables and database entries, the total number of map-cache entries, and information about each VRF.
	show lisp service ipv4 statistics **show lisp service ipv6 statistics**	Displays the LISP IPv4 or IPv6 packet statistics for all EID prefixes. Use this command to check the total number of packet encapsulations, decapsulations, map requests, map replies, map registers, and other LISP-related packet information, for the IPv4 or IPv6 service.
	show lisp service ipv4 forwarding eid remote detail **show lisp service ipv6 forwarding eid remote detail**	Displays the forwarding information for the remote or destination EID prefixes. Use this command to view the EID prefix, associated locator status bits, and total encapsulated packets and bytes for each remote EID-prefix.
	show lisp platform	

Step	Task	Purpose
		Displays the limits of the given platform or the device. This command shows the LISP instance limits, Layer 3 limits, Layer 2 limits, and the supported configuration style on the device. Use this command to understand the limits of the device and plan its usage and role in the fabric.

To see a sample configuration for an internal border node, go to Configuration Example for an Internal Border Node

To see a sample configuration for an internal and external border node, go to Configuration Example for an Internal and External Border

Detailed Steps to Configure a Border Node

This section describes the tasks involved in configuring an internal border, an external border, and an anywhere border which is both internal and external.

Configure VRF

To configure VRFs on a border node, perform this task:

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	**vrf definition** *vrf-name* **Example:** `Device(config)# vrf definition VN3`	Configures a VRF table, and enters VRF configuration mode.
Step 4	**rd** *route-distinguisher* **Example:** `Device(config-vrf)# rd 1:4099`	Creates routing and forwarding tables for a VRF instance.
Step 5	**address-family ipv4** **Example:**	Specifies the address family, and enters address family configuration mode.

	Command or Action	Purpose
	`Device(config-vrf)# **address-family ipv4**`	
Step 6	**route-target export** *route-target-ext-community* **Example:** `Device(config-vrf-af)# route-target export 1:4099`	Creates a list of export route target communities for the specified VRF. Enter either an AS system number and an arbitrary number (xxx:y) or an IP address and an arbitrary number (A.B.C.D:y). The *route-target-ext-community* value should be the same as the *route-distinguisher* value entered in the earlier step.
Step 7	**route-target import** *route-target-ext-community* **Example:** `Device(config-vrf-af)# route-target import 1:4099`	Creates a list of import route target communities for the specified VRF.
Step 8	**exit-address-family** **Example:** `Device(config-vrf-af)# exit-address-family`	Exits address family configuration mode, and enters VRF configuration mode.
Step 9	**address-family ipv6** **Example:** `Device(config-vrf)# address-family ipv6`	Specifies the address family, and enters address family configuration mode.
Step 10	**route-target export** *route-target-ext-community* **Example:** `Device(config-vrf-af)# route-target export 1:4099`	Creates a list of export route target communities for the specified VRF. Enter either an AS system number and an arbitrary number (xxx:y) or an IP address and an arbitrary number (A.B.C.D:y). The *route-target-ext-community* value should be the same as the *route-distinguisher* value entered in the earlier step.
Step 11	**route-target import** *route-target-ext-community* **Example:** `Device(config-vrf-af)# route-target import 1:4099`	Creates a list of import route target communities for the specified VRF.
Step 12	**exit-address-family** **Example:** `Device(config-vrf-af)# exit-address-family`	Exits address family configuration mode, and enters VRF configuration mode.

	Command or Action	Purpose

Step 13	**end**
	Example:
	`Device(config-vrf)# end`
		Returns to privileged EXEC mode.

Configure Layer 3 Handoff SVI

To configure Layer 3 handoff SVI on a border node, perform this task:

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	**vlan** *vlan-id* **Example:** `Device(config)# vlan 222`	Specifies a VLAN ID, and enters VLAN configuration mode.
Step 4	**name** *vlan-name* **Example:** `Device(config-vlan)# name 222`	Specifies a name for the VLAN.
Step 5	**exit** **Example:** `Device(config-vlan)# exit`	Exits VLAN configuration mode, and enters global configuration mode.
Step 6	**interface** *vlan-id* **Example:** `Device(config)# interface Vlan222`	Specifies the interface for which you are adding a description, and enters interface configuration mode.
Step 7	**description** *string* **Example:** `Device(config-if)# description vrf-external`	Adds a description for the interface.
Step 8	**vrf forwarding** *name* **Example:** `Device(config-if)# vrf forwarding VN3`	Associates the VRF instance with the interface.

	Command or Action	Purpose
Step 9	**ip address** *ip_address subnet_mask* **Example:** `Device(config-if)# ip address 10.20.1.1 255.255.255.252`	Configures the IP address and IP subnet.
Step 10	**no ip redirects** **Example:** `Device(config-if)# no ip redirects`	Disables sending of Internet Control Message Protocol (ICMP) redirect messages.
Step 11	**ipv6 address** *address* **Example:** `Device(config-if)# ipv6 address 2001:DB8:20::1/126`	Configures an IPv6 address on the interface.
Step 12	**ipv6 enable** **Example:** `Device(config-if)# ipv6 enable`	Enables IPv6 on the interface.

Step 13	**end**
	Example:
	`Device(config-if)# end`
		Returns to privileged EXEC mode.

Configure the Interface that Connects to an Upstream Router

To configure the interface that connects to an upstream router, perform this task:

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	**interface** *interface-number* **Example:** `Device(config)# interface FortyGigabitEthernet1/0/4`	Creates an interface to connect to an upstream router, and enters interface configuration mode.
Step 4	**switchport mode trunk** **Example:**	Configures the interface as a VLAN trunk port.

	Command or Action	Purpose
	`Device(config-if)# switchport mode trunk`	

Step 5	**end**
	Example:
	`Device(config-if)# end`
		Returns to privileged EXEC mode.

Configure Loopback for Overlay Segment in User-Defined VRF

To configure loopback for the overlay segment in user-defined VRF on a border node, perform this task:

Note This loopback is used to advertise the overlay subnet prefixes to the external network.

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	**interface loopback 50** **Example:** `Device(config)# interface loopback 50`	Creates a loopback interface for the overlay segment, and enters interface configuration mode.
Step 4	**description** *name* **Example:** `Device(config-if)# description Loopback Border`	Adds a description for an interface.
Step 5	**vrf forwarding** *vrf-name* **Example:** `Device(config-if)# vrf forwarding VN3`	Associates the VRF with the Layer 3 interface.
Step 6	**ip address** *address mask* **Example:** `Device(config-if)# ip address 10.50.1.1 255.255.255.255`	Assigns an IP address to the interface. Ensure that this is the IP address of the SVI for the user-defined VRF.

	Command or Action	Purpose
Step 7	**ipv6 address** *address* **Example:** `Device(config-if)# ipv6 address 2001:DB8:2050::1/128`	Assigns an IPv6 address to the interface.
Step 8	**ipv6 enable** **Example:** `Device(config-if)# ipv6 enable`	Enables IPv6 on the interface.

Step 9	**end**
	Example:
	`Device(config-if)# end`
		Returns to privileged EXEC mode.

Configure Loopback for Overlay Segment in the Default Instance of LISP (Global Routing Table)

To configure the overlay segment in the default instance of LISP, perform this task:

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	**interface loopback 91** **Example:** `Device(config)# interface loopback 91`	Creates a loopback interface for the default instance, and enters interface configuration mode.
Step 4	**ip address** *address mask* **Example:** `Device(config-if)# ip address 10.91.1.1 255.255.255.255`	Assigns an IP address to the interface. Ensure that this is the IP address of the SVI for the default instance.

Step 5	**end**
	Example:
	`Device(config-if)# end`
		Returns to privileged EXEC mode.

Configure LISP

To configure LISP on a border node, perform this task:

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	**router lisp** **Example:** `Device(config)# router lisp`	Enters LISP configuration mode.
Step 4	**locator-table default** **Example:** `Device(config-router-lisp)# locator-table default`	Selects the default (global) routing table for association with the routing locator address space.
Step 5	**locator-set** *loc-set-name* **Example:** `Device(config-router-lisp)# locator-set default_etr_locator`	Specifies a locator-set, and enters the locator-set configuration mode. A locator-set identifies the routing-locator that LISP uses when it registers the local endpoints. In this step, configure a default locator set.
Step 6	**ipv4-interface Loopback** *loopback-interface-id* **priority** *locator-priority* **weight** *locator-weight* **Example:** `Device(config-router-lisp-locator-set)# ipv4-interface Loopback0 priority 10 weight 10`	Specifies that the IPv4 address of the loopback interface should be used to reach the locator. Priority and weight values are associated with the locator address to define traffic policies when multiple RLOCs are defined for the same EID-prefix block. A locator with a lower priority value takes preference. When multiple locators have the same priority, they can be used in a load-sharing manner. Weight is a value 0–100 and represents the percentage of traffic to be load-shared to that locator.
Step 7	**exit-locator-set** **Example:** `Device(config-router-lisp-locator-set)# exit-locator-set`	Exits locator-set configuration mode, and enters LISP configuration mode.

	Command or Action	Purpose
Step 8	**locator-set** *loc-set-name* **Example:** `Device(config-router-lisp)# locator-set eid_locator`	Specifies a locator-set, and enters the locator-set configuration mode. Ensure that this locator set is different from the default locator that was created in Step 5.
Step 9	**ipv4-interface Loopback** *loopback-interface-id* **priority** *locator-priority* **weight** *locator-weight* **Example:** `Device(config-router-lisp-locator-set)# IPv4-interface Loopback0 priority 10 weight 10`	Specifies that the IPv4 address of the loopback interface should be used to reach the locator. Priority and weight values are associated with the locator address to define traffic policies when multiple RLOCs are defined for the same EID-prefix block. A locator with a lower priority value takes preference. When multiple locators have the same priority, they can be used in a load-sharing manner. Weight is a value 0–100 and represents the percentage of traffic to be load-shared to that locator.
Step 10	**auto-discover-rlocs** **Example:** `Device(config-router-lisp-locator-set)# auto-discover-rlocs`	Auto discover the locators registered by other ingress or egress tunnel routers (xTRs).
Step 11	**exit-locator-set** **Example:** `Device(config-router-lisp-locator-set)# exit-locator-set`	Exits locator-set configuration mode, and enters LISP configuration mode.
Step 12	**locator default-set** *loc-set-name* **Example:** `Device(config-router-lisp)# locator default-set eid_locator`	Specifies a default locator-set.

Step 13	**service** { **ipv4** | **ipv6** }
	Example:
	`Device(config-router-lisp)# service ipv4`
		Enables network services on the default instance.
		service ipv4: Enables Layer 3 network services for the IPv4 address family.
		service ipv6: Enables Layer 3 network services for the IPv6 address family.

Step 14	**encapsulation vxlan**
	Example:
	`Device(config-router-lisp-serv-ipv4)# encapsulation vxlan`
	`Device(config-router-lisp-serv-ipv6)# encapsulation vxlan`
		Specifies VXLAN-based encapsulation.

	Command or Action	Purpose

Step 15	**map-cache publications**
	Example:
	`Device(config-router-lisp-serv-ipv4)# map-cache publications`
	`Device(config-router-lisp-serv-ipv6)# map-cache publications`
		Exports the publication entries to the map cache. These entries are used for forwarding the traffic.

Step 16	**import publication publisher** *publisher-address*
	Example:
	`Device(config-router-lisp-serv-ipv4)# import publication publisher 172.16.1.66`
	`Device(config-router-lisp-serv-ipv6)# import publication publisher 172.16.1.66`
		Imports the publications from the publisher that is specified by the *publisher-address*. *publisher-address* is the IP address of the Loopback 0 interface of the control plane node.
		If your fabric site has more than one control plane node, there are as many publishers. Execute this command for each of those *publisher-address* (control plane nodes). Imported publications are stored in a publication table.

Step 17	**itr map-resolver** *map-resolver-address*
	Example:
	`Device(config-router-lisp-serv-ipv4)# itr map-resolver 172.16.1.66`
	`Device(config-router-lisp-serv-ipv6)# itr map-resolver 172.16.1.66`
		Configures a locator address for the LISP map resolver to which this router sends map request messages for EID-to-RLOC mapping resolutions.
		A control plane node is the LISP map resolver. *map-resolver-address* is the IP address of the Loopback 0 interface of the control plane node. If your fabric site has more than one control plane node, execute this command for each of the *map-resolver-address* (control plane nodes). Execute this command even if the border and control plane nodes are located on the same device.

Step 18	**etr map-server** *map-server-address* **key** *authentication-key*
	Example:
	`Device(config-router-lisp-serv-ipv4)# etr map-server 172.16.1.66 key some-key`
	`Device(config-router-lisp-serv-ipv6)# etr map-server 172.16.1.66 key some-key`
		Configures a map server to be used by the Egress Tunnel Router (ETR) for endpoint registrations, and specifies the authentication key to be used with this map server.
		A control plane node is the LISP map server. *map-server-address* is the IP address of the Loopback 0 interface of the control plane node. If your fabric site has more than one control plane node, execute this command for each of the *map-server-address* (control plane nodes). Execute this command even if the border and control plane nodes are located on the same device.

	Command or Action	Purpose
		Note Ensure that you use the same *authentication-key* that was configured on the control plane node.

Step 19	**etr map-server** *map-server-address* **proxy-reply**
	Example:
	`Device(config-router-lisp-serv-ipv4)# etr map-server 172.16.1.66 proxy-reply`
	`Device(config-router-lisp-serv-ipv6)# etr map-server 172.16.1.66 proxy-reply`
		Configures the map server to send map replies on behalf of the ETR.
		map-server-address is the IP address of the Loopback 0 interface of control plane node. If your fabric site has more than one control plane node, execute this command for each of the *map-server-address* (control plane nodes). Execute this command even if the border and control plane nodes are located on the same device.

Step 20	**etr**
	Example:
	`Device(config-router-lisp-serv-ipv4)# etr`
	`Device(config-router-lisp-serv-ipv6)# etr`
		Configures the device as an Egress Tunnel Router (ETR).

Step 21	**sgt**
	Example:
	`Device(config-router-lisp-serv-ipv4)# sgt`
	`Device(config-router-lisp-serv-ipv6)# sgt`
		(Optional) Enables the Security Group Tag (SGT) function for SGT tag propagation. Configure this command only if you need SGT propagation in your fabric network.

Step 22	**route-export publications**
	Example:
	`Device(config-router-lisp-serv-ipv4)# route-export publications`
	`Device(config-router-lisp-serv-ipv6)# route-export publications`
		Exports the LISP publications into the routing information base (RIB).

Step 23	**distance publications** *distance*
	Example:
	`Device(config-router-lisp-serv-ipv4)# distance publications 250`
	`Device(config-router-lisp-serv-ipv6)# distance publications 250`
		Specifies the administrative distance to RIB when the LISP publications are exported to the RIB.

Step 24	**proxy-etr** **Example:**	Enables Proxy Egress Tunnel Router (PETR) functionality for IPv4 EIDs.

	Command or Action	Purpose

	`Device(config-router-lisp-serv-ipv4)# proxy-etr`
	`Device(config-router-lisp-serv-ipv6)# proxy-etr`	

Step 25	**proxy-itr** *address*
	Example:
	`Device(config-router-lisp-serv-ipv4)# proxy-itr 172.16.1.67`
	`Device(config-router-lisp-serv-ipv6)# proxy-itr 172.16.1.67`
		Enables Proxy Ingress Tunnel Router (PITR) functionality for IPv4 or IPv6 EIDs.
		For *address*, specify the IP address of the Loopback 0 interface on the device.

Step 26	Do one of the following:
	- **exit-service-ipv4**
	- **exit-service-ipv6**
	Example:
	`Device(config-router-lisp-serv-ipv4)# exit-service-ipv4`
	`Device(config-router-lisp-serv-ipv4)# exit-service-ipv6`
		Exits service configuration mode, and enters LISP configuration mode.
		Use the appropriate command, depending on which service mode you are exiting from (IPv4 or IPv6 mode).

Step 27	**ipv4 locator reachability minimum-mask-length** *length* **Example:** `Device(config-router-lisp)# ipv4 locator reachability minimum-mask-length 32`	Specifies the shortest mask prefix to accept when looking up a remote RLOC in the RIB. LISP checks the host reachability from the routing locator.
Step 28	**ipv4 source-locator** *interface-number* **Example:** `Device(config-router-lisp)# ipv4 source-locator loopback0`	Configures the source locator for the outbound LISP packets. Set the loopback interface as the source locator.
Step 29	**exit-router-lisp** **Example:** `Device(config-router-lisp)# exit-router-lisp`	Exits LISP configuration mode, and enters global configuration mode.

Step 30	**end**
	Example:
	`Device(config)# end`
		Returns to privileged EXEC mode.

Step 31	**show lisp locator-set**
	Example:
	`Device# show lisp locator-set`
	`LISP Locator-set information:`
	
	`172.16.1.67, local, reachable, loopback`
	`Device#`
		Displays the LISP Locator Set information configured on the device.

Create Layer 3 Instance ID for Default Instance

To create a Layer 3 instance ID for default instance on a border node, perform this task:

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	**router lisp** **Example:** `Device(config)# router lisp`	Enters LISP configuration mode.
Step 4	**instance-id** *id* **Example:** `Device(config-router-lisp)# instance-id 4097`	Specifies an instance ID. In this step, configure the Layer 3 default instance ID. The *id* of the instance can range from 1 to 16777200.
Step 5	**remote-rloc-probe on-route-change** **Example:** `Device(config-router-lisp-inst)# remote-rloc-probe on-route-change`	Configures parameters for probing of remote routing locators (RLOCs).
Step 6	**service ipv4** **Example:** `Device(config-router-lisp-inst)# service ipv4`	Enables Layer 3 network services for the IPv4 address family.
Step 7	**eid-table default** **Example:** `Device(config-router-lisp-inst-serv-ipv4)# eid-table default`	Configures the default (global) routing table for association with the configured instance-service.
Step 8	**map-cache** *address* **map-request** **Example:** `Device(config-router-lisp-inst-serv-ipv4)# map-cache 10.91.1.0/24 map-request`	Specifies the destination EID for which map-requests are sent.
Step 9	**exit-service-ipv4** **Example:**	Exits IPv4 service configuration mode, and enters LISP instance configuration mode.

	Command or Action	Purpose
	`Device(config-router-lisp-inst-serv-ipv4)# exit-service-ipv4`	
Step 10	**exit-instance-id** **Example:** `Device(config-router-lisp-inst)# exit-instance-id`	Exits instance configuration mode, and enters LISP configuration mode.
Step 11	**exit-router-lisp** **Example:** `Device(config-router-lisp)# exit-router-lisp`	Exits LISP configuration mode, and enters global configuration mode.

Step 12	**end**
	Example:
	`Device(config)# end`
		Returns to privileged EXEC mode.

Create Layer 3 Instance ID for User-Defined VRF - External Border

To create a Layer 3 instance ID for the user-defined VRF on the external border node, perform this task:

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	**router lisp** **Example:** `Device(config)# router lisp`	Enters LISP configuration mode.
Step 4	**instance-id** *id* **Example:** `Device(config-router-lisp)# instance-id 4099`	In this step, specify the instance ID for a user-defined VRF. The *id* of the instance can range from 1 to 16777200.
Step 5	**remote-rloc-probe on-route-change** **Example:** `Device(config-router-lisp-inst)# remote-rloc-probe on-route-change`	Configures parameters for probing of remote routing locators (RLOCs).

	Command or Action	Purpose

Step 6	**service** {**ipv4** | **ipv6**}
	Example:
	`Device(config-router-lisp-inst)#` **`service ipv4`**
	`Device(config-router-lisp-inst)#` **`service ipv6`**
		Enables Layer 3 network services for the IPv4 or IPv6 address family.

Step 7	**eid-table vrf** *vrf-name*
	Example:
	`Device(config-router-lisp-inst-serv-ipv4)#` **`eid-table vrf VN3`**
	`Device(config-router-lisp-inst-serv-ipv6)#` **`eid-table vrf VN3`**
		Configures the VRF table for association with the configured instance-service.

Step 8	**database-mapping** *eid-prefix/prefix-length* **locator-set** *RLOC_name* **default-etr local**
	Example:
	`Device(config-router-lisp-inst-serv-ipv4)#`
	`database-mapping 0.0.0.0/0 locator-set default_etr_locator default-etr local`
	`Device(config-router-lisp-inst-serv-ipv6)#`
	`database-mapping ::/0 locator-set default_etr_locator default-etr local`
		Configures an IPv4 or IPv6 default ETR for a default route

Step 9	Do one of the following:**exit-service-ipv4** • **exit-service-ipv4** • **exit-service-ipv6** **Example:** `Device(config-router-lisp-inst-serv-ipv4)#` **`exit-service-ipv4`** `Device(config-router-lisp-inst-serv-ipv6)#` **`exit-service-ipv6`**	Exits service configuration mode, and enters LISP instance configuration mode. Use the appropriate command, depending on which service mode you are exiting from (IPv4 or IPv6 service mode).

Step 10	**exit-instance-id**
	Example:
	`Device(config-router-lisp-inst)#`
	`exit-instance-id`
		Exits instance configuration mode, and enters LISP configuration mode.

Step 11	**end**
	Example:
	`Device(config)#` **`end`**
		Returns to privileged EXEC mode.

Step 12	**show lisp instance-id * ipv4** **Example:** `Device#` **`show lisp instance-id * ipv4`**	Displays details of each LISP instance that has the IPv4 service enabled.

	Command or Action	Purpose

	To view only the LISP instance IDs that have IPv4 enabled, filter the output as shown:
	`Device# ` **`show lisp instance-id * ipv4
		i Instance ID`**
		`  Instance ID:`
		`      4097`
		`  Instance ID:`
		`      4099`
		`Device#`	

Create Layer 3 Instance ID for User-Defined VRF - Internal Border

An internal border imports and registers the routes advertised by an upstream router. The internal border uses the **route-import database** command to register these routes into Control Plane. The routes that are learnt are filtered according to the **route-map** option specified, to prevent routing loops.

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	**router lisp** **Example:** `Device(config)# ` **`router lisp`**	Enters LISP configuration mode.
Step 4	**instance-id** *id* **Example:** `Device(config-router-lisp)# ` **`instance-id 4099`**	In this step, specify the instance ID for a user-defined VRF. The *id* of the instance can range from 1 to 16777200.

Step 5	**remote-rloc-probe on-route-change**
	Example:
	`Device(config-router-lisp-inst)#`
	`remote-rloc-probe on-route-change`
		Configures parameters for probing of remote routing locators (RLOCs).

Step 6	**service** {**ipv4** | **ipv6**}
	Example:
	`Device(config-router-lisp-inst)# ` **`service ipv4`**
		Enables Layer 3 network services for the IPv4 or IPv6 address family.

	Command or Action	Purpose
	`Device(config-router-lisp-inst)# service ipv6`	

Step 7	**eid-table vrf** *vrf-name*
	Example:
	`Device(config-router-lisp-inst-serv-ipv4)# eid-table vrf VN3`
	`Device(config-router-lisp-inst-serv-ipv6)# eid-table vrf VN3`
		Configures the VRF table for association with the configured instance-service.

Step 8	**map-cache** *address* **map-request**
	Example:
	`Device(config-router-lisp-inst-serv-ipv4)# map-cache 0.0.0.0/0 map-request`
	`Device(config-router-lisp-inst-serv-ipv6)# map-cache ::/0 map-request`
		Specifies the destination EID to which map-requests are sent.

Step 9	**route-import database** *protocol autonomous-system-number* [**route-map** *map-name* **locator-set** *locator-set-name*]
	Example:
	`Device(config-router-lisp-inst-serv-ipv4)#`
	`route-import database bgp 600 route-map MATCH_DC_ROUTE locator-set eid_locator`
	`Device(config-router-lisp-inst-serv-ipv6)#`
	`route-import database bgp 600 route-map MATCH_DC_ROUTE_V6 locator-set eid_locator`
		Configures the import of Routing Information Base (RIB) routes to define local EID prefixes and associates them with the specified locator set.
		(Optional) The **route-map** keyword specifies that imported IP prefixes should be filtered according to the specified route-map name.

Step 10	Do one of the following:**exit-service-ipv4**
	• **exit-service-ipv4**
	• **exit-service-ipv6**
	Example:
	`Device(config-router-lisp-inst-serv-ipv4)# exit-service-ipv4`
	`Device(config-router-lisp-inst-serv-ipv6)# exit-service-ipv6`
		Exits service configuration mode, and enters LISP instance configuration mode.
		Use the appropriate command, depending on which service mode you are exiting from (IPv4 or IPv6 service mode).

Step 11	**exit-instance-id** **Example:** `Device(config-router-lisp-inst)# exit-instance-id`	Exits instance configuration mode, and enters LISP configuration mode.

Configure a BGP Routing Process

To configure a BGP routing process on a border node, perform this task:

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	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 600`	Configures a BGP routing process, and enters router configuration mode for the specified routing process. • Use the *autonomous-system-number* argument to specify an integer, from 0 and 65534, that identifies the device to other BGP speakers.
Step 4	**bgp router-id** *ip-address* **Example:** `Device(config-router)# bgp router-id interface Loopback0`	(Optional) Configures a fixed 32-bit router ID as the identifier of the local device running BGP. • Use the *ip-address* argument to specify a unique router ID within the network. **Note** Configuring a router ID using the **bgp router-id** command resets all active BGP peering sessions.
Step 5	**bgp log-neighbor-changes** **Example:** `Device(config-router)# bgp log-neighbor-changes`	Enables logging of BGP neighbor status changes (up or down) and neighbor resets. • Use this command for troubleshooting network connectivity problems and measuring network stability. Unexpected neighbor resets might indicate high error rates or high packet loss in the network and should be investigated.
Step 6	**bgp graceful-restart** **Example:** `Device(config-router)# bgp graceful-restart`	Enables Nonstop Forwarding (NSF) awareness on the device. By default, NSF awareness is disabled.

	Command or Action	Purpose
Step 7	**address-family ipv4** **Example:** `Device(config-router)# address-family ipv4`	Enters address family configuration mode to configure routing sessions that use address family-specific command configurations.
Step 8	**bgp aggregate-timer** *seconds* **Example:** `Device(config-router-af)# bgp aggregate-timer 0`	Configures the interval at which the BGP routes are aggregated. A value of 0 (zero) disables timer-based aggregation and starts aggregation immediately.

Step 9	**network** *network-number* [**mask** *network-mask*] [**route-map** *route-map-name*]
	Example:
	`Device(config-router-af)# network 10.20.2.0 mask 255.255.255.252`
	`Device(config-router-af)# network 10.91.1.1 mask 255.255.255.255`
		Specifies the network to be advertised by BGP and adds it to the BGP routing table.
		• For exterior protocols, the **network** command controls which networks are advertised. Interior protocols use the **network** command to determine where to send updates.

Step 10	**aggregate-address** *address mask* **summary-only** **Example:** `Device(config-router-af)# aggregate-address 10.91.1.0 255.255.255.0 summary-only`	Creates an aggregate entry in a BGP database.
Step 11	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Device(config-router-af)# neighbor 10.20.2.2 remote-as 300`	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.
Step 12	**neighbor** *ip-address* **update-source** *interface-type* interface-number **Example:** `Device(config-router-af)# neighbor 10.20.2.2 update-source Vlan111`	Allows the BGP sessions to use any operational interface for TCP connections.
Step 13	**neighbor** *ip-address* **activate** **Example:** `Device(config-router-af)# neighbor 10.20.2.2 activate`	Enables the exchange of information with a BGP neighbor.
Step 14	**neighbor** *ip-address* **send-community**[**both**] **Example:**	Specifies that a communities attribute should be sent to a BGP neighbor.

	Command or Action	Purpose
	`Device(config-router-af)# neighbor 10.20.2.2 send-community both`	
Step 15	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits the address family configuration mode and enters router configuration mode.

Step 16	**address-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*]
	Example:
	`Device(config-router)# address-family ipv4 vrf VN3`
	`Device(config-router)# address-family ipv6 vrf VN3`
		Enters address family configuration mode to configure routing sessions that use address family-specific command configurations.
		Use the **vrf** option to specify the VRF instance with which the subsequent address family configuration commands are associated.

Step 17	**bgp aggregate-timer** *seconds* **Example:** `Device(config-router-af)# bgp aggregate-timer 0`	Configures the interval at which the BGP routes are aggregated. A value of 0 (zero) disables timer-based aggregation and starts aggregation immediately.

Step 18	**network** *network-number* [**mask** *network-mask*] [**route-map** *route-map-name*]
	Example:
	`Device(config-router-af)# network 10.20.1.0 mask 255.255.255.252`
	`Device(config-router-af)# network 10.50.1.1 mask 255.255.255.255`
	`Device(config-router-af)# network 2001:DB8:20::/126`
	`Device(config-router-af)# network 2001:DB8:2050::1/128`
		Specifies the network to be advertised by BGP and adds it to the BGP routing table.
		• For exterior protocols, the **network** command controls which networks are advertised. Interior protocols use the **network** command to determine where to send updates.

Step 19	**aggregate-address** *address mask* **summary-only**
	Example:
	`Device(config-router-af)# aggregate-address 10.50.1.0 255.255.255.0 summary-only`
	`Device(config-router-af)# aggregate-address 2001:DB8:50::/64 summary-only`
		Creates an aggregate entry in a BGP database.

Step 20	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:**	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 or IPv6 multiprotocol BGP neighbor table of the local router.

	Command or Action	Purpose

	`Device(config-router-af)# **neighbor 10.20.1.2 remote-as 300**`
	`Device(config-router-af)# **neighbor 2001:DB8:20::2 remote-as 300**`	

Step 21	**neighbor** *ip-address* **update-source** *interface-type interface-number*
	Example:
	`Device(config-router-af)# neighbor 10.20.1.2 update-source Vlan222`
	`Device(config-router-af)# neighbor 2001:DB8:20::2 update-source Vlan222`
		Allows the BGP sessions to use any operational interface for TCP connections.

Step 22	**neighbor** *ip-address* **activate**
	Example:
	`Device(config-router-af)# neighbor 10.20.1.2 activate`
	`Device(config-router-af)# neighbor 2001:DB8:20::2 activate`
		Enables the exchange of information with a BGP neighbor.

Step 23	**neighbor** *ip-address* **send-community** [**both**]
	Example:
	`Device(config-router-af)# neighbor 10.20.1.2 send-community both`
	`Device(config-router-af)# neighbor 2001:DB8:20::2 send-community both`
		Specifies that a communities attribute should be sent to a BGP neighbor.

Step 24	**neighbor** *ip-address* **weight** [*number*]
	Example:
	`Device(config-router-af)# neighbor 10.20.1.2 weight 65535`
	`Device(config-router-af)# neighbor 2001:DB8:20::2 weight 65535`
		Assigns a weight to a neighbor connection.

Step 25	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits the address family configuration mode and enters router configuration mode.
Step 26	**exit** **Example:** `Device(config-router)# exit`	Exits router configuration mode and enters global configuration mode.

Step 27	**end**
	Example:
	`Device(config-route-map)# end`
		Exits router map configuration mode and returns to privileged EXEC mode.

<table>
<tr><th></th><th>Command or Action</th><th>Purpose</th></tr>
<tr><td><b>Step 28</b></td><td>

show ip route vrf *vrf-name*

Example:

```
Device# show ip route vrf VN3

Routing Table: VN3
Codes: L - local, C - connected, S -
static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external,
 O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1,
 N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 -
 OSPF external type 2, m - OMP
       n - NAT, Ni - NAT inside, No -
NAT outside, Nd - NAT DIA
       i - IS-IS, su - IS-IS summary,
L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * -
candidate default, U - per-user static
 route
       H - NHRP, G - NHRP registered,
g - NHRP registration summary
       o - ODR, P - periodic downloaded
 static route, l - LISP
       a - application route
       + - replicated route, % - next
hop override, p - overrides from PfR
       & - replicated local route
overrides by connected

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted,
 4 subnets, 3 masks
C        10.20.1.0/30 is directly
connected, Vlan222
L        10.20.1.1/32 is directly
connected, Vlan222
B       10.50.1.0/24 [200/0], 00:32:34,
 Null0
C        10.50.1.1/32 is directly
connected, Loopback50
Device#
```

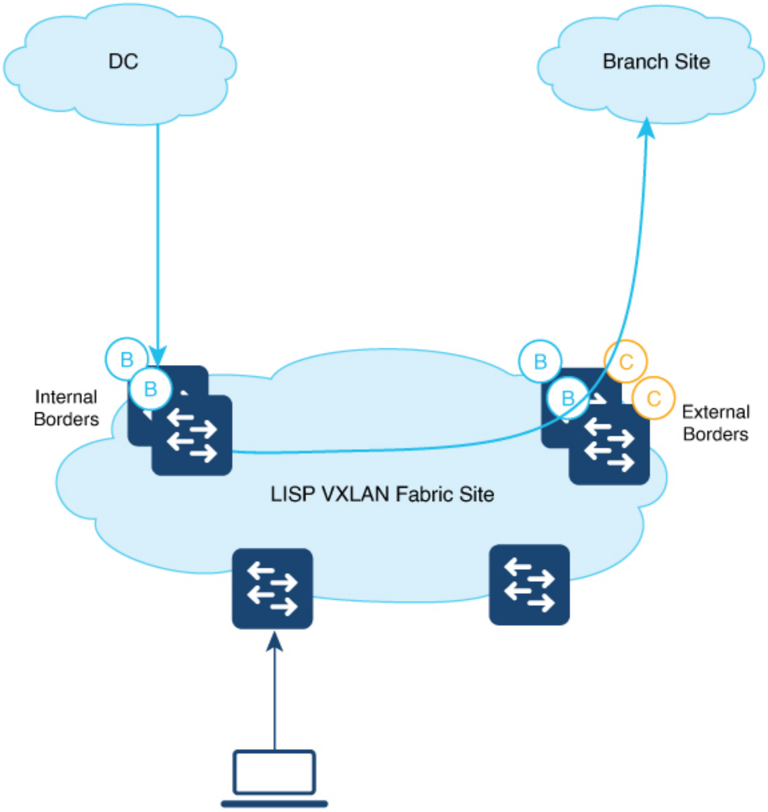
</td><td>Displays the route table on the device, for a specified VRF.</td></tr>
</table>

Redistribute Routing Information through External Border

To redistribute routing information from LISP to other routing protocols, use the **redistribute lisp** command in the address-family configuration mode.

Consider a scenario where the LISP VXLAN fabric site is connected to a Data Center (DC) through its internal border. An external border connects the fabric to a non-fabric network, a Branch Site. Traffic from the Data Center that is destined to the Branch Site can transit through the LISP VXLAN fabric site. The prefixes from the internal border are routed to the external border which redistributes the routing information into BGP.

Here is an illustration that depicts the scenario described in this section.



To redistribute routes from LISP, perform this task:

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	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 600`	Configures a BGP routing process, and enters router configuration mode for the specified routing process. • Use the *autonomous-system-number* argument to specify an integer, from 0 and 65534, that identifies the device to other BGP speakers.

	Command or Action	Purpose
Step 4	**address-family ipv4** **Example:** `Device(config-router)# address-family ipv4`	Enters address family configuration mode to configure routing sessions that use address family-specific command configurations.
Step 5	**redistribute** *protocol* **metric** *metric-value* **route-map** *map-tag* **Example:** `Device(config-router-af)# redistribute lisp metric 10 route-map LISP_TO_BGP`	Redistributes routes from one routing domain into another routing domain. Here, LISP routes are redistributed into the BGP domain. The **route-map LISP_TO_BGP** configuration filters the specific routes that are to be redistributed. Only the filtered routes are imported into the BGP domain.The LISP_TO_BGP route map is described in the following steps.
Step 6	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits the address family configuration mode and enters router configuration mode.
Step 7	**exit** **Example:** `Device(config-router)# exit`	Exits router configuration mode and enters global configuration mode.
Step 8	**route-map** *map-name* [**permit** | **deny**] [*sequence-number*] **Example:** `Device(config)# route-map LISP_TO_BGP permit 10`	Configures a route map for the BGP and enters route map configuration mode. Route map entries are read in order. You can identify the order using the *sequence_number* argument.
Step 9	**description** *description* **Example:** `Device(config-route-map)# description AS-number tag`	Adds a description for the route map.
Step 10	**set as-path tag** **Example:** `Device(config-route-map)# set as-path tag`	Modifies an autonomous system path for BGP routes.

Step 11	**end**
	Example:
	`Device(config-route-map)# end`
		Exits router map configuration mode and returns to privileged EXEC mode.

Configure Prefix-List and Route-Map

Note This procedure is applicable to an internal border node and both internal and external border node. It is not applicable to an external border node.

To configure prefix list and route map on a border node, perform this task:

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	{**ip** | **ipv6**} **prefix-list** *prefix-list-name* [**seq** *seq-value*] {**deny** *network* / *length* | **permit** *network* / *length* }
	Example:
	`Device(config)# ip prefix-list DENY_0.0.0.0 seq 10 permit 0.0.0.0/0`
	`Device(config)# ip prefix-list L3HANDOFF_PREFIXES seq 828011002 permit 10.20.1.0/30`
	`Device(config)# ipv6 prefix-list DENY_IPV6_0 seq 10 permit ::/0`
	`Device(config)# ipv6 prefix-list L3HANDOFF_PREFIXES seq 568642686 permit 2001:DB8:20::/126`
		Creates a prefix list and defines a range of IP prefixes to import into the VRF table.

Step 4	**route-map** *map-name* [**permit** | **deny**] [*sequence-number*] **Example:** `Device(config)# route-map MATCH_DC_ROUTE deny 5`	Configures a route map and enters route map configuration mode.
Step 5	**description** *description* **Example:** `Device(config-route-map)# description Deny IPV4 default route`	(Optional) Adds a description for the route map.
Step 6	**match ip address** {*access-list-number* | *access-list-name*} [*... access-list-number* | *... access-list-name*]	(Optional) Creates a match clause to permit routes that match the specified

	Command or Action	Purpose
	Example: `Device(config-route-map)#` **`match ip address prefix-list DENY_0.0.0.0`**	*access-list-number* or *access-list-name* argument.

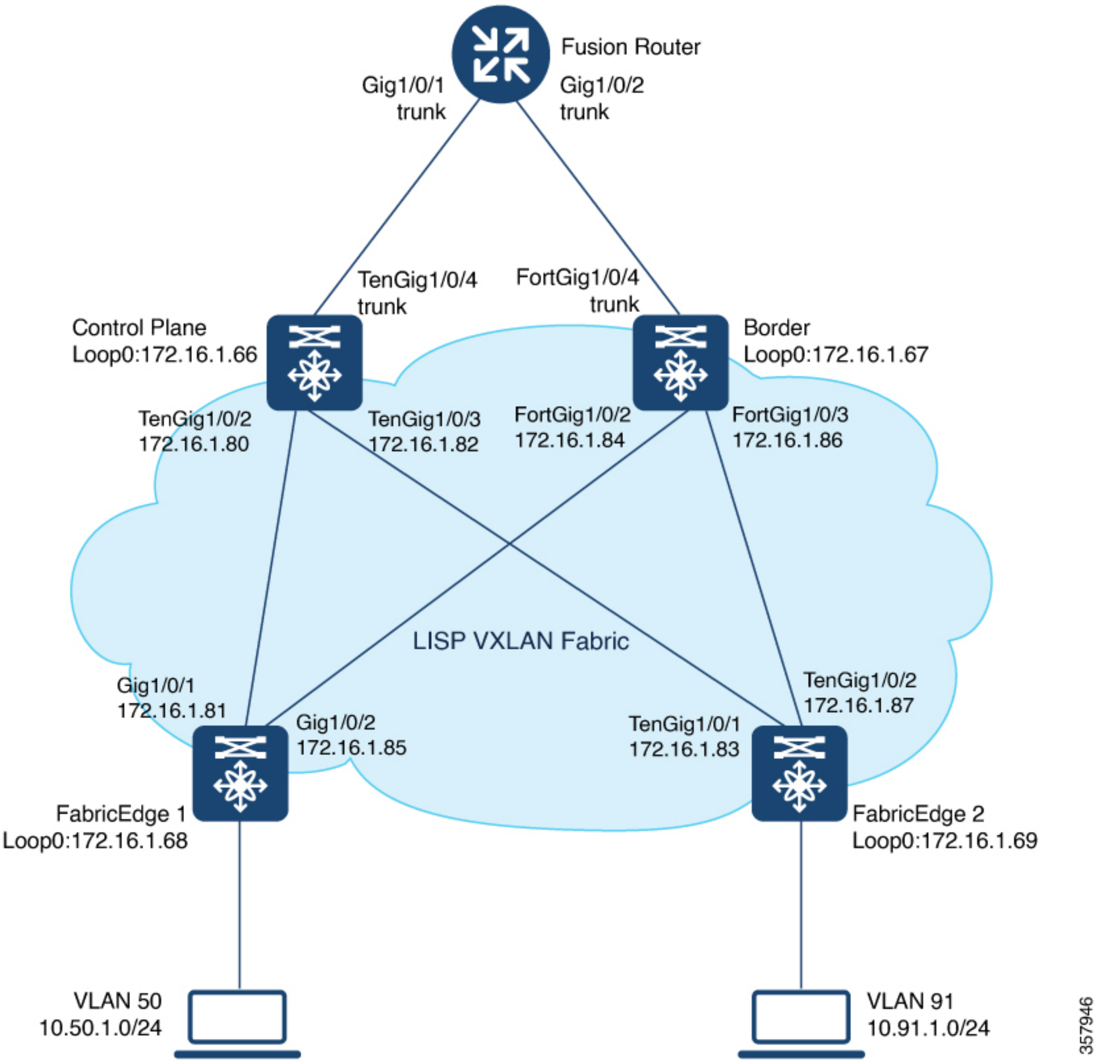
Step 7	Repeat steps 4 to 7 to configure more route maps.
	Example:
	`route-map MATCH_DC_ROUTE deny 17`
	` description Deny L3Handoff Prefixes`
	` match ip address prefix-list L3HANDOFF_PREFIXES`
	`!`
	`route-map MATCH_DC_ROUTE permit 20`
	` description Permit DC routes`
	` match tag 300`
	`!`
	`route-map MATCH_DC_ROUTE_V6 deny 5`
	` description Deny IPV6 default route`
	` match ipv6 address prefix-list DENY_IPV6_0`
	`!`
	`route-map MATCH_DC_ROUTE_V6 deny 17`
	` description Deny L3Handoff IPV6 Prefixes`
	
	` match ipv6 address prefix-list L3HANDOFF_PREFIXES`
	`!`
	`route-map MATCH_DC_ROUTE_V6 permit 20`
	` description Permit DC routes`
	` match tag 300`	

Step 8	**end**
	Example:
	`Device(config-route-map)#` **`end`**
		Returns to privileged EXEC mode.

Configuration Examples for Border Node

The example configurations described in this section are for a border node of a LISP VXLAN fabric that is shown in the Figure 4: LISP VXLAN Fabric Topology. The fabric illustrated in the topology consists of a border node, a control plane node, and two fabric edge nodes. VLAN50 is configured on Fabric Edge 1 and VLAN91 is configured on Fabric Edge 2.

Figure 4: LISP VXLAN Fabric Topology



Configuration Example for an External Border Node

An external border node connects to the network that is external to the fabric, such as the internet. An external border is the default exit point for the virtual networks in the fabric. Ensure that you configure the external border with default routes to reach external unknown destinations.

Here is a sample configuration for an external border with Layer 3 handoff. In the Figure 4: LISP VXLAN Fabric Topology:

- External border has a Loopback0 address of 172.16.1.67
- Control plane node has a Loopback0 address of 172.16.1.66
- Layer 3 handoff segment for VN3 (user-defined VRF) is 10.20.1.0/30, 2001:DB8:20::/126
- Layer 3 handoff segment for Default Instance is 10.20.2.0/30

Ensure that there is IP reachability between all fabric nodes in the underlay.

EBN

```
vrf definition VN3
 rd 1:4099
 !
 address-family ipv4
  route-target export 1:4099
  route-target import 1:4099
 exit-address-family
 !
 address-family ipv6
  route-target export 1:4099
  route-target import 1:4099
 exit-address-family

vlan 222
 name 222
!
vlan 111
 name 111
!
interface Vlan111
 description interface to External router
 ip address 10.20.2.1 255.255.255.252
 no ip redirects
!
interface Vlan222
 description interface to External router
 vrf forwarding VN3
 ip address 10.20.1.1 255.255.255.252
 no ip redirects
 ipv6 address 2001:DB8:20::1/126
 ipv6 enable

!
interface FortyGigabitEthernet1/0/4
 switchport mode trunk

interface Loopback50
 description Loopback Border
 vrf forwarding VN3
 ip address 10.50.1.1 255.255.255.255
 ipv6 address 2001:DB8:2050::1/128
 ipv6 enable
 ipv6 dhcp relay trust
!

interface Loopback91
 description Loopback Border
 ip address 10.91.1.1 255.255.255.255
!

router lisp
 locator-table default
 locator-set default_etr_locator
  IPv4-interface Loopback0 priority 10 weight 10
  exit-locator-set
 !
 locator-set eid_locator
  IPv4-interface Loopback0 priority 10 weight 10
  auto-discover-rlocs
  exit-locator-set
 !
 locator default-set eid_locator
!
 service ipv4
  encapsulation vxlan
  map-cache publications
  import publication publisher 172.16.1.66
  itr map-resolver 172.16.1.66
  etr map-server 172.16.1.66 key some-key
  etr map-server 172.16.1.66 proxy-reply
  etr
  sgt
  route-export publications
  distance publications 250
  proxy-etr
  proxy-itr 172.16.1.67
  exit-service-ipv4
 !
 service ipv6
  encapsulation vxlan
  map-cache publications
  import publication publisher 172.16.1.66
  itr map-resolver 172.16.1.66
  etr map-server 172.16.1.66 key some-key
  etr map-server 172.16.1.66 proxy-reply
  etr
  sgt
  route-export publications
  distance publications 250
  proxy-etr
  proxy-itr 172.16.1.67
  exit-service-ipv6
 !
 instance-id 4097
  remote-rloc-probe on-route-change
  service ipv4
   eid-table default
   map-cache 10.91.1.0/24 map-request
   exit-service-ipv4
  !
  instance-id 4099
  remote-rloc-probe on-route-change
  service ipv4
   eid-table vrf VN3
   database-mapping 0.0.0.0/0 locator-set default_etr_locator default-etr local
   exit-service-ipv4
  !
  service ipv6
   eid-table vrf VN3
   database-mapping ::/0 locator-set default_etr_locator default-etr local
   exit-service-ipv6
  !
  exit-instance-id
 !
 ipv4 locator reachability minimum-mask-length 32
 ipv4 source-locator Loopback0
 exit-router-lisp
!
 router bgp 600
 bgp router-id interface Loopback0
 bgp log-neighbor-changes
 bgp graceful-restart
 !
 address-family ipv4
  bgp redistribute-internal
  bgp aggregate-timer 0
  network 10.20.2.0 mask 255.255.255.252
  network 10.91.1.1 mask 255.255.255.255
  aggregate-address 10.91.1.0 255.255.255.0 summary-only
  redistribute lisp metric 10 route-map LISP_TO_BGP
  neighbor 10.20.2.2 remote-as 300
  neighbor 10.20.2.2 update-source Vlan111
  neighbor 10.20.2.2 activate
  neighbor 10.20.2.2 send-community both
 exit-address-family !
 !
 address-family ipv4 vrf VN3
  bgp aggregate-timer 0
  network 10.20.1.0 mask 255.255.255.252
  network 10.50.1.1 mask 255.255.255.255
  aggregate-address 10.50.1.0 255.255.255.0 summary-only
  redistribute lisp metric 10 route-map LISP_TO_BGP
  neighbor 10.20.1.2 remote-as 300
  neighbor 10.20.1.2 update-source Vlan222
  neighbor 10.20.1.2 activate
  neighbor 10.20.1.2 send-community both
  neighbor 10.20.1.2 weight 65535
 exit-address-family
 !
 address-family ipv6 vrf VN3
  redistribute lisp metric 10 route-map LISP_TO_BGP
  bgp aggregate-timer 0
  network 2001:DB8:20::/126
  network 2001:DB8:2050::1/128
  aggregate-address 2001:DB8:50::/64 summary-only
  neighbor 2001:DB8:20::2 remote-as 300
  neighbor 2001:DB8:20::2 update-source Vlan222
  neighbor 2001:DB8:20::2 activate
  neighbor 2001:DB8:20::2 send-community both
  neighbor 2001:DB8:20::2 weight 65535
 exit-address-family
!

route-map LISP_TO_BGP permit 10
 description AS-number tag
 set as-path tag
```

Verify Distributed Border and Control Plane Node

You can verify the configurations on the control plane node, border node and the fabric edge node using the **show** commands. This section provides sample outputs for the **show** commands on the fabric node devices in the topology wherein the border and control plane nodes are not colocated.

In the topology, 172.16.1.68 and 172.16.1.69 are Fabric Edge Nodes; 172.16.1.67 is the Border Node; 172.16.1.66 is the Control Plane Node.

Table 1: Show Commands for the Control Plane Node

View the LISP session details on the control plane node:

```
CP# show lisp session

Sessions for VRF default, total: 6, established: 3
Peer                           State      Up/Down        In/Out    Users
172.16.1.69:16244                Up         02:17:44        9/17     7
172.16.1.68:37085                Up         02:17:46        9/20     7
172.16.1.67:11364                Up         00:07:04       13/47     7

CP#
```

Table 2: Show Commands for the Border Node

View the LISP session details on the border node:

```
Border# show lisp session

Sessions for VRF default, total: 1, established: 1
Peer                           State      Up/Down        In/Out    Users
172.16.1.66:4342                 Up         00:07:21       47/13     7
Border#
```

View the Locator Set information on the border node:

```
Border# show lisp locator-set
LISP Locator-set information:

172.16.1.67, local, reachable, loopback
Border#
```

View the information about LISP instance IDs for IPv4 service:

```
Border# show lisp instance-id * ipv4

=================================================
Output for router lisp 0 instance-id 4097
=================================================
  Instance ID:                              4097
  Router-lisp ID:                           0
  Locator table:                            default
  EID table:                                default
  Ingress Tunnel Router (ITR):              disabled
  Egress Tunnel Router (ETR):               enabled
  Proxy-ITR Router (PITR):                  enabled RLOCs: 172.16.1.67
  Proxy-ETR Router (PETR):                  enabled
  NAT-traversal Router (NAT-RTR):           disabled
  Mobility First-Hop Router:                disabled
  Map Server (MS):                          disabled
  Map Resolver (MR):                        disabled
  Mr-use-petr:                              disabled
  First-Packet pETR:                        disabled
  Multiple IP per MAC support:              disabled
  Delegated Database Tree (DDT):            disabled
  Multicast Flood Access-Tunnel:            disabled
  Publication-Subscription:                 enabled
    Publisher(s):                           172.16.1.66
  Site Registration Limit:                  0
  Map-Request source:                       derived from EID destination
  ITR Map-Resolver(s):                      172.16.1.66
  ETR Map-Server(s):                        172.16.1.66 (never)
  xTR-ID:                                 0x585ED747-0x87D8E878-0xC58A505D-0x10E643FC

  site-ID:                                  unspecified
  ITR local RLOC (last resort):             172.16.1.67
  ITR Solicit Map Request (SMR):            accept and process
    Max SMRs per map-cache entry:           8 more specifics
    Multiple SMR suppression time:          2 secs
  ETR accept mapping data:                  disabled, verify disabled
  ETR map-cache TTL:                        1d00h
  Locator Status Algorithms:
    RLOC-probe algorithm:                   disabled
    RLOC-probe on route change:             N/A (periodic probing disabled)
    RLOC-probe on member change:            disabled
    LSB reports:                            process
    IPv4 RLOC minimum mask length:          /32
    IPv6 RLOC minimum mask length:          /0
  Map-cache:
    Static mappings configured:             1
    Map-cache size/limit:                   1/214528
    Imported route count/limit:             0/5000
    Map-cache activity check period:        60 secs
    Map-cache signal suppress:              disabled
    Conservative-allocation:                disabled
    Map-cache FIB updates:                  established
    Persistent map-cache:                   disabled
    Map-cache activity-tracking:            enabled
  Global Top Source locator configuration:
     Loopback0 (172.16.1.67)
  Database:
    Total database mapping size:            0
    static database size/limit:             0/214528
    dynamic database size/limit:            0/214528
    route-import database size/limit:       0/5000
    import-site-reg database size/limit:    0/214528
    dummy database size/limit:             0/214528
    import-publication database size/limit: 0/214528
    import-publication-cfg-prop database siz0
    proxy database size:                    0
    Inactive (deconfig/away) size:          0
  Publication entries exported to:
    Map-cache:                              0
    RIB:                                    0
    Database:                               0
    Prefix-list:                            0
  Site-registeration entries exported to:
    Map-cache:                              0
    RIB:                                    0
  Publication (Type - Config Propagation) en
    Database:                               0
  Encapsulation type:                       vxlan

=================================================
Output for router lisp 0 instance-id 4099
=================================================
  Instance ID:                              4099
  Router-lisp ID:                           0
  Locator table:                            default
  EID table:                                vrf VN3
  Ingress Tunnel Router (ITR):              disabled
  Egress Tunnel Router (ETR):               enabled
  Proxy-ITR Router (PITR):                  enabled RLOCs: 172.16.1.67
  Proxy-ETR Router (PETR):                  enabled
  NAT-traversal Router (NAT-RTR):           disabled
  Mobility First-Hop Router:                disabled
  Map Server (MS):                          disabled
  Map Resolver (MR):                        disabled
  Mr-use-petr:                              disabled
  First-Packet pETR:                        disabled
  Multiple IP per MAC support:              disabled
  Delegated Database Tree (DDT):            disabled
  Multicast Flood Access-Tunnel:            disabled
  Publication-Subscription:                 enabled
    Publisher(s):                           172.16.1.66
  Site Registration Limit:                  0
  Map-Request source:                       derived from EID destination
  ITR Map-Resolver(s):                      172.16.1.66
  ETR Map-Server(s):                        172.16.1.66 (00:37:05)
  xTR-ID:                                  0x585ED747-0x87D8E878-0xC58A505D-0x10E643FC

  site-ID:                                  unspecified
  ITR local RLOC (last resort):             172.16.1.67
  ITR Solicit Map Request (SMR):            accept and process
    Max SMRs per map-cache entry:           8 more specifics
    Multiple SMR suppression time:          2 secs
  ETR accept mapping data:                  disabled, verify disabled
  ETR map-cache TTL:                        1d00h
  Locator Status Algorithms:
    RLOC-probe algorithm:                   disabled
    RLOC-probe on route change:             N/A (periodic probing disabled)
    RLOC-probe on member change:            disabled
    LSB reports:                            process
    IPv4 RLOC minimum mask length:          /32
    IPv6 RLOC minimum mask length:          /0
  Map-cache:
    Static mappings configured:             0
    Map-cache size/limit:                   1/214528
    Imported route count/limit:             0/5000
    Map-cache activity check period:        60 secs
    Map-cache signal suppress:              disabled
    Conservative-allocation:                disabled
    Map-cache FIB updates:                  established
    Persistent map-cache:                   disabled
    Map-cache activity-tracking:            enabled
  Global Top Source locator configuration:
     Loopback0 (172.16.1.67)
  Database:
    Total database mapping size:            2
    static database size/limit:             2/214528
    dynamic database size/limit:            0/214528
    route-import database size/limit:       0/5000
    import-site-reg database size/limit:    0/214528
    dummy database size/limit:              0/214528
    import-publication database size/limit: 0/214528
    import-publication-cfg-prop database siz0
    proxy database size:                    0
    Inactive (deconfig/away) size:          0
  Publication entries exported to:
    Map-cache:                              0
    RIB:                                    0
    Database:                               0
    Prefix-list:                            0
  Site-registeration entries exported to:
    Map-cache:                              0
    RIB:                                    0
  Publication (Type - Config Propagation) en
    Database:                               0
  Encapsulation type:                       vxlan
Border#
```

View the route table on the border node for the VN3 VRF:

```
Border# show ip route vrf VN3

Routing Table: VN3
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, m - OMP
       n - NAT, Ni - NAT inside, No - NAT outside, Nd - NAT DIA
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       H - NHRP, G - NHRP registered, g - NHRP registration summary
       o - ODR, P - periodic downloaded static route, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR
       & - replicated local route overrides by connected

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 4 subnets, 3 masks
C        10.20.1.0/30 is directly connected, Vlan222
L        10.20.1.1/32 is directly connected, Vlan222
B        10.50.1.0/24 [200/0], 00:32:34, Null0
C        10.50.1.1/32 is directly connected, Loopback50
Border#
```

Table 3: Show Commands for the Fabric Edge Node

View the LISP sessions on the fabric edge node:

```
FabricEdge# show lisp session

Sessions for VRF default, total: 2, established: 1
Peer                           State      Up/Down        In/Out    Users
172.16.1.66:4342                 Up         02:21:53       20/9      14
FabricEdge#
```

View the Locator Set information on the fabric edge node:

```
FabricEdge# show lisp locator-set
LISP Locator-set information:

172.16.1.68, local, reachable, loopback

FabricEdge#
```

View the route table on the fabric edge node for the VN3 VRF:

```
FabricEdge# show ip route vrf VN3

Routing Table: VN3
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, m - OMP
       n - NAT, Ni - NAT inside, No - NAT outside, Nd - NAT DIA
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       H - NHRP, G - NHRP registered, g - NHRP registration summary
       o - ODR, P - periodic downloaded static route, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR
       & - replicated local route overrides by connected

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        10.50.1.0/24 is directly connected, Vlan50
L        10.50.1.1/32 is directly connected, Vlan50
FabricEdge#
```

Configuration Example for an Internal Border Node

Here is a sample configuration for an internal border with Layer 3 handoff.

In the Figure 4: LISP VXLAN Fabric Topology:

- Internal border has a Loopback0 address of 172.16.1.67
- Control plane node has a Loopback0 address of 172.16.1.66
- Layer 3 handoff segment is 10.20.1.0/30, 2001:DB8:20::/126
- Layer 3 handoff segment for Default Instance is 10.20.2.0/30

Ensure that there is IP reachability between all fabric nodes in the underlay.

IBN

```
vrf definition VN3
 rd 1:4099
 !
 address-family ipv4
  route-target export 1:4099
  route-target import 1:4099
 exit-address-family
 !
 address-family ipv6
  route-target export 1:4099
  route-target import 1:4099
 exit-address-family
!

vlan 222
 name 222
!
vlan 111
 name 111
!
interface Vlan111
 description interface to External router
 ip address 10.20.2.1 255.255.255.252
 no ip redirects
!
interface Vlan222
 description interface to External router
 vrf forwarding VN3
 ip address 10.20.1.1 255.255.255.252
 no ip redirects
 ipv6 address 2001:DB8:20::1/126
 ipv6 enable
!
interface FortyGigabitEthernet1/0/4
 switchport mode trunk


interface Loopback50
 description Loopback Border
 vrf forwarding VN3
 ip address 10.50.1.1 255.255.255.255
 ipv6 address 2001:DB8:2050::1/128
 ipv6 enable
 ipv6 dhcp relay trust
!
interface Loopback91
 description Loopback Border
 ip address 10.91.1.1 255.255.255.255
!

router lisp
 locator-table default
 locator-set eid_locator
 IPv4-interface Loopback0 priority 10 weight 10
  auto-discover-rlocs
  exit-locator-set
 !
 locator default-set eid_locator
!
 service ipv4
  encapsulation vxlan
  map-cache publications
  import publication publisher 172.16.1.66
  itr map-resolver 172.16.1.66
  etr map-server 172.16.1.66 key some-key
  etr map-server 172.16.1.66 proxy-reply
  etr
  sgt
  route-export publications
  distance publications 250
  proxy-itr 172.16.1.67
  exit-service-ipv4
 !
 service ipv6
  encapsulation vxlan
  map-cache publications
  import publication publisher 172.16.1.66
  itr map-resolver 172.16.1.66
  etr map-server 172.16.1.66 key some-key
  etr map-server 172.16.1.66 proxy-reply
  etr
  sgt
  route-export publications
  distance publications 250
  proxy-itr 172.16.1.67
  exit-service-ipv6
 !
 instance-id 4097
  remote-rloc-probe on-route-change
  service ipv4
   eid-table default
   map-cache 10.91.1.0/24 map-request
   exit-service-ipv4
  !
  exit-instance-id
 !
 instance-id 4099
  remote-rloc-probe on-route-change
  service ipv4
   eid-table vrf VN3
   map-cache 0.0.0.0/0 map-request
   route-import database bgp 600 route-map MATCH_DC_ROUTE locator-set eid_locator
   exit-service-ipv4
  !
  service ipv6
   eid-table vrf VN3
   map-cache ::/0 map-request
   route-import database bgp 600 route-map MATCH_DC_ROUTE_V6 locator-set eid_locator
   exit-service-ipv6
  !
  exit-instance-id
 !
 ipv4 locator reachability minimum-mask-length 32
 ipv4 source-locator Loopback0
 exit-router-lisp


router bgp 600
 bgp router-id interface Loopback0
 bgp log-neighbor-changes
 bgp graceful-restart
 !
 address-family ipv4
  bgp redistribute-internal
  bgp aggregate-timer 0
  network 10.20.2.0 mask 255.255.255.252
  network 10.91.1.1 mask 255.255.255.255
  aggregate-address 10.91.1.0 255.255.255.0 summary-only
  redistribute lisp metric 10 route-map LISP_TO_BGP
  neighbor 10.20.2.2 remote-as 300
  neighbor 10.20.2.2 update-source Vlan111
  neighbor 10.20.2.2 activate
  neighbor 10.20.2.2 send-community both
 exit-address-family
 !
 address-family ipv4 vrf VN3
  bgp aggregate-timer 0
  network 10.20.1.0 mask 255.255.255.252
  network 10.50.1.1 mask 255.255.255.255
  aggregate-address 10.50.1.0 255.255.255.0 summary-only
  redistribute lisp metric 10 route-map LISP_TO_BGP
  neighbor 10.20.1.2 remote-as 300
  neighbor 10.20.1.2 update-source Vlan222
  neighbor 10.20.1.2 activate
  neighbor 10.20.1.2 send-community both
  neighbor 10.20.1.2 weight 65535
 exit-address-family
 !
 address-family ipv6 vrf VN3
  redistribute lisp metric 10 route-map LISP_TO_BGP
  bgp aggregate-timer 0
  network 2001:DB8:20::/126
  network 2001:DB8:2050::1/128
  aggregate-address 2001:DB8:2050::/64 summary-only
  neighbor 2001:DB8:20::2 remote-as 300
  neighbor 2001:DB8:20::2 update-source Vlan222
  neighbor 2001:DB8:20::2 activate
  neighbor 2001:DB8:20::2 send-community both
  neighbor 2001:DB8:20::2 weight 65535
 exit-address-family


!
route-map LISP_TO_BGP permit 10
 description AS-number tag
 set as-path tag
!

ip prefix-list DENY_0.0.0.0 seq 10 permit 0.0.0.0/0
!
ip prefix-list L3HANDOFF_PREFIXES seq 63755909 permit 10.20.2.0/30
ip prefix-list L3HANDOFF_PREFIXES seq 828011002 permit 10.20.1.0/30
!
ipv6 prefix-list DENY_IPV6_0 seq 10 permit ::/0
!
ipv6 prefix-list L3HANDOFF_PREFIXES seq 568642686 permit 2001:DB8:20::/126

route-map MATCH_DC_ROUTE deny 5
 description Deny IPV4 default route
 match ip address prefix-list DENY_0.0.0.0
!
route-map MATCH_DC_ROUTE deny 17
 description Deny L3Handoff Prefixes
 match ip address prefix-list L3HANDOFF_PREFIXES
!
route-map MATCH_DC_ROUTE permit 20
 description Permit DC routes
 match tag 300
!
route-map MATCH_DC_ROUTE_V6 deny 5
```

```
 description Deny IPV6 default route
 match ipv6 address prefix-list DENY_IPV6_0
!
route-map MATCH_DC_ROUTE_V6 deny 17
 description Deny L3Handoff IPV6 Prefixes
 match ipv6 address prefix-list L3HANDOFF_PREFIXES
!
route-map MATCH_DC_ROUTE_V6 permit 20
 description Permit DC routes
 match tag 300
```

Configuration Example for an Internal and External Border

Here is a sample configuration for an internal and external border with Layer 3 handoff.

In the Figure 4: LISP VXLAN Fabric Topology:

- Border has a Loopback0 address of 172.16.1.67
- Control plane node has a Loopback0 address of 172.16.1.66
- Layer 3 handoff segment for VN3 (user-defined VRF) is 10.20.1.0/30, 2001:DB8:20::/126
- Layer 3 handoff segment for Default Instance is 10.20.2.0/30

Ensure that there is IP reachability between all fabric nodes in the underlay.

Internal+External BN

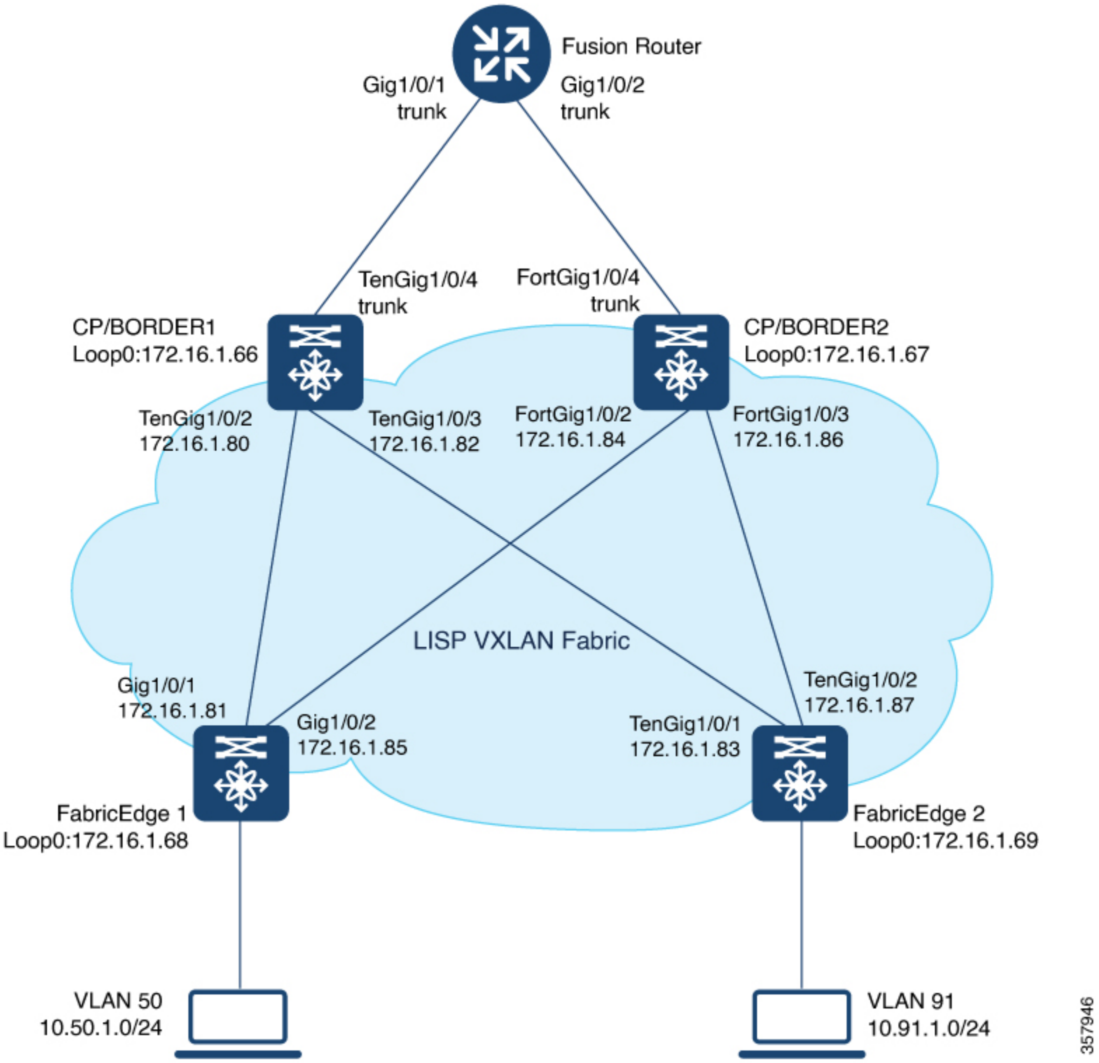
```
vrf definition VN3
 rd 1:4099
 !
 address-family ipv4
  route-target export 1:4099
  route-target import 1:4099
 exit-address-family
 !
 address-family ipv6
  route-target export 1:4099
  route-target import 1:4099
 exit-address-family

vlan 222
 name 222
!
vlan 111
 name 111
!
interface Vlan111
 description interface to External router
 ip address 10.20.2.1 255.255.255.252
 no ip redirects
!
interface Vlan222
 description interface to External router
 vrf forwarding VN3
 ip address 10.20.1.1 255.255.255.252
 no ip redirects
 ipv6 address 2001:DB8:20::1/126
 ipv6 enable

!
```

```
interface FortyGigabitEthernet1/0/4
 switchport mode trunk


interface Loopback50
 description Loopback Border
 vrf forwarding VN3
 ip address 10.50.1.1 255.255.255.255
 ipv6 address 2001:DB8:2050::1/128
 ipv6 enable
 ipv6 dhcp relay trust
!


interface Loopback91
 description Loopback Border
 ip address 10.91.1.1 255.255.255.255
!

router lisp
 locator-table default
 locator-set default_etr_locator
  IPv4-interface Loopback0 priority 10 weight 10
  exit-locator-set
 !
 locator-set eid_locator
  IPv4-interface Loopback0 priority 10 weight 10
  auto-discover-rlocs
  exit-locator-set
 !
 locator default-set eid_locator
!
 service ipv4
  encapsulation vxlan
  map-cache publications
  import publication publisher 172.16.1.66
  itr map-resolver 172.16.1.66
  etr map-server 172.16.1.66 key some-key
  etr map-server 172.16.1.66 proxy-reply
  etr
  sgt
  route-export publications
  distance publications 250
  proxy-etr
  proxy-itr 172.16.1.67
  exit-service-ipv4
 !
 service ipv6
  encapsulation vxlan
  map-cache publications
  import publication publisher 172.16.1.66
  itr map-resolver 172.16.1.66
  etr map-server 172.16.1.66 key some-key
  etr map-server 172.16.1.66 proxy-reply
  etr
  sgt
  route-export publications
  distance publications 250
  proxy-etr
  proxy-itr 172.16.1.67
  exit-service-ipv6
 !
 instance-id 4097
  remote-rloc-probe on-route-change
  service ipv4
   eid-table default
   map-cache 10.91.1.0/24 map-request
   exit-service-ipv4
  !
  instance-id 4099
  remote-rloc-probe on-route-change
  service ipv4
   eid-table vrf VN3
   database-mapping 0.0.0.0/0 locator-set default_etr_locator default-etr local
   route-import database bgp 600 route-map MATCH_DC_ROUTE locator-set eid_locator
   exit-service-ipv4
  !
  service ipv6
   eid-table vrf VN3
   database-mapping ::/0 locator-set default_etr_locator default-etr local
   route-import database bgp 600 route-map MATCH_DC_ROUTE_V6 locator-set eid_locator
   exit-service-ipv6
  !
  exit-instance-id
 !
 ipv4 locator reachability minimum-mask-length 32
 ipv4 source-locator Loopback0
 exit-router-lisp
!


router bgp 600
 bgp router-id interface Loopback0
 bgp log-neighbor-changes
 bgp graceful-restart
 !
 address-family ipv4
  bgp redistribute-internal
  bgp aggregate-timer 0
  network 10.20.2.0 mask 255.255.255.252
  network 10.91.1.1 mask 255.255.255.255
  aggregate-address 10.91.1.0 255.255.255.0 summary-only
  redistribute lisp metric 10 route-map LISP_TO_BGP
  neighbor 10.20.2.2 remote-as 300
  neighbor 10.20.2.2 update-source Vlan111
  neighbor 10.20.2.2 activate
  neighbor 10.20.2.2 send-community both
 exit-address-family
 !
 address-family ipv4 vrf VN3
  bgp aggregate-timer 0
  network 10.20.1.0 mask 255.255.255.252
  network 10.50.1.1 mask 255.255.255.255
  aggregate-address 10.50.1.0 255.255.255.0 summary-only
  redistribute lisp metric 10 route-map LISP_TO_BGP
  neighbor 10.20.1.2 remote-as 300
  neighbor 10.20.1.2 update-source Vlan222
  neighbor 10.20.1.2 activate
  neighbor 10.20.1.2 send-community both
  neighbor 10.20.1.2 weight 65535
 exit-address-family
 !
 address-family ipv6 vrf VN3
  redistribute lisp metric 10 route-map LISP_TO_BGP
  bgp aggregate-timer 0
  network 2001:DB8:20::/126
  network 2001:DB8:2050::1/128
  aggregate-address 2001:DB8:2050::/64 summary-only
  neighbor 2001:DB8:20::2 remote-as 300
  neighbor 2001:DB8:20::2 update-source Vlan222
  neighbor 2001:DB8:20::2 activate
  neighbor 2001:DB8:20::2 send-community both
  neighbor 2001:DB8:20::2 weight 65535
 exit-address-family
!

ip prefix-list DENY_0.0.0.0 seq 10 permit 0.0.0.0/0
!
ip prefix-list L3HANDOFF_PREFIXES seq 63755909 permit 10.20.2.0/30
ip prefix-list L3HANDOFF_PREFIXES seq 828011002 permit 10.20.1.0/30
!
ipv6 prefix-list DENY_IPV6_0 seq 10 permit ::/0
!
ipv6 prefix-list L3HANDOFF_PREFIXES seq 568642686 permit 2001:DB8:20::/126
!
route-map MATCH_DC_ROUTE deny 5
 description Deny IPV4 default route
 match ip address prefix-list DENY_0.0.0.0
!
route-map MATCH_DC_ROUTE deny 17
 description Deny L3Handoff Prefixes
 match ip address prefix-list L3HANDOFF_PREFIXES
!
route-map MATCH_DC_ROUTE permit 20
 description Permit DC routes
 match tag 300
!
route-map MATCH_DC_ROUTE_V6 deny 5
 description Deny IPV6 default route
 match ipv6 address prefix-list DENY_IPV6_0
!
route-map MATCH_DC_ROUTE_V6 deny 17
 description Deny L3Handoff IPV6 Prefixes
 match ipv6 address prefix-list L3HANDOFF_PREFIXES
!
route-map MATCH_DC_ROUTE_V6 permit 20
 description Permit DC routes
 match tag 300
!

route-map LISP_TO_BGP permit 10
 description AS-number tag
 set as-path tag
```

Configuration Example for Colocated Border Node

Here is a sample configuration for a colocated control plane node and external border node (BNCP) without Layer 3 handoff.

Figure 5: LISP VXLAN Fabric with Colocated Border and Control Plane Nodes



Ensure that there is IP reachability between all fabric nodes in the underlay.

BNCP

```
vrf definition VN3
 rd 1:4099
 !
 address-family ipv4
  route-target export 1:4099
  route-target import 1:4099
 exit-address-family
 !
 address-family ipv6
  route-target export 1:4099
  route-target import 1:4099
 exit-address-family
!
```

```
interface Loopback50
 description Loopback Border
 vrf forwarding VN3
 ip address 10.50.1.1 255.255.255.255
 ipv6 address 2001:DB8:2050::1/128
 ipv6 enable
 ipv6 dhcp relay trust
!
!
interface Loopback91
 description Loopback Border
 ip address 10.91.1.1 255.255.255.255
!

router lisp
 locator-table default
 locator-set default_etr_locator
  IPv4-interface Loopback0 priority 10 weight 10
  exit-locator-set
 !
 locator-set rloc_site1
  IPv4-interface Loopback0 priority 10 weight 10
  auto-discover-rlocs
  exit-locator-set
 !
 locator default-set rloc_set1
 service ipv4
  encapsulation vxlan
  map-cache publications
  import publication publisher 172.16.1.66
  import publication publisher 172.16.1.67
  itr map-resolver 172.16.1.66
  itr map-resolver 172.16.1.67
  etr map-server 172.16.1.66 key auth-key
  etr map-server 172.16.1.66 proxy-reply
  etr map-server 172.16.1.67 key some-key
  etr map-server 172.16.1.67 proxy-reply
  etr
  sgt
  route-export publications
  distance publications 250
  proxy-etr
  proxy-itr 172.16.1.66
  map-server
  map-resolver
  exit-service-ipv4
 !
 service ipv6
  encapsulation vxlan
  map-cache publications
  import publication publisher 172.16.1.66
  import publication publisher 172.16.1.67
  itr map-resolver 172.16.1.66
  itr map-resolver 172.16.1.67
  etr map-server 172.16.1.66 key auth-key
  etr map-server 172.16.1.66 proxy-reply
  etr map-server 172.16.1.67 key some-key
  etr map-server 172.16.1.67 proxy-reply
  etr
  sgt
  route-export publications
  distance publications 250
  proxy-etr
  proxy-itr 172.16.1.66
  map-server
  map-resolver
  exit-service-ipv6
 !

 instance-id 4097
  remote-rloc-probe on-route-change
  service ipv4
   eid-table default
   map-cache 10.91.1.0/24 map-request
   exit-service-ipv4
  !
  exit-instance-id
 !

 instance-id 4099
  remote-rloc-probe on-route-change
  service ipv4
   eid-table vrf VN3
   database-mapping 0.0.0.0/0 locator-set default_etr_locator default-etr local
   exit-service-ipv4
  !
  service ipv6
   eid-table vrf VN3
   database-mapping ::/0 locator-set default_etr_locator default-etr local
   exit-service-ipv6
  !
  exit-instance-id
 !
 site site_uci
  description map-server uci_map_server
  authentication-key some-key
  eid-record instance-id 4097 0.0.0.0/0 accept-more-specifics     //To import routes from
external network
  eid-record instance-id 4097 10.91.1.0/24 accept-more-specifics  //Fabric prefix
  eid-record instance-id 4099 0.0.0.0/0 accept-more-specifics     //To import routes from
external network
  eid-record instance-id 4099 10.50.1.0/24 accept-more-specifics  //Fabric prefix
  eid-record instance-id 4099 ::/0 accept-more-specifics          //To import routes from
external network
  eid-record instance-id 4099 2001:DB8:2050::/64 accept-more-specifics
  eid-record instance-id 8194 any-mac
  eid-record instance-id 8197 any-mac
  allow-locator-default-etr instance-id 4097 ipv4
  allow-locator-default-etr instance-id 4099 ipv4
  allow-locator-default-etr instance-id 4099 ipv6
  exit-site
 !
 ipv4 locator reachability minimum-mask-length 32
 ipv4 source-locator Loopback0
!

router bgp 700
 bgp router-id interface Loopback0
 bgp log-neighbor-changes
 bgp graceful-restart
 !
 address-family ipv4
  bgp redistribute-internal
  bgp aggregate-timer 0
  network 10.91.1.1 mask 255.255.255.255
 exit-address-family
 !
 address-family ipv4 vrf VN3
  bgp aggregate-timer 0
  network 10.50.1.1 mask 255.255.255.255
 exit-address-family
 !
 address-family ipv6 vrf VN3
  bgp aggregate-timer 0
  network 2001:DB8:2050::1/128
 exit-address-family
!
!
```

Verify Colocated Border and Control Plane Node

This section provides sample outputs for the **show** commands on the fabric edge nodes in the topology shown Figure 5: LISP VXLAN Fabric with Colocated Border and Control Plane Nodes.

In the topology, 172.16.1.68 and 172.16.1.69 are Fabric Edge Nodes; 172.16.1.67 is a colocated border and control plane node; 172.16.1.66 is another colocated border and control plane node.

The **show lisp session** command displays a summary of the the LISP sessions on the colocated control plane and border node device.

Note that the 4342 port on 172.16.1.66 and 172.16.1.67 is the control plane LISP server.

As you can see in the output below, each colocated control plane and border node shows two LISP sessions on the same device.

The LISP session entries for 172.16.1.66:4342 and 172.16.1.67:4342 indicate the LISP session from the border node to the control plane on the respective device. The LISP session entries 172.16.1.66:52946 and 172.16.1.67:13864 indicate the sessions from the control plane to the border on the respective device.

```
BNCP# show lisp session

Sessions for VRF default, total: 10, established: 6
Peer                           State      Up/Down        In/Out    Users
172.16.1.69:27785                Up         1d04h           9/27     8
172.16.1.66:4342                 Up         1d04h         172/27     7
172.16.1.66:52946                Up         1d04h          27/172    7
172.16.1.68:33554                Up         1d02h          11/17     8
172.16.1.67:4342                 Up         1d03h          39/17     8
172.16.1.67:13864                Up         1d03h          14/35     7
BNCP#
```

View the LISP session with the edge node:

```
BNCP# show lisp session 172.16.1.69

Peer address:     172.16.1.69:27785
Local address:    172.16.1.66:4342
Session Type:     Passive
Session State:    Up (1d04h)
Messages in/out:  9/27
Bytes in/out:     276/1666
Fatal errors:     0
Rcvd unsupported: 0
Rcvd invalid VRF: 0
Rcvd override:    0
Rcvd malformed:   0
Sent deferred:    0
SSO redundancy:   unsynchronized
Auth Type:        None

Accepting Users:  1
```

```
Users:           8
  Type                      ID                                    In/Out    State
  Capability Exchange       N/A                                    1/1      waiting
  MS Reliable Registration  lisp 0 IID 4097 AFI IPv4               1/0      idle
    WLC subscription received
  MS Reliable Registration  lisp 0 IID 4097 AFI IPv6               1/0      idle
    WLC subscription received
  MS Reliable Registration  lisp 0 IID 4099 AFI IPv4               1/0      idle
    WLC subscription received
  MS Reliable Registration  lisp 0 IID 4099 AFI IPv6               1/0      idle
    WLC subscription received
  MS Reliable Registration  lisp 0 IID 8194 AFI MAC                1/0      idle
    WLC subscription received
  MS Reliable Registration  lisp 0 IID 8197 AFI MAC                1/0      idle
    WLC subscription received
  MS Reliable Registration  lisp 0 IID 16777214 AFI IPv4           2/13     waiting
    WLC subscription received
BNCP#
```

View a summary of the LISP service IPv4 instances on the colocated border and control plane node:

```
BNCP# show lisp service ipv4 summary
Router-lisp ID:   0
Instance count:   5
Key: DB - Local EID Database entry count (@ - RLOC check pending
                                          * - RLOC consistency problem),
     DB no route - Local EID DB entries with no matching RIB route,
     Cache - Remote EID mapping cache size, IID - Instance ID,
     Role - Configured Role

                      Interface    DB  DB no  Cache Incom Cache
EID VRF name             (.IID)  size  route   size plete  Idle Role
default              LISP0.4097     0      0      1  0.0%  0.0% ETR-PITR-PETR
VN3                  LISP0.4099     1      1      0    0%    0% ETR-PITR-PETR

Number of eid-tables:                                 2
Total number of database entries:                     1 (inactive 0)
Maximum database entries:                        214528
EID-tables with inconsistent locators:                0
Total number of map-cache entries:                    1
Maximum map-cache entries:                       214528
EID-tables with incomplete map-cache entries:         0
EID-tables pending map-cache update to FIB:           0
BNCP1#
```

View the LISP EID statistics related to packet encapsulations, de-encapsulations, map requests, map replies, map registers, and other LISP-related packets on the colocated border and control plane node::

```
BNCP# show lisp service ipv4 statistics
LISP EID Statistics for all EID instances - last cleared: never
Control Packets:
  Map-Requests in/out:                              170/2
    Map-Requests in (5 sec/1 min/5 min):            0/5/22
    Encapsulated Map-Requests in/out:               51/0
    RLOC-probe Map-Requests in/out:                 119/2
    SMR-based Map-Requests in/out:                  0/0
    Extranet SMR cross-IID Map-Requests in:         0
    Map-Requests expired on-queue/no-reply          0/0
    Map-Resolver Map-Requests forwarded:            0
    Map-Server Map-Requests forwarded:              0
  Map-Reply records in/out:                         0/0
    Authoritative records in/out:                   0/0
    Non-authoritative records in/out:               0/0
    Negative records in/out:                        0/0
    RLOC-probe records in/out:                      0/0
    Map-Server Proxy-Reply records out:             0
  WLC Map-Subscribe records in/out:                 11/5
    Map-Subscribe failures in/out:                  0/0
  WLC Map-Unsubscribe records in/out:               0/0
    Map-Unsubscribe failures in/out:                0/0
  Map-Register records in/out:                      16/14
    Map-Registers in (5 sec/1 min/5 min):           0/0/0
    Map-Server AF disabled:                         0
    Not valid site eid prefix:                      7
    Authentication failures:                        0
    Disallowed locators:                            0
    Miscellaneous:                                  0
  WLC Map-Register records in/out:                  0/0
    WLC AP Map-Register in/out:                     0/0
    WLC Client Map-Register in/out:                 0/0
    WLC Map-Register failures in/out:               0/0
  Map-Notify records in/out:                        22/35
    Authentication failures:                        0
  WLC Map-Notify records in/out:                    0/0
    WLC AP Map-Notify in/out:                       0/0
    WLC Client Map-Notify in/out:                   0/0
    WLC Map-Notify failures in/out:                 0/0
  Publish-Subscribe in/out:
    Subscription Request records in/out:            6/6
      IID subscription requests in/out:             6/6
      Pub-refresh subscription requests in/out:     0/0
      Policy subscription requests in/out:          0/0
    Subscription Request failures in/out:           0/0
    Subscription Status records in/out:             11/10
      End of Publication records in/out:            11/10
      Subscription rejected records in/out:         0/0
      Subscription removed records in/out:          0/0
    Subscription Status failures in/out:            0/0
    Solicit Subscription records in/out:            12/15
    Solicit Subscription failures in/out:           0/0
    Publication records in/out:                     7/6
    Publication failures in/out:                    0/0
Errors:
  Mapping record TTL alerts:                        0
  Map-Request invalid source rloc drops:            0
  Map-Register invalid source rloc drops:           0
  DDT Requests failed:                              0
  DDT ITR Map-Requests dropped:                     0 (nonce-collision: 0, bad-xTR-nonce:
0)
Cache Related:
  Cache entries created/deleted:                    1/0
  NSF CEF replay entry count                        0
  Number of rejected EID-prefixes due to limit:     0
Forwarding:
  Number of data signals processed:                 0 (+ dropped 0)
  Number of reachability reports:                   0 (+ dropped 0)
  Number of SMR signals dropped:                    0
LISP RLOC Statistics - last cleared: never
Control Packets:
  RTR Map-Requests forwarded:                       0
  RTR Map-Notifies forwarded:                       0
  DDT-Map-Requests in/out:                          0/0
  DDT-Map-Referrals in/out:                         0/0
Errors:
  Map-Request format errors:                        0
  Map-Reply format errors:                          0
  Map-Referral format errors:                       0
LISP Miscellaneous Statistics - last cleared: never
Errors:
  Invalid IP version drops:                         0
  Invalid IP header drops:                          0
  Invalid IP proto field drops:                     0
  Invalid packet size drops:                        0
  Invalid LISP control port drops:                  0
  Invalid LISP checksum drops:                      0
  Unsupported LISP packet type drops:               0
  Unknown packet drops:                             0
BNCP#
```

View the detailed information on the remote IPv4 EID-prefix forwarding. Remote EID-prefixes are the destination prefixes.

```
BNCP# show lisp service ipv4 forwarding eid remote detail
Prefix                 Fwd action  Locator status bits   encap_iid
10.91.1.0/24           signal      0x00000000            N/A
  packets/bytes      2/1152
  path list 7FAE553FE0D8, 4 locks, per-destination, flags 0x49 [shble, rif, hwcn]
    ifnums:
      LISP0.4097(75)
    1 path
      path 7FAE574157A8, share 1/1, type attached prefix, for IPv4
        attached to LISP0.4097, glean for LISP0.4097
    1 output chain
      chain[0]: glean for LISP0.4097
BNCP#
```

View the LISP IPv4 service instance forwarding state.

```
BNCP# show lisp service ipv4 forwarding state
LISP forwarding state for EID table IPv4:Default
  Instance ID                  4097
  EID VRF                      Default (0x0)
    IPv4
      Configured roles         ETR|PITR|PETR
      EID table                IPv4:Default
      ALT table                <null>
      Locator status bits      Disabled
      Nonce                    SGT
      TTL Propagation          Enabled
      Table Suppression        Disabled
      SGT Policy Fwd           Disabled
    IPv6
      Configured role          DISABLED
      EID table                <null>
      ALT table                <null>
      Locator status bits      Disabled
      Nonce                    N/A
      TTL Propagation          Enabled
      Table Suppression        Disabled
      SGT Policy Fwd           Disabled
    L2
      Configured role          DISABLED
      L2 Domain ID             0
      IPv4 Unnum I/F           N/A
      IPv6 Unnum I/F           N/A
    RLOC transport VRF         Default (0x0)
      IPv4 RLOC table          IPv4:Default
      IPv6 RLOC table          IPv6:Default
      IPv4 path MTU discovery  min  576 max 65535
      IPv6 path MTU discovery  min 1280 max 65535
      IPv4 RLOC fltr handle    0x0
      IPv6 RLOC fltr handle    0x0
    LISP router ID             0
    LISP virtual interface     LISP0.4097
    User                       LISP
BNCP#
```

```
BNCP# show lisp service ipv4 forwarding statistics
IPv4 LISP Forwarding Statistics
 Map requests              0
 Map requests resolve DGT  0
 Unexpected map requests   0
 Map cache deletes         0
BNCP#
```

View the dynamic interfaces that are created after LISP configuration on the colocated control plane and border node:

```
BNCP# show ip interface brief | i LISP
Interface              IP-Address      OK? Method Status                Protocol
LISP0                  unassigned      YES unset  up                    up
LISP0.4097             172.16.1.66     YES unset  up                    up
LISP0.4099             10.50.1.1       YES unset  up                    up
BNCP#
```

CHAPTER 5

Configuring Fabric Edge Node

A LISP VXLAN fabric edge node is the access layer where the traffic enters or exits the network towards the users, devices or endpoints. You can configure the following platforms as a fabric edge node:

- Cisco Catalyst 9300 Series Switches
- Cisco Catalyst 9400 Series Switches
- Cisco Catalyst 9500 Series Switches

Functions of Fabric Edge Node

A fabric edge node performs the following functions in the fabric:

- **Endpoint Registration**: Identifies and authenticates a wired endpoint before registering the endpoint ID information with the control plane node.
- **AAA Authenticator**: An integral part of the IEEE 802.1X port-based authentication process, the edge node collects authentication credentials from the connected devices, relays it to the Authentication Server, and enforces the authorization result.
- **Anycast Layer 3 Gateway**: An edge node acts as Layer 3 anycast gateway, providing optimal forwarding and mobility for the endpoints within the fabric. On edge nodes, the anycast Layer 3 gateway is instantiated as a Switched Virtual Interface (SVI) with a hard-coded anycast MAC address that is uniform across all edge nodes within the fabric site.
- **VXLAN encapsulation/decapsulation**: Packets received from the end points are encapsulated by the fabric edge node. Depending on the destination, the encapsulated packets are forwarded to another edge node or the border node. When fabric encapsulated traffic is received for an endpoint, the fabric edge node decapsulates the traffic and sends it to that endpoint.

How to Configure a Fabric Edge Node

Note Before you begin, ensure that the underlay network links are configured for routed access connectivity.

Step	Task	Purpose
Step 1	Configure VRF	Configure a VRF to support IPv4 and IPv6 routing tables. VRF maintains the routing and forwarding information for devices within a virtual network. A VRF instance has its own IP routing table, a forwarding table, and one or more interfaces assigned to it. The VRF tables help the routing device reach the locator address space.
Step 2	Configure DHCP Options and Snooping	Configure a fabric edge node as a DHCP relay agent to relay the DHCP traffic between fabric endpoints and DHCP server. DHCP Snooping on a VLAN enables DT-PROGRAMMATIC policy that supports onboarding of DHCPv4 hosts.
Step 3	Configure Device Tracking	Configure Switch Integrated Security Features based (SISF-based) device tracking to track the presence, location, and movement of endpoints in the fabric. SISF snoops traffic received by the device, extracts device identity (MAC and IP address), and stores them in a binding table.
Step 4	Configure VLANs	Configure VLANs to segment your network and achieve traffic isolation between the segments.
Step 5	Configure an SVI Interface	Configure an SVI interface for each VRF and for the Default Instance. An SVI interface is a VLAN interface that allows traffic to be routed between the VRFs.

Step	Task	Purpose
Step 6	Configure LISP	• Set up the Ingress Tunnel Router (ITR) functionality for both IPv4 and IPv6 address families. An ITR encapsulates and forwards the incoming packets across the overlay either to another fabric edge node or to the border node, depending on the destination. • Set up the Egress Tunnel Router (ETR) functionality for both IPv4 and IPv6 address families. An ETR decapsulates the received VXLAN-encapsulated packets and sends the packets to the endpoint.
Step 7	Configure Layer 3 VNI and Segment for Default Instance Configure Layer 3 VNI and Segment for User-Defined VRF	In a LISP VXLAN fabric, the VXLAN-GPO header has a VXLAN Network Identifier (VNI) field that servers as an identifier of a specific virtual network. VXLAN VNI helps carry the macro segmentation information within the fabric site. A Layer 3 VNI identifies a Layer 3 overlay. • Configure Layer 3 VNI for the Default Instance. The default instance is used to connect the network infrastructure elements like Access Points and Layer 2 switches to the fabric access layer. • Configure Layer 3 VNI for VLANs in User-Defined VRF.
Step 8	Configure Layer 2 VNI and Segment for Default Instance Configure Layer 2 VNI for VLANs in User-Defined VRF	A Layer 2 VNI identifies a Layer 2 overlay. • Configure Layer 2 VNI for the Default Instance. • Configure Layer 2 VNI for the User-Defined VRF. Configuring Layer 2 VNI programmatically enables these first-hop-security policies on the VLANs: LISP-DT-GUARD-VLAN and LISP-AR-RELAY-VLAN. LISP-DT-GUARD-VLAN policy mitigates IP theft, MAC theft and DOS attacks. LISP-AR-RELAY policy helps in converting ARP broadcast and Neighbor Solicitation (NS) multicast packets to unicast.

Step	Task	Purpose
Step 9	Verify the configurations on the fabric edge node using these show commands: For sample outputs of the **show** commands, refer Verify the Configuration of Fabric Edge Node, on page 108.	
	show lisp session	Displays a summary of the LISP sessions that the fabric edge node has established with the control plane node.
	show lisp service ipv4 statistics **show lisp service ipv6 statistics**	Displays the LISP packet statistics for all EID prefixes. Use this command to check the total number of packet encapsulations, decapsulations, map requests, map replies, map registers, and other LISP-related packet information, for the IPv4 or IPv6 service.
	show lisp service ipv4 summary **show lisp service ipv6 summary**	Displays a summary of the LISP service instances that are created on the device.
	show ip interface brief	Displays a summary of the LISP interfaces that are created dynamically. Filter the output to view the dynamically created LISP interfaces, using the **show ip interface brief | i LISP** command.
	show lisp locator-set	Displays information about the Locator Set configured on the fabric edge node.
	show ip route vrf	Displays the routing table that is configured on the fabric edge node, for a specified VRF.
	show lisp platform	Displays the limits of the given platform or the device. This command shows the LISP instance limits, Layer 3 limits, Layer 2 limits, and the supported configuration style on the device. Use this command to understand the limits of the device before planning its usage and role in the fabric.

Configure VRF

To configure a VRF on a fabric edge node, perform this task:

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	**vrf definition** *vrf-name* **Example:** `Device(config)# vrf definition campus`	Configures a VRF table, and enters VRF configuration mode.

Step 4	**address-family** {**ipv4** | **ipv6**}
	Example:
	`Device(config-vrf)# address-family ipv4`
		Specifies the address family as IPv4, and enters address family configuration mode.

Step 5	**exit-address-family** **Example:** `Device(config-vrf-af)# exit-address-family`	Exits address family configuration mode, and enters VRF configuration mode.

Step 6	**end**
	Example:
	`Device(config-vrf)# end`
		Returns to privileged EXEC mode.

Configure Device Tracking

To configure device tracking on a fabric edge node, perform this task:

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.

	Command or Action	Purpose
Step 3	**device-tracking policy** *policy-name* **Example:** `Device(config)# device-tracking policy IPDT_POLICY`	Creates a device-tracking policy with the specified name, and enters the device-tracking configuration mode.
Step 4	**tracking enable** **Example:** `Device(config-device-tracking)# tracking enable`	Enables polling for the specified policy.
Step 5	**exit** **Example:** `Device(config-device-tracking)# exit`	Exits device-tracking configuration mode, and enters global configuration mode.
Step 6	**interface** *interface-id* **Example:** `Device(config)# interface GigabitEthernet1/0/3`	Specifies an interface and enters interface configuration mode.
Step 7	**device-tracking attach-policy** *policy-name* **Example:** `Device(config-if)# device-tracking attach-policy IPDT_POLICY`	Attaches the device tracking policy to the interface.

Step 8	**end**
	Example:
	`Device(config-device-tracking)# end`
		Returns to privileged EXEC mode.

Configure VLANs

To configure VLAN on a fabric edge node, perform this task:

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.

	Command or Action	Purpose
Step 3	**vlan configuration** *vlan-id* **Example:** `Device(config)# vlan configuration 50`	Allows you to configure VLANs without actually creating them.
Step 4	**ipv6 nd raguard** **Example:** `Device(config)# ipv6 nd raguard`	Configures the default Router Advertisement (RA) Guard policy on the VLAN. The RA Guard feature analyzes the RAs and filters out bogus RAs sent by unauthorized devices. In host mode, all router advertisement and router redirect messages are disallowed on the port.
Step 5	**ipv6 dhcp guard** **Example:** `Device(config)# ipv6 dhcp guard`	Configures the default DHCP Guard policy on the VLAN. The IPv6 DHCP Guard feature blocks reply and advertisement messages that come from unauthorized DHCPv6 servers and relay agents.
Step 6	**vlan** *vlan-id* **Example:** `Device(config)# vlan 50`	Specifies a VLAN ID, and enters VLAN configuration mode.
Step 7	**name** *vlan-name* **Example:** `Device(config-vlan)# name AVlan50`	Specifies a name for the VLAN.
Step 8	**exit** **Example:** `Device(config-vlan)# exit`	Exits VLAN configuration mode, and enters global configuration mode.
Step 9	**vlan** *vlan-id* **Example:** `Device(config)# vlan 91`	Specifies a VLAN ID, and enters VLAN configuration mode.
Step 10	**name** *vlan-name* **Example:** `Device(config-vlan)# name AVlan91`	Specifies a name for the VLAN.
Step 11	**exit** **Example:** `Device(config-vlan)# exit`	Exits VLAN configuration mode, and enters global configuration mode.
Step 12	**end** **Example:**	Returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config)# end`	

Configure an SVI Interface

To configure an SVI interface for a VLAN on a fabric edge node, perform this task.

Repeat these steps to configure an SVI interface for each VLAN.

To configure an SVI interface for a Default Instance, execute only those steps that are applicable to the IPv4 address family. Do not execute the commands for IPv6 address family because a default instance does not support IPv6.

Note IPv6 client address assignment through Stateless Address Auto-Configuration (SLAAC) depends on Router Solicitation (RS), Router Advertisement (RA), Neighbor Solicitation (NS), and Neighbor Discovery (ND) message sequences. A default RA interval of 200 seconds results in a longer duration for IP address resolution. To enable faster address convergence using SLAAC, we recommend that you configure a lower RA interval, such as 1000 milliseconds.

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	**interface** *vlan-id*
	Example:
	For a user-defined VRF:
	`Device(config)# interface Vlan50`
	For a Default Instance:
	`Device(config)# interface Vlan91`
		Specifies the interface for which you are adding a description, and enters interface configuration mode.

Step 4	**description** *string* **Example:** `Device(config-if)# description server1`	Adds a description for an interface.
Step 5	**mac-address** *address* **Example:**	Specifies the MAC address for the VLAN interface (SVI).

	Command or Action	Purpose

	For a user-defined VRF:
	`Device(config-if)# mac-address 0000.0c9f.f18e`
	For a Default Instance:
	`Device(config-if)# mac-address 0000.0c9f.f984`
		We recommend that you use a MAC address starting from the base range value of 0000.0C9F.F05F.
		Note Configure the same MAC address for a given SVI on all the fabric edge nodes.

Step 6	**vrf forwarding** *name* **Example:** `Device(config-if)# vrf forwarding VN3`	Associates the VRF instance with the interface. **Note** This step is not applicable for an SVI of the default instance.

Step 7	**ip address** *ip_address subnet_mask*
	Example:
	For a user-defined VRF:
	`Device(config-if)# ip address 10.50.1.1 255.255.255.0`
	For a Default Instance:
	`Device(config-if)# ip address 10.91.1.1 255.255.255.0`
		Configures the IP address and IP subnet.
		This is the a common EID subnet that is shared across all the fabric edge nodes and the SVI is the Anycast Layer 3 Gateway.

Step 8	**ip helper-address** *ip_address* **Example:** `Device(config-if)# ip helper-address 172.16.2.2`	Configures the IP helper address. DHCP broadcasts will be forwarded as a unicast to this specific helper address rather than be dropped by the router.
Step 9	**no ip redirects** **Example:** `Device(config-if)# no ip redirects`	Disables sending of Internet Control Message Protocol (ICMP) redirect messages.
Step 10	**ipv6 address** *address* **Example:** `Device(config-if)# ipv6 address 2001:DB8:2050::1/64`	Configures an IPv6 address on the interface.
Step 11	**ipv6 enable** **Example:** `Device(config-if)# ipv6 enable`	Enables IPv6 on the interface.

Step 12	**ipv6 nd** {**dad attempts** | **prefix** | **managed-config-flag** | **other-config-flag** | **router-preference** | }
	Example:
	`Device(config-if)# ipv6 nd dad attempts 0`
	`Device(config-if)# ipv6 nd prefix`
		Configures IPv6 neighbor discovery on the interface.
		• **dad attempts**: Specifies the number of consecutive neighbor solicitation messages that are sent on an interface while duplicate address detection is

	Command or Action	Purpose

	`2001:DB8:2050::/64 2592000 604800 no-autoconfig`
	`Device(config-if)# ipv6 nd managed-config-flag`
	`Device(config-if)# ipv6 nd other-config-flag`
	`Device(config-if)# ipv6 nd router-preference High`
		performed on the unicast IPv6 addresses of the interface.
		• **prefix**: Specifies IPv6 prefixes that are included in IPv6 neighbor discovery router advertisements.
		• **managed-config-flag**: Specifies IPv6 interfaces neighbor discovery to allow the hosts to uses DHCP for address configuration.
		• **other-config-flag**: Specifies IPv6 interfaces neighbor discovery to allow the hosts to uses DHCP for non-address configuration.
		• **router-preference**: Specifies a default router preference (DRP) for the router on a specific interface.

Step 13	**ipv6 dhcp relay** {**destination** | **source-interface** | **trust**}
	Example:
	`Device(config-if)# ipv6 dhcp relay destination 2001:DB8:2::2`
	`Device(config-if)# ipv6 dhcp relay source-interface Vlan50`
	`Device(config-if)# ipv6 dhcp relay trust`
		Configures Dynamic Host Configuration Protocol (DHCP) for IPv6 relay service on the interface.
		• **destination**: Specifies a destination address to which client messages are forwarded.
		• **source-interface**: Specifies an interface to use as the source when relaying messages received on this interface.
		• **trust**: Specifies the interface to be trusted to process relay-replies.

Step 14	**no lisp mobility liveness test** **Example:** `Device(config-if)# no lisp mobility liveness test`	Removes mobility liveness settings discovered on this interface.

Step 15	**lisp mobility** *dynamic-eid-name*
	Example:
	For a user-defined VRF:
	`Device(config-if)# lisp mobility AVlan50-IPV4`
	`Device(config-if)# lisp mobility AVlan50-IPV6`
	For a Default Instance:
		Specifies the name of the LISP dynamic-EID policy to apply to this interface.

	Command or Action	Purpose
	`Device(config-if)# lisp mobility AVlan91-IPV4`	

Step 16	**end**
	Example:
	`Device(config-if)# end`
		Returns to privileged EXEC mode.

Configure DHCP Options and Snooping

To configure DHCP options and snooping on a fabric edge node, perform this task:

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	**ip dhcp relay information option** **Example:** `Device(config)# ip dhcp relay information option`	Enables the system to insert the DHCP relay agent information option (option-82 field) in forwarded BOOTREQUEST messages to a DHCP server.
Step 4	**ip dhcp snooping vlan** {*vlan id* | *vlan range*} **Example:** `Device(config)# ip dhcp snooping vlan 50,91`	Enables DHCP snooping on a VLAN or VLAN range. It also enables the DT-PROGRAMMATIC policy that supports onboarding of DHCPv4 hosts. DT-PROGRMMATIC policy enables device-tracking for the IEEE 802.1X, web authentication, Cisco TrustSec, and IPSG features.
Step 5	**ip dhcp snooping** **Example:** `Device(config)# ip dhcp snooping`	Enables DHCP snooping globally.

Step 6	**end**
	Example:
	`Device(config)# end`
		Returns to privileged EXEC mode.

Configure LISP

To configure LISP on a fabric edge node, perform this task:

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	**router lisp** **Example:** `Device(config)# router lisp`	Enters LISP configuration mode.
Step 4	**locator-table default** **Example:** `Device(config-router-lisp)# locator-table default`	Selects the default (global) routing table for association with the routing locator address space.
Step 5	**locator-set** *loc-set-name* **Example:** `Device(config-router-lisp)# locator-set rloc_set2`	Specifies a locator-set and enters the locator-set configuration mode.
Step 6	**ipv4-interface Loopback** *loopback-interface-id* **priority** *locator-priority* **weight** *locator-weight* **Example:** `Device(config-router-lisp-locator-set)# IPv4-interface Loopback0 priority 10 weight 10`	Configures the loopback IP address to ensure the device is reachable.
Step 7	**exit-locator-set** **Example:** `Device(config-router-lisp-locator-set)# exit-locator-set`	Exits locator-set configuration mode, and enters LISP configuration mode.
Step 8	**locator default-set** *rloc-set-name* **Example:** `Device(config-router-lisp)# locator default-set rloc_set2`	Marks a locator-set as default.

	Command or Action	Purpose

Step 9	**service** { **ipv4** | **ipv6** }
	Example:
	`Device(config-router-lisp)# service ipv4`
	`Device(config-router-lisp)# service ipv6`
		Enables network services on the default instance.
		service ipv4: Enables Layer 3 network services for the IPv4 address family.
		service ipv6: Enables Layer 3 network services for the IPv6 address family.

Step 10	**encapsulation vxlan**
	Example:
	`Device(config-router-lisp-serv-ipv4)#`
	`encapsulation vxlan`
	`Device(config-router-lisp-serv-ipv6)#`
	`encapsulation vxlan`
		Specifies VXLAN-based encapsulation.

Step 11	**itr map-resolver** *map-address*
	Example:
	`Device(config-router-lisp-serv-ipv4)#`
	`itr map-resolver 172.16.1.66`
	`Device(config-router-lisp-serv-ipv4)#`
	`itr map-resolver 172.16.1.67`
	`Device(config-router-lisp-serv-ipv6)#`
	`itr map-resolver 172.16.1.66`
	`Device(config-router-lisp-serv-ipv6)#`
	`itr map-resolver 172.16.1.67`
		Configures map-resolver address for sending map requests, on the Ingress Tunnel Router (ITR).
		A control plane node is the LISP map resolver. Specify the IP address of the Loopback 0 interface on control plane node as the *map-address*. If your fabric site has more than one control plane nodes, execute this command for each of the *map-address* (control plane nodes).

Step 12	**etr map-server** *map-server-address* **key** *authentication-key*
	Example:
	`Device(config-router-lisp-serv-ipv4)#`
	`etr map-server 172.16.1.66 key some-key`
	`Device(config-router-lisp-serv-ipv4)#`
	`etr map-server 172.16.1.67 key auth-key`
	`Device(config-router-lisp-serv-ipv6)#`
	`etr map-server 172.16.1.66 key some-key`
	`Device(config-router-lisp-serv-ipv6)#`
	`etr map-server 172.16.1.67 key auth-key`
		Configures a map server to be used by the Egress Tunnel Router (ETR), and specifies the authentication key to be used with this map server.
		Note Ensure that you use the same *authentication-key* that was configured on the control plane node.
		A control plane node is the LISP map server. Specify the IP address of the Loopback 0 interface on control plane node as the *map-server-address*. If your fabric site has more than one control plane node, execute this command for each of the *map-server-address* (control plane nodes).

Step 13	**etr map-server** *map-server-address* **proxy-reply**
	Example:
	`Device(config-router-lisp-serv-ipv4)#`
	`etr map-server 172.16.1.66 proxy-reply`
		Configures a map server to be used by the Egress Tunnel Router (ETR), and specifies that the map server answers the map-requests on behalf the ETR.

	Command or Action	Purpose

	`Device(config-router-lisp-serv-ipv4)#` **`etr map-server 172.16.1.67 proxy-reply`**
	`Device(config-router-lisp-serv-ipv6)#` **`etr map-server 172.16.1.66 proxy-reply`** `Device(config-router-lisp-serv-ipv6)#` **`etr map-server 172.16.1.67 proxy-reply`**
		A control plane node is the LISP map server. Specify the IP address of the Loopback 0 interface on control plane node as the *map-server-address*. If your fabric site has more than one control plane node, execute this command for each of the *map-server-address* (control plane nodes).

Step 14	**etr** **Example:** **Example:** `Device(config-router-lisp-serv-ipv4)#` **`etr`** `Device(config-router-lisp-serv-ipv6)#` **`etr`**	Configures the device as an Egress Tunnel Router (ETR).

Step 15	**sgt**
	Example:
	`Device(config-router-lisp-serv-ipv4)#` **`sgt`**
	`Device(config-router-lisp-serv-ipv6)#` **`sgt`**
		Enables the Security Group Tag (SGT) function for SGT tag propagation.

Step 16	**proxy-itr** *address*
	Example:
	`Device(config-router-lisp-serv-ipv4)#` **`proxy-itr 172.16.1.68`**
	`Device(config-router-lisp-serv-ipv6)#` **`proxy-itr 172.16.1.68`**
		Configures the device to act as a Locator/ID Separation Protocol (LISP) Proxy Ingress Tunnel Router (PITR).
		For *address*, specify the Loopback 0 IP address of this device.

Step 17	Do one of the following: • **exit-service-ipv4** • **exit-service-ipv6** **Example:** `Device(config-router-lisp-serv-ipv4)#` **`exit-service-ipv4`** `Device(config-router-lisp-serv-ipv6)#` **`exit-service-ipv6`**	Exits service configuration mode, and enters LISP configuration mode. Use the appropriate command, depending on which service mode you are exiting from (IPv4 or IPv6 service mode).
Step 18	**service ethernet** **Example:** `Device(config-router-lisp)#` **`service ethernet`**	Enables Layer 2 network services.

	Command or Action	Purpose

Step 19	**itr map-resolver** *map-address*
	Example:
	`Device(config-router-lisp-serv-eth)#`
	`itr map-resolver 172.16.1.66`
	`Device(config-router-lisp-serv-eth)#`
	`itr map-resolver 172.16.1.67`
		Configures map-resolver address for sending map requests, on the Ingress Tunnel Router (ITR).

Step 20	**itr**
	Example:
	`Device(config-router-lisp-serv-eth)#`
	`itr`
		Configures the device as an Ingress Tunnel Router (ITR).

Step 21	**etr map-server** *map-server-address* **key** [**0**|**6** | **7** } *authentication-key*
	Example:
	`Device(config-router-lisp-serv-eth)#`
	`etr map-server 172.16.1.66 key some-key`
	`Device(config-router-lisp-serv-eth)#`
	`etr map-server 172.16.1.67 key auth-key`
		Configures a map server to be used by the Egress Tunnel Router (ETR), and specifies the key type.
		Key type 0 indicates that password is entered as clear text.
		Key type 6 indicates that password is in the AES encrypted form.
		Key type 7 indicates that password is a weak encrypted one.
		The map server and ETR must be configured with matching passwords for the map-registration process to successfully complete. The map server must be preconfigured with the EID prefixes that match the EID-prefixes configured on this ETR using the **database-mapping** command, and a password matching the one provided with the **key** keyword on this ETR.
		Note Ensure that you use the same *authentication-key* that was configured on the control plane node.
		Specify the IP address of the Loopback 0 interface on control plane node as the *map-server-address*. If your fabric site has more than one control plane node, execute this command for each of the *map-server-address* (control plane nodes).

Step 22	**etr map-server** *map-server-address* **proxy-reply**	Configures a map server to be used by the Egress Tunnel Router (ETR), and specifies

	Command or Action	Purpose

	Example:
	`Device(config-router-lisp-serv-eth)#`
	`etr map-server 172.16.1.66 proxy-reply`
	`Device(config-router-lisp-serv-eth)#`
	`etr map-server 172.16.1.67 proxy-reply`
		that the map server answers the map-requests on behalf the ETR.
		Specify the IP address of the Loopback 0 interface on control plane node as the *map-server-address*. If your fabric site has more than one control plane node, execute this command for each of the *map-server-address* (control plane nodes).

Step 23	**etr**
	Example:
	`Device(config-router-lisp-serv-eth)#`
	`etr`
		Configures the device as an Egress Tunnel Router (ETR).

Step 24	**exit-service-ethernet**
	Example:
	`Device(config-router-lisp-serv-eth)#`
	`exit-service-ethernet`
		Exits service configuration mode, and enters LISP configuration mode.

Step 25	**ipv4 locator reachability minimum-mask-length** *length* **Example:** `Device(config-router-lisp)#` **`ipv4 locator reachability minimum-mask-length 32`**	Specifies the shortest mask prefix to accept when looking up a remote RLOC in the RIB. LISP checks the host reachability from the routing locator.
Step 26	**ipv4 source-locator** *interface-number* **Example:** `Device(config-router-lisp)#` **`ipv4 source-locator loopback0`**	Configures the source locator for the outbound LISP packets. Set the loopback interface as the source locator.

Step 27	**exit-router-lisp**
	Example:
	`Device(config-router-lisp)#`
	`exit-router-lisp`
		Exits LISP configuration mode, and enters global configuration mode.

Step 28	**end**
	Example:
	`Device(config)#` **`end`**
		Returns to privileged EXEC mode.

Step 29	**show lisp locator-set**
	Example:
	`Device#` **`show lisp locator-set`**
	`LISP Locator-set information:`
	
	`172.16.1.68, local, reachable, loopback`
		Displays information about the Locator Set that is configured on the device.

Configure Layer 3 VNI and Segment for Default Instance

A default instance connects network infrastructure elements like Access Points and Layer 2 switches to the fabric access layer. To configure Layer 3 VNI for the default instance, perform this task:

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	**router lisp** **Example:** `Device(config)# router lisp`	Enters LISP configuration mode.
Step 4	**instance-id** *id* **Example:** `Device(config-router-lisp)# instance-id 4097`	Specifies the instance ID.
Step 5	**remote-rloc-probe on-route-change** **Example:** `Device(config-router-lisp-inst)# remote-rloc-probe on-route-change`	Configures parameters for probing of remote local routing locators (RLOCs).
Step 6	**dynamic-eid** *eid-name* **Example:** `Device(config-router-lisp-inst)# dynamic-eid AVlan91-IPV4`	Creates a dynamic Endpoint Identifier (EID) policy and enters the dynamic-eid configuration mode on the fabric edge node. To configure LISP host mobility, you must create a dynamic-eid policy that can be referenced by the **lisp mobility** *dynamic-eid-name* interface command. Hence the *eid-name* that is associated with **dynamic-eid** command should be the same as *dynamic-eid-name* that is used to configure LISP mobility. For the *dynamic-eid-name*, refer to the lisp mobility configuration step of the Configure an SVI Interface procedure.
Step 7	**database-mapping** *eid-prefix/prefix-length* **locator-set** *RLOC_name* **Example:**	Configures an IPv4 endpoint identifier-to-routing locator (EID-to-RLOC) mapping relationship and an associated traffic policy for LISP.

<table>
<tr><th></th><th>Command or Action</th><th>Purpose</th></tr>
<tr><td></td><td><pre>Device(config-router-lisp-inst-dynamic-eid)#
 database-mapping 10.91.1.0/24
locator-set rloc_set2</pre></td><td></td></tr>
<tr><td><b>Step 8</b></td><td><b>exit-dynamic-eid</b>
<b>Example:</b><pre>Device(config-router-lisp-inst-dynamic-eid)#
 exit-dynamic-eid</pre></td><td>Exits dynamic-eid configuration mode, and enters LISP instance configuration mode.</td></tr>
<tr><td><b>Step 9</b></td><td><b>service ipv4</b>
<b>Example:</b><pre>Device(config-router-lisp-inst)# service
 ipv4</pre></td><td>Enables Layer 3 network services for the IPv4 address family.</td></tr>
<tr><td><b>Step 10</b></td><td><b>eid-table default</b>
<b>Example:</b><pre>Device(config-router-lisp-inst-serv-ipv4)#
 eid-table default</pre></td><td>Configures the default (global) routing table for association with the configured instance-service.</td></tr>
<tr><td><b>Step 11</b></td><td><b>exit-service-ipv4</b>
<b>Example:</b><pre>Device(config-router-lisp-inst-serv-ipv4)#
 exit-service-ipv4</pre></td><td>Exits IP service configuration mode, and enters LISP instance configuration mode.</td></tr>
<tr><td><b>Step 12</b></td><td><b>exit-instance-id</b>
<b>Example:</b><pre>Device(config-router-lisp-inst)#
exit-instance-id</pre></td><td>Exits instance configuration mode, and enters LISP configuration mode.</td></tr>
<tr><td><b>Step 13</b></td><td><b>end</b>
<b>Example:</b><pre>Device(config-router-lisp)# end</pre></td><td>Returns to privileged EXEC mode.</td></tr>
<tr><td><b>Step 14</b></td><td><b>show lisp session</b>
<b>Example:</b><pre>Device# show lisp session
Sessions for VRF default, total: 2,
established: 1
Peer   State
    Up/Down   In/Out   Users
172.16.1.66:4342   Up
      02:21:53   20/9   14
Device#</pre></td><td>Displays a summary of the LISP sessions that this fabric edge node has set up with the control plane node.</td></tr>
</table>

Configure Layer 2 VNI and Segment for Default Instance

A Default Instance connects network infrastructure elements like Access Points and Layer-2 switches to the fabric access layer. To configure Layer 2 VNI for the Default Instance, perform this task:

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	**router lisp** **Example:** `Device(config)# router lisp`	Enters LISP configuration mode.
Step 4	**instance-id** *id* **Example:** `Device(config-router-lisp)# instance-id 8194`	Specifies the instance ID. Ensure that the Layer 2 VNI ID is different from the Layer 3 VNI ID that you have configured in the earlier task.
Step 5	**remote-rloc-probe on-route-change** **Example:** `Device(config-router-lisp-inst)# remote-rloc-probe on-route-change`	Configures parameters for probing of remote local routing locators (RLOCs).
Step 6	**service ethernet** **Example:** `Device(config-router-lisp-inst)# service ethernet`	Enables Layer 2 network services.
Step 7	**eid-table vlan** *vlan-id* **Example:** `Device(config-router-lisp-inst-serv-ethernet)# eid-table vlan 91`	Configures the specified VLAN table for association with the configured instance.
Step 8	**database-mapping** *eid-prefix/prefix-length* **locator-set** *RLOC_name* **Example:** `Device(config-router-lisp-inst-serv-ethernet-eid-table)# database-mapping mac locator-set rloc_set2`	Configures an IPv4 endpoint identifier-to-routing locator (EID-to-RLOC) mapping relationship and an associated traffic policy for LISP.
Step 9	**exit-service-ethernet** **Example:** `Device(config-router-lisp-inst-serv-ethernet)# exit-service-ethernet`	Exits service Ethernet configuration mode, and enters LISP instance configuration mode.

	Command or Action	Purpose
Step 10	**exit-instance-id** **Example:** `Device(config-router-lisp-inst)# exit-instance-id`	Exits instance configuration mode, and enters LISP configuration mode.

Step 11	**end**
	Example:
	`Device(config-router-lisp)# end`
		Returns to privileged EXEC mode.

Configure Layer 3 VNI and Segment for User-Defined VRF

To configure a Layer 3 VNI for user-defined VRF, perform this task:

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	**router lisp** **Example:** `Device(config)# router lisp`	Enters LISP configuration mode.
Step 4	**instance-id** *id* **Example:** `Device(config-router-lisp)# instance-id 4099`	Specifies the instance ID.
Step 5	**remote-rloc-probe on-route-change** **Example:** `Device(config-router-lisp-inst)# remote-rloc-probe on-route-change`	Configures parameters for probing of remote local routing locators (RLOCs).
Step 6	**dynamic-eid** *eid-name* **Example:** `Device(config-router-lisp-inst)# dynamic-eid AVlan50-IPV4`	Creates a dynamic End Point Identifier (EID) policy, and enters the dynamic-eid configuration mode on an xTR.

	Command or Action	Purpose
Step 7	**database-mapping** *eid-prefix/prefix-length* **locator-set** *RLOC_name* **Example:** `Device(config-router-lisp-inst-dynamic-eid)# database-mapping 10.50.1.0/24 locator-set rloc_set2`	Configures an IPv4 endpoint identifier-to-routing locator (EID-to-RLOC) mapping relationship.
Step 8	**exit-dynamic-eid** **Example:** `Device(config-router-lisp-inst-dynamic-eid)# exit-dynamic-eid`	Exits dynamic-eid configuration mode, and enters LISP instance configuration mode.
Step 9	**dynamic-eid** *eid-name* **Example:** `Device(config-router-lisp-inst)# dynamic-eid AVlan50-IPV6`	Creates a dynamic Endpoint Identifier (EID) policy and enters the dynamic-eid configuration mode on a fabric edge node. To configure LISP host mobility, you must create a dynamic-eid policy that can be referenced by the **lisp mobility** *dynamic-eid-name* interface command. Hence the *eid-name* that is associated with **dynamic-eid** command should be the same as *dynamic-eid-name* that is used to configure LISP mobility. For the *dynamic-eid-name*, refer to the lisp mobility configuration step of the Configure an SVI Interface procedure.
Step 10	**database-mapping** *eid-prefix/prefix-length* **locator-set** *RLOC_name* **Example:** `Device(config-router-lisp-inst-dynamic-eid)# database-mapping 2001:DB8:2050::/64 locator-set rloc_set2`	Configures an IPv6 endpoint identifier-to-routing locator (EID-to-RLOC) mapping relationship.
Step 11	**exit-dynamic-eid** **Example:** `Device(config-router-lisp-inst-dynamic-eid)# exit-dynamic-eid`	Exits dynamic-eid configuration mode, and enters LISP instance configuration mode.
Step 12	**service ipv4** **Example:** `Device(config-router-lisp-inst)# service ipv4`	Enables Layer 3 network services for the IPv4 address family.
Step 13	**eid-table vrf** *vrf-name* **Example:** `Device(config-router-lisp-inst-serv-ipv4)# eid-table vrf VN3`	Configures the VRF table for association with the configured instance-service.

	Command or Action	Purpose
Step 14	**map-cache** *address* **map-request** **Example:** `Device(config-router-lisp-inst-serv-ipv4)# map-cache 0.0.0.0/0 map-request`	Sends map-request for LISP destination IPv4 EID.
Step 15	**exit-service-ipv4** **Example:** `Device(config-router-lisp-inst-serv-ipv4)# exit-service-ipv4`	Exits service IPv4 configuration mode, and enters LISP instance configuration mode.
Step 16	**service ipv6** **Example:** `Device(config-router-lisp-inst)# service ipv6`	Enables Layer 3 network services for the IPv6 address family.
Step 17	**eid-table vrf** *vrf-name* **Example:** `Device(config-router-lisp-inst-serv-ipv6)# eid-table vrf VN3`	Configures the VRF table for association with the configured instance-service.
Step 18	**map-cache** *address* **map-request** **Example:** `Device(config-router-lisp-inst-serv-ipv6)# map-cache ::/0 map-request`	Sends map-request for LISP destination IPv6 EID.
Step 19	**exit-service-ipv6** **Example:** `Device(config-router-lisp-inst-serv-ipv6)# exit-service-ipv6`	Exits service IPv6 configuration mode, and enters LISP instance configuration mode.
Step 20	**exit-instance-id** **Example:** `Device(config-router-lisp-inst)# exit-instance-id`	Exits instance configuration mode, and enters LISP configuration mode.

Step 21	**end**
	Example:
	`Device(config-router-lisp)# end`
		Returns to privileged EXEC mode.

Step 22	**show ip route vrf** *vrf-name*
	Example:
	`Device# show ip route vrf VN3`
	
	`Routing Table: VN3`
	`Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP`
	`       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area`
		Displays the routing table on the device, for a specified VRF.

	Command or Action	Purpose

	`       N1 - OSPF NSSA external type 1,`
	` N2 - OSPF NSSA external type 2`
	`       E1 - OSPF external type 1, E2 -`
	` OSPF external type 2, m - OMP`
	`       n - NAT, Ni - NAT inside, No -`
	`NAT outside, Nd - NAT DIA`
	`       i - IS-IS, su - IS-IS summary,`
	`L1 - IS-IS level-1, L2 - IS-IS level-2`
	`       ia - IS-IS inter area, * -`
	`candidate default, U - per-user static`
	` route`
	`       H - NHRP, G - NHRP registered,`
	`g - NHRP registration summary`
	`       o - ODR, P - periodic downloaded`
	` static route, l - LISP`
	`       a - application route`
	`       + - replicated route, % - next`
	`hop override, p - overrides from PfR`
	`       & - replicated local route`
	`overrides by connected`
	
	`Gateway of last resort is not set`
	
	`      10.0.0.0/8 is variably subnetted,`
	` 2 subnets, 2 masks`
	`C        10.50.1.0/24 is directly`
	`connected, Vlan50`
	`L        10.50.1.1/32 is directly`
	`connected, Vlan50`
	`Device#`	

Configure Layer 2 VNI for VLANs in User-Defined VRF

To configure Layer 2 VNI for VLANs in user-defined virtual routing and forwarding instance on a fabric edge node, perform this task:

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	**router lisp** **Example:** `Device(config)# `**`router lisp`**	Enters LISP configuration mode.
Step 4	**instance-id** *id*	Specifies the instance ID.

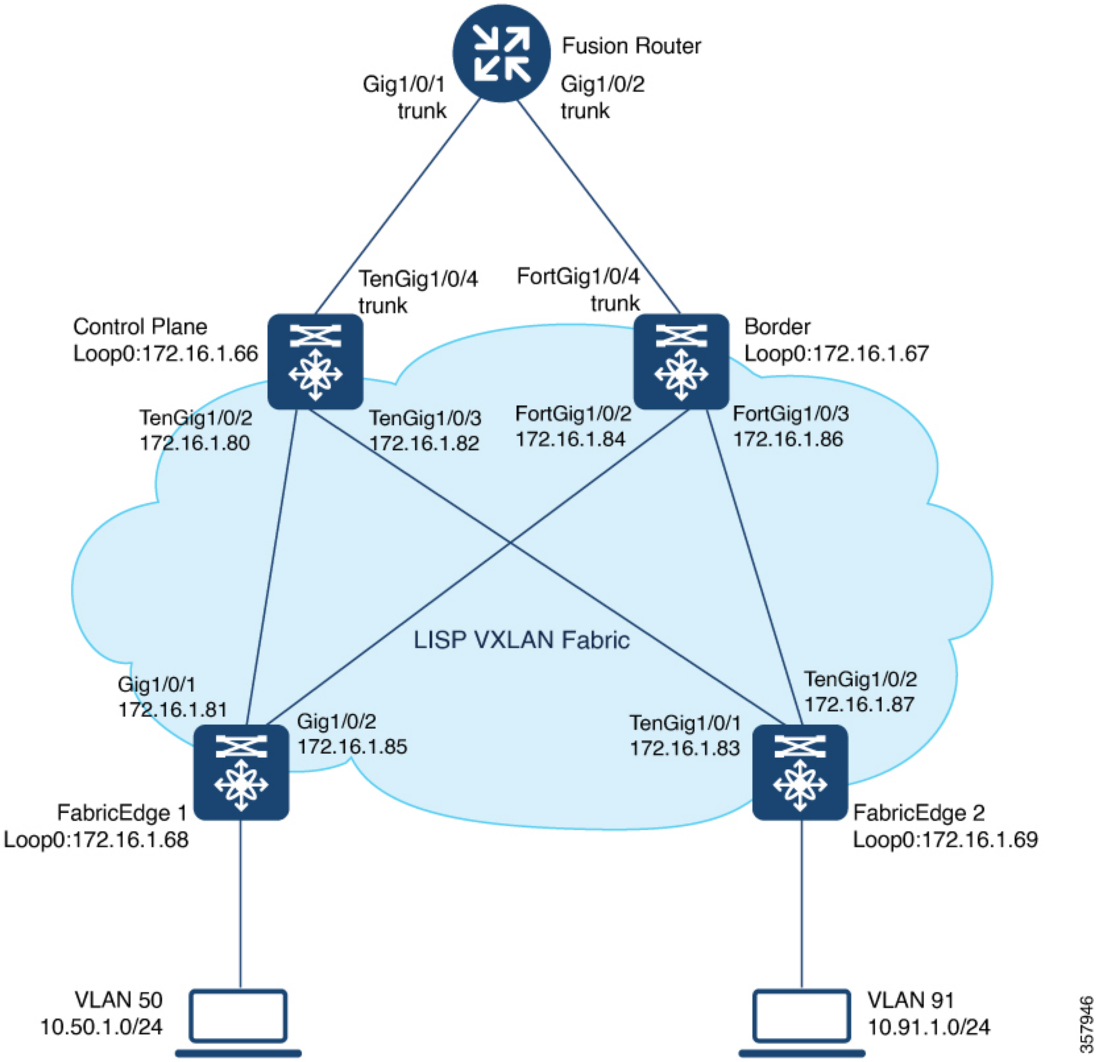
	Command or Action	Purpose
	Example: `Device(config-router-lisp)#` **`instance-id 8197`**	Ensure that each Layer 2 VNI ID is unique and is different from the Layer 3 VNI IDs that you have configured in the earlier task.
Step 5	**remote-rloc-probe on-route-change** **Example:** `Device(config-router-lisp-inst)#` **`remote-rloc-probe on-route-change`**	Configures parameters for probing of remote local routing locators (RLOCs).
Step 6	**service ethernet** **Example:** `Device(config-router-lisp-inst)#` **`service ethernet`**	Enables Layer 2 network services.
Step 7	**eid-table vlan** *vlan-id* **Example:** `Device(config-router-lisp-inst-serv-ethernet)#` **`eid-table vlan 50`**	Configures the specified VLAN table for association with the configured instance.
Step 8	**database-mapping** *eid-prefix/prefix-length* **locator-set** *RLOC_name* **Example:** `Device(config-router-lisp-inst-serv-ethernet-eid-table)#` **`database-mapping mac locator-set rloc_set2`**	Configures an IPv4 endpoint identifier-to-routing locator (EID-to-RLOC) mapping relationship and an associated traffic policy for LISP.
Step 9	**exit** **Example:** `Device(config-router-lisp-inst-serv-ethernet-eid-table)#` **`exit`**	Exits EID table configuration mode.
Step 10	**exit-service-ethernet** **Example:** `Device(config-router-lisp-inst-serv-ethernet)#` **`exit-service-ethernet`**	Exits service Ethernet configuration mode, and enters LISP configuration mode.
Step 11	**exit-instance-id** **Example:** `Device(config-router-lisp-inst)#` **`exit-instance-id`**	Exits instance configuration mode, and enters LISP configuration mode.

Step 12	**end**
	Example:
	`Device(config-router-lisp)#` **`end`**
		Returns to privileged EXEC mode.

Configuration Example for LISP VXLAN Fabric Edge Node

This example shows a sample configuration for a fabric edge node in the Figure 6: LISP VXLAN Fabric Topology below.

Figure 6: LISP VXLAN Fabric Topology



EN

```
vrf definition VN3
 !
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 exit-address-family
!
```

```
ip dhcp relay information option
ip dhcp snooping vlan 50,91
ip dhcp snooping
!
device-tracking policy IPDT_POLICY
 tracking enable
!
interface GigabitEthernet1/0/3
 device-tracking attach-policy IPDT_POLICY
!
vlan configuration 50
 ipv6 nd raguard
 ipv6 dhcp guard
!
vlan 50
 name AVlan50
!
vlan 91
 name AVlan91
!
interface Vlan50
 description server1
 mac-address 0000.0c9f.f18e
 vrf forwarding VN3
 ip address 10.50.1.1 255.255.255.0
 ip helper-address 172.16.2.2
 no ip redirects
 ipv6 address 2001:DB8:2050::1/64
 ipv6 enable
 ipv6 nd dad attempts 0
 ipv6 nd prefix 2001:DB8:2050::/64 2592000 604800 no-autoconfig
 ipv6 nd managed-config-flag
 ipv6 nd other-config-flag
 ipv6 nd router-preference High
 ipv6 dhcp relay destination 2001:DB8:2::2
 ipv6 dhcp relay source-interface Vlan50
 ipv6 dhcp relay trust
 no lisp mobility liveness test
 lisp mobility AVlan50-IPV4
 lisp mobility AVlan50-IPV6
!

interface Vlan91
 description server2
 mac-address 0000.0c9f.f984
 ip address 10.91.1.1 255.255.255.0
 ip helper-address 172.16.2.2
 no ip redirects
 no lisp mobility liveness test
 lisp mobility AVlan91-IPV4
!


router lisp
 locator-table default
 locator-set rloc_set2
  IPv4-interface Loopback0 priority 10 weight 10
  exit-locator-set
 !
 locator default-set rloc_set2
 service ipv4
  encapsulation vxlan
  itr map-resolver 172.16.1.66
  itr map-resolver 172.16.1.67
  etr map-server 172.16.1.66 key some-key
  etr map-server 172.16.1.66 proxy-reply
  etr map-server 172.16.1.67 key auth-key
  etr map-server 172.16.1.67 proxy-reply
  etr
  sgt
  proxy-itr 172.16.1.68
  exit-service-ipv4
 !
 service ipv6
  encapsulation vxlan
  itr map-resolver 172.16.1.66
  itr map-resolver 172.16.1.67
  etr map-server 172.16.1.66 key some-key
  etr map-server 172.16.1.66 proxy-reply
  etr map-server 172.16.1.67 key auth-key
  etr map-server 172.16.1.67 proxy-reply
  etr
  sgt
  proxy-itr 172.16.1.68
  exit-service-ipv6
 !
 service ethernet
  itr map-resolver 172.16.1.66
  itr map-resolver 172.16.1.67
  itr
  etr map-server 172.16.1.66 key some-key
  etr map-server 172.16.1.66 proxy-reply
  etr map-server 172.16.1.67 key auth-key
  etr map-server 172.16.1.67 proxy-reply
  etr
  exit-service-ethernet
 !

 instance-id 4097
  remote-rloc-probe on-route-change
  dynamic-eid AVlan91-IPV4
   database-mapping 10.91.1.0/24 locator-set rloc_set2
   exit-dynamic-eid
  !
  service ipv4
   eid-table default
   exit-service-ipv4
  !
  service ipv6
   eid-table default
   exit-service-ipv6
  !
  exit-instance-id
 !
 instance-id 4099
  remote-rloc-probe on-route-change
  dynamic-eid AVlan50-IPV4
   database-mapping 10.50.1.0/24 locator-set rloc_set2
   exit-dynamic-eid
  !
  dynamic-eid AVlan50-IPV6
   database-mapping 2001:DB8:2050::/64 locator-set rloc_set2
   exit-dynamic-eid
  !
  service ipv4
   eid-table vrf VN3
   map-cache 0.0.0.0/0 map-request
   exit-service-ipv4
  !
```

```
  service ipv6
   eid-table vrf VN3
   map-cache ::/0 map-request
   exit-service-ipv6
  !
  exit-instance-id
 !
 !

 instance-id 8194
  remote-rloc-probe on-route-change
  service ethernet
   eid-table vlan 91
   database-mapping mac locator-set rloc_set2
   exit-service-ethernet
  !
  exit-instance-id
 !
 !

 instance-id 8197
  remote-rloc-probe on-route-change
  service ethernet
   eid-table vlan 50
   database-mapping mac locator-set rloc_set2
   exit-service-ethernet
  !
  exit-instance-id
 !
 !
 ipv4 locator reachability minimum-mask-length 32
 ipv4 source-locator Loopback0
 exit-router-lisp
!
```

Verify the Configuration of Fabric Edge Node

This section provides sample outputs for the **show** commands on the fabric edge nodes in the topology shown Figure 6: LISP VXLAN Fabric Topology.

View a summary of the LISP sessions that are created on the edge node:

```
FabricEdge# show lisp session

Sessions for VRF default, total: 2, established: 2
Peer                           State      Up/Down        In/Out    Users
172.16.1.66:4342                 Up         1d04h          27/9      14
172.16.1.67:4342                 Up         1d03h          19/9      14
FabricEdge#
```

View the LISP session with the Control Plane Node (172.16.1.66) :

```
FabricEdge# show lisp session 172.16.1.66 port 4342
Peer address:     172.16.1.66:4342
Local address:    172.16.1.69:27785
Session Type:     Active
Session State:    Up (1d04h)
Messages in/out:  27/9
Bytes in/out:     1666/276
Fatal errors:     0
Rcvd unsupported: 0
Rcvd invalid VRF: 0
Rcvd override:    0
Rcvd malformed:   0
Sent deferred:    0
SSO redundancy:   N/A
Auth Type:        None

Accepting Users:  0
Users:            14
  Type                      ID                                      In/Out    State
  Pubsub subscriber         lisp 0 IID 4097 AFI IPv4                 1/0      Idle
  Pubsub subscriber         lisp 0 IID 4097 AFI IPv6                 1/0      Idle
  Pubsub subscriber         lisp 0 IID 4099 AFI IPv4                 1/0      Idle
  Pubsub subscriber         lisp 0 IID 4099 AFI IPv6                 1/0      Idle
  Pubsub subscriber         lisp 0 IID 8194 AFI MAC                  2/0      Idle
  Pubsub subscriber         lisp 0 IID 8197 AFI MAC                  2/0      Idle
  Capability Exchange       N/A                                      1/1      waiting
  ETR Reliable Registration lisp 0 IID 4097 AFI IPv4                 0/1      TCP
  ETR Reliable Registration lisp 0 IID 4097 AFI IPv6                 0/1      TCP
  ETR Reliable Registration lisp 0 IID 4099 AFI IPv4                 0/1      TCP
  ETR Reliable Registration lisp 0 IID 4099 AFI IPv6                 0/1      TCP
  ETR Reliable Registration lisp 0 IID 8194 AFI MAC                  0/1      TCP
  ETR Reliable Registration lisp 0 IID 8197 AFI MAC                  0/1      TCP
  ETR Reliable Registration lisp 0 IID 16777214 AFI IPv4            13/2      TCP
FabricEdge#
```

View the Locator set information:

```
FabricEdge# show lisp locator-set
LISP Locator-set information:

172.16.1.68, local, reachable, loopback
```

View the dynamic interfaces that are created after configuring LISP instances:

```
FabricEdge# show ip interface brief | i LISP
L2LISP0                172.16.1.68       YES unset  up                    up
L2LISP0.8194           172.16.1.68       YES unset  up                    up
L2LISP0.8197           172.16.1.68       YES unset  up                    up
LISP0                  unassigned        YES unset  up                    up
LISP0.4097             172.16.1.68       YES unset  up                    up
LISP0.4099             10.50.1.1         YES unset  up                    up
FabricEdge#
```

View the IPv4 map-cache entries:

```
FabricEdge# show lisp instance-id 4099 ipv4 map-cache
LISP IPv4 Mapping Cache for LISP 0 EID-table vrf VN3 (IID 4099), 2 entries

0.0.0.0/0, uptime: 18:03:23, expires: 00:12:10, via map-reply, unknown-eid-forward
action: send-map-request + Encapsulating to proxy ETR
  PETR       Uptime    State      Pri/Wgt     Encap-IID  Metric
  172.16.1.67  18:03:23  up          10/10        -         0
10.50.1.0/24, uptime: 19:59:51, expires: never, via dynamic-EID, send-map-request
  Negative cache entry, action: send-map-request
```

View the LISP EID statistics related to packet encapsulations, decapsulations, map requests, map replies, map registers, and other LISP-related packets:

```
FabricEdge# show lisp service ipv4 statistics
LISP EID Statistics for all EID instances - last cleared: never
Control Packets:
  Map-Requests in/out:                              2/2465
    Map-Requests in (5 sec/1 min/5 min):            0/0/0
    Encapsulated Map-Requests in/out:               0/2465
    RLOC-probe Map-Requests in/out:                 2/0
    SMR-based Map-Requests in/out:                  2/0
    Extranet SMR cross-IID Map-Requests in:         0
    Map-Requests expired on-queue/no-reply          0/493
    Map-Resolver Map-Requests forwarded:            0
    Map-Server Map-Requests forwarded:              0
  Map-Reply records in/out:                         0/0
    Authoritative records in/out:                   0/0
    Non-authoritative records in/out:               0/0
    Negative records in/out:                        0/0
    RLOC-probe records in/out:                      0/0
    Map-Server Proxy-Reply records out:             0
  WLC Map-Subscribe records in/out:                 0/11
    Map-Subscribe failures in/out:                  0/0
  WLC Map-Unsubscribe records in/out:               0/0
    Map-Unsubscribe failures in/out:                0/0
  Map-Register records in/out:                      0/150
    Map-Registers in (5 sec/1 min/5 min):           0/0/0
    Map-Server AF disabled:                         0
    Not valid site eid prefix:                      0
    Authentication failures:                        0
    Disallowed locators:                            0
    Miscellaneous:                                  0
  WLC Map-Register records in/out:                  0/0
    WLC AP Map-Register in/out:                     0/0
    WLC Client Map-Register in/out:                 0/0
    WLC Map-Register failures in/out:               0/0
  Map-Notify records in/out:                        24/0
    Authentication failures:                        0
  WLC Map-Notify records in/out:                    0/0
    WLC AP Map-Notify in/out:                       0/0
    WLC Client Map-Notify in/out:                   0/0
    WLC Map-Notify failures in/out:                 0/0
  Publish-Subscribe in/out:
    Subscription Request records in/out:            0/0
      IID subscription requests in/out:             0/0
      Pub-refresh subscription requests in/out:     0/0
      Policy subscription requests in/out:          0/0
    Subscription Request failures in/out:           0/0
    Subscription Status records in/out:             0/0
      End of Publication records in/out:            0/0
      Subscription rejected records in/out:         0/0
      Subscription removed records in/out:          0/0
    Subscription Status failures in/out:            0/0
    Solicit Subscription records in/out:            21/0
    Solicit Subscription failures in/out:           0/0
    Publication records in/out:                     0/0
    Publication failures in/out:                    0/0
Errors:
  Mapping record TTL alerts:                        0
  Map-Request invalid source rloc drops:            0
  Map-Register invalid source rloc drops:           0
  DDT Requests failed:                              0
  DDT ITR Map-Requests dropped:                     0 (nonce-collision: 0, bad-xTR-nonce:
0)
Cache Related:
  Cache entries created/deleted:                    7/4
  NSF CEF replay entry count                        0
  Number of rejected EID-prefixes due to limit:     0
Forwarding:
  Number of data signals processed:                 0 (+ dropped 0)
  Number of reachability reports:                  0 (+ dropped 0)
  Number of SMR signals dropped:                   0
LISP RLOC Statistics - last cleared: never
Control Packets:
  RTR Map-Requests forwarded:                      0
  RTR Map-Notifies forwarded:                      0
  DDT-Map-Requests in/out:                         0/0
  DDT-Map-Referrals in/out:                        0/0
Errors:
  Map-Request format errors:                       0
  Map-Reply format errors:                         0
  Map-Referral format errors:                      0
LISP Miscellaneous Statistics - last cleared: never
Errors:
  Invalid IP version drops:                        0
  Invalid IP header drops:                         0
  Invalid IP proto field drops:                    0
  Invalid packet size drops:                       0
  Invalid LISP control port drops:                 0
  Invalid LISP checksum drops:                     0
  Unsupported LISP packet type drops:              0
  Unknown packet drops:                            0
FabricEdge#
```

View a summary of the IPv4 service instances on the fabric edge node:

```
FabricEdge# show lisp service ipv4 summary
Router-lisp ID:   0
Instance count:   5
Key: DB - Local EID Database entry count (@ - RLOC check pending
                                          * - RLOC consistency problem),
     DB no route - Local EID DB entries with no matching RIB route,
     Cache - Remote EID mapping cache size, IID - Instance ID,
     Role - Configured Role

                      Interface    DB  DB no  Cache Incom Cache
EID VRF name             (.IID)  size  route   size plete  Idle Role
default              LISP0.4097     1      0      1  0.0%  0.0% ETR-PITR
VN3                  LISP0.4099     1      0      2  0.0%  0.0% ETR-PITR

Number of eid-tables:                                2
Total number of database entries:                    2 (inactive 0)
Maximum database entries:                       214528
EID-tables with inconsistent locators:               0
Total number of map-cache entries:                   3
Maximum map-cache entries:                      214528
EID-tables with incomplete map-cache entries:        0
EID-tables pending map-cache update to FIB:          0
FabricEdge#
```

View the details of the routing table that is created when a Layer 3 VRF is configured:

```
FabricEdge# show ip route vrf VN3

Routing Table: VN3
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, m - OMP
       n - NAT, Ni - NAT inside, No - NAT outside, Nd - NAT DIA
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       H - NHRP, G - NHRP registered, g - NHRP registration summary
       o - ODR, P - periodic downloaded static route, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR
       & - replicated local route overrides by connected

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        10.50.1.0/24 is directly connected, Vlan50
L        10.50.1.1/32 is directly connected, Vlan50
FabricEdge#
```

CHAPTER 6

Configuring Wireless Support in a LISP VXLAN Fabric

A wireless network uses radio waves to connect the end points to the rest of the network. The main components of a wireless network infrastructure are the wireless Access Points (APs) and a Wireless Controller. An AP allows a wireless-capable device to connect to a wired network. A wireless controller controls and manages all the APs in the network. It is responsible for the AP image and configuration management, radio resource management, client session management and roaming, and all the other wireless control plane functions.

This chapter describes only the configurations that are required to support a wireless network in a LISP VXLAN Fabric. Before you proceed, we recommend that you look through the earlier chapters of this document for the functionality and configuration of a LISP VXLAN fabric.

Wireless Support in a LISP VXLAN Fabric

A LISP VXLAN fabric supports the wireless infrastructure in the these modes: Over-the-Top Centralized Wireless and Fabric-Enabled Wireless.

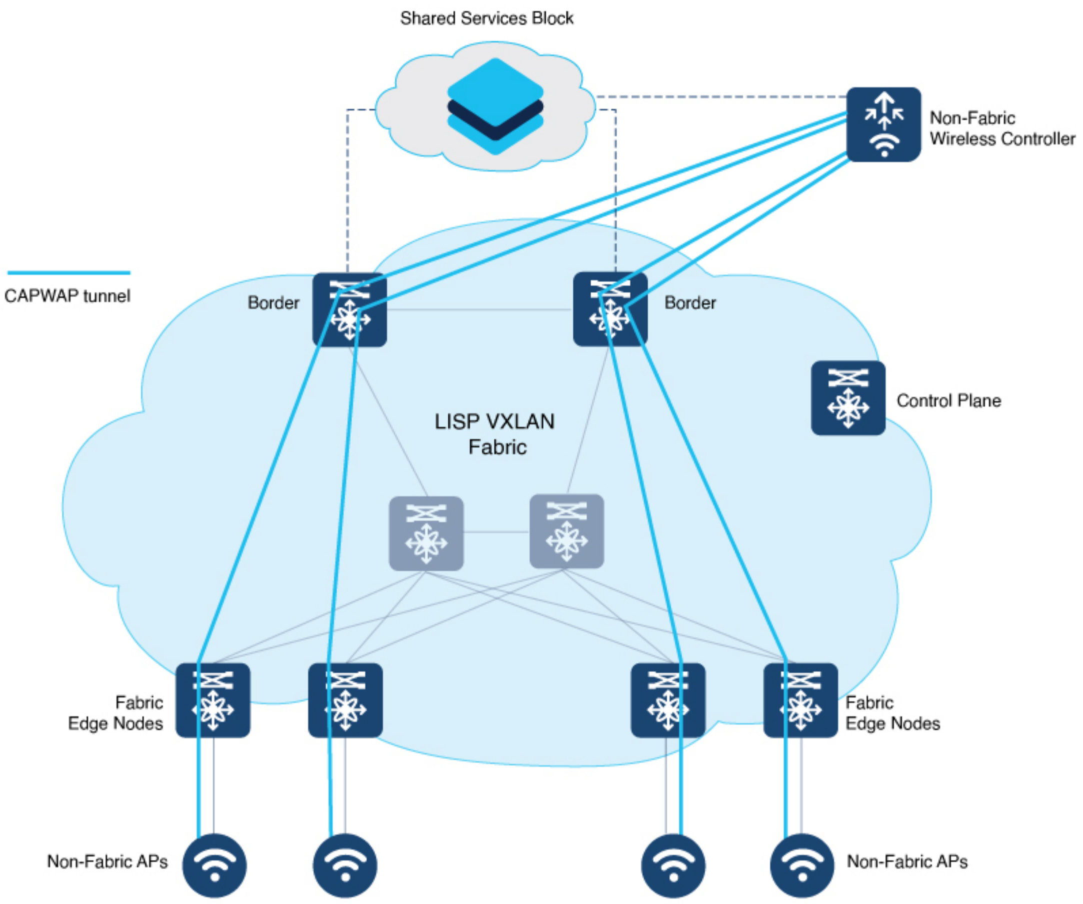
Over-the-Top Centralized Wireless

In an over-the-top (OTT) centralized wireless deployment, traditional wireless client traffic is encapsulated in Control and Provisioning of Wireless Access Points (CAPWAP) at the access point. The CAPWAP data is encapsulated in VXLAN at the fabric edge node, and forwarded to the fabric border node. At the border node, the VXLAN encapsulation is removed and the CAPWAP data traffic is forwarded to the wireless controller.

The CAPWAP tunnel between wireless controller and an AP traverses the campus backbone network, using the wired fabric as a transport medium.

OTT wireless deployment is suitable when you are migrating from a traditional network to a LISP VXLAN fabric network, wherein you might want to first migrate the wired infrastructure and plan wireless integration at a later time.

Figure 7: Over-the-Top Centralized Wireless Topology



Consider the following before you deploy OTT centralized wireless in your LISP VXLAN fabric.

- Wireless controller is located external to the fabric.
- APs are connected to the fabric edge node and are located in the default instance in the fabric overlay. The APs are registered with the control plane node as wired clients.
- After an AP gets an IP address from DHCP, it joins the wireless controller through CAPWAP tunnel. For information on AP connectivity to wireless controller, refer to *Cisco Wireless Controller Configuration Guide*.
- Wireless SSID is mapped to the VLAN or subnet at wireless controller using dynamic interfaces.

- Wireless clients are authenticated and onboarded by the wireless controller.
- A network device that is located upstream of the border advertises the wireless network to the fabric border.
- Communication between a wired host in the fabric and a wireless client outside fabric occurs through the fabric border.

Configuring OTT Centralized Wireless

This task describes only the fabric configurations that are required to enable OTT wireless, assuming that the wireless infrastructure is already functioning in the traditional way.

Before you begin

- Ensure that you have configured the control plane node, border node, and fabric edge node in a LISP VXLAN fabric for wired clients. For configuration information, refer to the earlier chapters in this document.
- Ensure that there is a specific subnet reachability in the underlay (global routing table) for the wireless controller subnet at the access layer. This is required for the access points to connect to the wireless controller.

Procedure

Step 1 On the fabric edge node, configure the switched virtual interface (SVI) for the AP VLAN.

Example:

```
interface Vlan92
 description For APs
 mac-address 0000.0c9f.ff39
 ip address 10.92.1.1 255.255.255.240
 no ip redirects
 no lisp mobility liveness test
 lisp mobility APVlan92-IPV4
end
!
```

The same SVI is present on every fabric edge node, with the same Virtual IP address and MAC address. This makes it a default gateway for all traffic from the APs.

Step 2 Configure Layer 3 VNI and Layer 2 VNI for the AP VLAN.

An AP is placed in the global routing table which has a LISP instance ID (VNI) attached.

In this example, Layer 3 instance ID for the global routing table is 4097 and the corresponding Layer 2 instance id is 8189.

Example:

```
router lisp
  instance-id 4097
   remote-rloc-probe on-route-change
   dynamic-eid APVlan92-IPV4
    database-mapping 10.92.1.0/28 locator-set rloc_set
    exit-dynamic-eid
   !
  exit-instance-id
 !
instance-id 8189
  remote-rloc-probe on-route-change
  service ethernet
   eid-table vlan 92
   database-mapping mac locator-set rloc_set
   exit-service-ethernet
  !
  exit-instance-id
 !
exit-router-lisp
!
```

Step 3 On the wireless controller, map the wireless SSID to the wireless client VLAN or subnet.

Example:

```
vlan 2055    //wireless client VLAN
 name Client_VLAN1

//Create wireless Policy Profile
wireless profile policy diy-localOTT-open_profile
 description diy-localOTT-open_profile
 dhcp-tlv-caching
 exclusionlist timeout 180
 http-tlv-caching
 service-policy input platinum-up
 service-policy output platinum
 vlan Client_VLAN1
 no shutdown

//Create Wirless SSID
wlan diy-localOTT-open_profile 17 diy-localOTT-open
 radio policy dot11 24ghz
 radio policy dot11 5ghz
 no security wpa
 no security wpa wpa2
 no security wpa wpa2 ciphers aes
 no security wpa akm dot1x
 no shutdown

//Create a Policy Tag to map the WLAN Profile to the Policy Profile
wireless tag policy wireless-policy-tag-open
wlan diy-localOTT-open_profile policy diy-localOTT-open_profile
```

Fabric-Enabled Wireless

A fabric-enabled wireless network integrates the wireless infrastructure with the wired fabric network. In a fabric with integrated wired and wireless, a single infrastructure for wired and wireless connectivity provides a uniform experience by having a common overlay for both the wired and wireless hosts. Wireless users get all the advantages of a fabric such as enhanced security with uniform policy application, data plane optimization, and operational simplicity.

- Wireless controller controls and manages all wireless functions. It interacts with the fabric control plane to notify the control plane node of all the wireless client joins, roams and disconnects.

- Fabric control plane node maintains the endpoint locator database for both the wired and wireless clients. It resolves the lookup requests from the fabric edge nodes to locate the endpoints. The control plane node notifies the fabric edge and border nodes about the wireless client mobility and RLOC information.
- Fabric APs connect directly to the fabric edge nodes. A fabric AP establishes a Control and Provisioning of Wireless Access Points (CAPWAP) tunnel to the fabric wireless controller and connects as local-mode AP. It applies all wireless specific features like SSID policies, AVC, QoS, so on, to the wireless endpoints.
- Fabric edge node onboards an AP into the fabric. It serves as a single Layer 3 default gateway for all the connected endpoints.
- Control plane traffic between the fabric APs and the fabric wireless controller is through the CAPWAP tunnel.
- For the data plane, a fabric AP establishes a VXLAN tunnel to the fabric edge node. Wireless data traffic traverses through this tunnel to reach the fabric edge node. The fabric edge node terminates the AP VXLAN tunnel and the client data traffic is placed on the wired fabric network. The VXLAN tunnel between the fabric AP and the fabric edge node carries the segmentation and policy information to and from the fabric edge node.

Note The rest of the document describes the fabric-enabled wireless mode of operation.

Platforms that Support Wireless Infrastructure in a LISP VXLAN Fabric

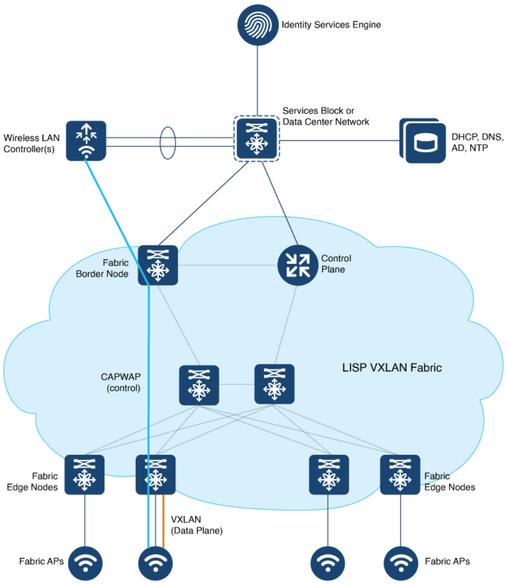
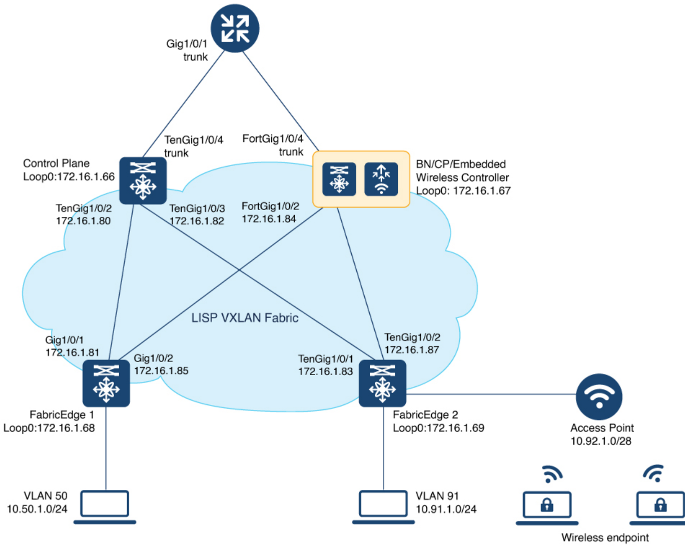
LISP VXLAN Fabric supports the following wireless devices:

- Cisco Catalyst 9800 Series Wireless Controller that is available in multiple form factors such as an Appliance, Cloud-based, or Embedded Wireless for a Switch.
- Wi-Fi 6 Access Points, which are the Cisco Catalyst 9100 Series APs.
- 802.11ac Wave 2 Access Points, which are the AP1540 Series, AP1560 Series, AP1800 Series, AP2800 Series, AP3800 Series, and AP4800 Series.

Wireless Controller

In a LISP VXLAN fabric, a wireless controller can either be hardware device or a software module that runs on a colocated control plane and border node.

The following table describes both these operational modes of a wireless controller.

Wireless Controller - Appliance or Virtual Form for Cloud	Embedded Wireless Controller
The wireless controller is a hardware device that is located external to the fabric. It is physically connected to the fabric border node or is located multiple hops upstream of the fabric border node (such as, in a Data Center). A fabric site can have one or multiple wireless controllers, but a wireless controller cannot be shared by different fabric sites. The wireless controller must have IP reachability with the control plane node of the LISP VXLAN fabric.	The wireless controller functionality is implemented as a software on a fabric node device. This is called an embedded wireless controller, which functions without a separate hardware device. Such an embedded wireless controller can be deployed in distributed branches or small campuses. Cisco Catalyst 9800 Embedded Wireless Controller software can be installed on a switch that functions as a colocated control plane and border node in the fabric. Cisco Catalyst 9300 Series switches, Cisco Catalyst 9400 Series switches, and Cisco Catalyst 9500 Series switches support Cisco Catalyst 9800 Embedded Wireless Controller. **Note** An embedded wireless controller works only in the fabric mode.
Figure 8: Fabric-Enabled Wireless with a Wireless Controller Appliance  Identity Services Engine Services Block or Data Center Network Wireless LAN Controller(s) DHCP, DNS, AD, NTP Fabric Border Node Control Plane CAPWAP (control) LISP VXLAN Fabric Fabric Edge Nodes VXLAN (Data Plane) Fabric Edge Nodes Fabric APs Fabric APs	***Figure 9: Fabric-Enabled Wireless with Embedded Wireless Controller***  Gig1/0/1 trunk TenGig1/0/4 trunk FortGig1/0/4 trunk Control Plane Loop0:172.16.1.66 BN/CP/Embedded Wireless Controller Loop0: 172.16.1.67 TenGig1/0/2 172.16.1.80 TenGig1/0/3 172.16.1.82 FortGig1/0/2 172.16.1.84 LISP VXLAN Fabric Gig1/0/1 172.16.1.81 Gig1/0/2 172.16.1.85 TenGig1/0/1 172.16.1.83 TenGig1/0/2 172.16.1.87 FabricEdge 1 Loop0:172.16.1.68 FabricEdge 2 Loop0:172.16.1.69 Access Point 10.92.1.0/28 VLAN 50 10.50.1.0/24 VLAN 91 10.91.1.0/24 Wireless endpoint

Fabric Access Points

The fabric APs connect directly to the fabric edge nodes and are part of the fabric overlay. AP subnets in the overlay are advertised to the external network and the wireless controller reaches the APs through the overlay. Control plane traffic from a fabric AP to the wireless controller (for the AP join operation) is through the CAPWAP tunnel.

All APs belong to a unique overlay virtual network called the Default Instance, which is mapped to the global routing table. A Default Instance connects network infrastructure elements like Access Points and Layer 2 switches to the fabric access layer. This unique overlay virtual network for all fabric APs simplifies the management of APs by including them within a single subnet.

Before onboarding the fabric APs, ensure that a default instance (instance-id 4097) is already configured on the fabric edge and border nodes. For configuration of a default instance, refer to *Configuring Fabric Edge Node* chapter. Map the AP subnet to the Layer 2 VNI and Layer 3 VNI for the Default Instance. Ensure that the fabric edge device is configured for Dot1x authentication of connected endpoints.

Workflow to Integrate Wireless in a LISP VXLAN Fabric

Before you begin the wireless integration, ensure that you have configured the fabric control plane node, border node, and the fabric edge node for a wired network.

Step	Purpose
Enabling the wireless controller for fabric operations	
Configure the wireless controller with the fabric control plane and virtual networks for the wireless clients and APs.	• Specify the fabric control plane name and its IP address. • Create the Layer 2 and Layer 3 VXLAN network identifiers (VNIDs) for the default instance. (A default instance is where the APs are placed.) • Create the Layer 2 VNID for the overlay virtual networks.
Configure the Wireless Management Interface of the wireless controller with the credentials to establish a secure connection with the fabric control plane node.	The wireless controller communicates with the control plane node on TCP port 4342 on the controller.
Create a **Fabric Profile** for the wireless clients.	• Specify the Layer 2 VNID. • Specify the SGT tag.
Create a **Policy Profile** to define the network policies and switching policies for a wireless client.	• Specify that traffic is local switching. • (Optional) Specify Quality of Service (QoS) – policing and marking policies on SSID and clients. • Specify AAA Override to override the VNID assignment of a client. This allows the AAA server to assign a specific virtual network to a client, based on the client's credentials and the policies configured on the AAA server.
Associate the previously created Fabric Profile with the Policy Profile.	The fabric inherits the associated policies.

Step	Purpose
Create a WLAN Profile to define the wireless characteristics of a WLAN.	• Specify the different types of SSID. For a fabric SSID, enable only Central Authentication. Disable Central Switching, Central DHCP and Flex NAT/PAT. • Specify the Security type for WLAN (PSK, 802.1x, WebAuthentication, and so on). If you define 802.1x or Central Web Authentication as the authentication method, ensure that you have configured AAA. • Specify advanced protocols such as 802.11k.
Create a Policy Tag to associate the SSID (WLAN Profile) with the Policy Profile.	Associating the Policy profile to an SSID applies the switching policies and the networking policies to the SSIDs.
Onboarding an AP	
Before onboarding an AP, ensure that a default instance (to host the AP subnets) is already created in the fabric.	
AP acquires an IP address through DHCP in the overlay.	After an AP connects to a fabric edge and boots up, it acquires an IP address from the DHCP server. The DHCP scope has option 43 configured, which defines the IP address of the wireless controller that the AP should reach out to.
AP registers with the fabric edge node.	The fabric edge node registers the AP's IP address and MAC address as endpoint ID (EID), with the control plane node.
AP registers with the wireless controller.	AP and the wireless controller exchange CAPWAP discovery and response messages. The wireless controller validates the AP and the AP validates the wireless controller to complete the discovery and AP join process. The validation on both the AP & WLC is a mutual authentication mechanism. An AP joins either through inbuilt certificates such as Manufacturer Installed Certificate (MIC) or third-party certificates such as Locally Significant Certificate (LSC).
Fabric edge builds a VXLAN tunnel to the AP. This serves as the data plane for the fabric wireless.	After an AP joins the fabric wireless controller in the local mode through CAPWAP, wireless controller queries the control plane about the AP's connectivity to the fabric infrastructure. After obtaining the RLOC of the AP, the wireless controller registers the AP with the control plane node. The control plane node then notifies the fabric edge about the presence of the AP. The fabric edge creates a VXLAN tunnel interface to the specified IP address of the AP.

Step	Purpose
Assign the previously created Policy Tag to the AP.	A Policy tag identifies the SSIDs and their policies, which are broadcasted by the AP. Site Tag and RF Tags also contain the settings to configure an AP. For information on the tags and their settings, refer to Understand Catalyst 9800 Wireless Controllers Configuration Model.
Onboarding Wireless Clients	
When a wireless client associates with a fabric AP, it is onboarded in the following manner: • Client authenticates with the wireless controller on an SSID that is enabled for fabric. • Wireless controller notifies the fabric AP to use VXLAN encapsulation to the fabric edge node and to populate the appropriate virtual network identifier (VNI) and source group tag (SGT) for that client in a VXLAN packet. • Wireless controller registers the client's MAC address in the fabric control plane node database. • After the client receives an IP address for itself through DHCP, the fabric edge node updates the control plane database with the client IP address. The MAC address and IP address of the client are mapped and correlated. The wireless client can now communicate through the fabric network.	

Wireless Client Roams

Consider a LISP VXLAN Wireless Figure 9: Fabric-Enabled Wireless with Embedded Wireless Controller where there are two fabric edge nodes (Fabric Edge 1 and Fabric Edge 2). Access point AP1 is connected to Fabric Edge 1 and AP2 is connected to Fabric Edge 2. A Catalyst 9800 Series embedded wireless controller runs on the colocated border and control plane node.

When a client that is connected to AP1 roams to AP2 (inter-switch roaming), the following sequence of events occur:

1. AP2 notifies the wireless controller about the client presence.
2. The wireless controller updates the forwarding table of AP2 with the client’s SGT and Layer 2 VNID.
3. The wireless controller updates the control plane node database with the client’s new RLOC (Fabric Edge 2).
4. The control plane notifies Fabric Edge 2 to add the client MAC address to its forwarding table.
5. The control plane then notifies Fabric Edge 1 to clean up the client info.
6. On receiving traffic from the client, Fabric Edge 2 updates the control plane with the client’s IP address.

An anycast gateway that is configured on all the fabric edges facilities seamless client roaming between the fabric edge nodes.

Prerequisites for Configuring Fabric-Enabled Wireless

- Ensure that the underlay network links are configured for routed access connectivity.
- Ensure that you have configured the fabric How to Configure a Control Plane Node, Detailed Steps to Configure a Border Node, and the How to Configure a Fabric Edge Node for a wired network.
- Ensure that there is a specific subnet reachability in the underlay (global routing table) for the wireless controller subnet at the access layer. This is required for the access points to connect to the wireless controller.
- For an embedded wireless controller:

  A fabric node switch that hosts the embedded controller should operate in Install mode for a wireless package to be installed on it. Install the Cisco Catalyst 9800 Series Wireless Controller as a sub-package on top of the base image on the fabric node switch.

  For information on booting a switch in Install mode and installing a sub-package, refer to Cisco Catalyst 9800 Series Wireless Controller Software Configuration Guide.

  Ensure that the wireless package is the same version as the base image on the switch (Cisco IOS XE). For example, if the switch is operating on Cisco IOS XE 17.10.1, install the 17.10.1 version of the wireless package on the switch.

  To download a wireless package, go to the Software Download page, navigate to the switch family, and select the **IOS XE Wireless Controller Software Package** Software Type.

  After the wireless package is installed, use the **show install summary** command on the switch to verify the version and state of the embedded wireless controller.

How to Configure Fabric-Enabled Wireless

Procedure

Step 1 Connect the wireless controller appliance to the fabric border node and initialize it.

For information on the initial setup of the wireless controller, refer to the Cisco Catalyst 9800 Wireless Controller Configuration Guide for the relevant release.

Step 2 Enable the wireless controller for fabric operations:

a. Configure the name and IP address of the wireless control plane.

b. Configure the wireless client VLAN and the AP VLAN.

c. Configure a fabric profile and associate the Layer 2 VXLAN network identifier (VNID), and optionally SGT, to the fabric profile.

d. Configure a wireless policy profile and map the fabric profile that was created in the previous step.

The following table describes the commands that configure a wireless controller for fabric operations.

Step	Command	Purpose
1	**configure terminal** **Example**: `WC# configure terminal`	Enters global configuration mode.
2	**wireless management interface** *interface-name* **Example**: `WC(config)# wireless management interface Vlan224`	Configure the management interface on the wireless controller.
3	**wireless fabric control-plane** *cp-name* **Example**: `WC(config)# wireless fabric control-plane default-control-plane`	Configures the name of the fabric control plane. You can assign a name of your choice to the control plane.

4	**ip address** *cp-ip address* **key** *authentication-key*
	Example:
	`WC(config-wireless-cp)# ip address 172.16.1.66 key some-key`
	`WC(config-wireless-cp)# end`
		Configures the IP address of the control plane and the authentication key shared with the control plane.

5	**wireless fabric name** *fabric-name* **l2-vnid** *l2-vnid* **control-plane-name** *cp-name*
	Example:
	`WC(config)# wireless fabric name wireless-Campus l2-vnid 8190`
	` control-plane-name default-control-plane`
		Configures the wireless client VLAN.

6	**wireless fabric name** *fabric-name* **l2-vnid** *l2-instance-id* **l3-vnid** *l3-instance-id* **control-plane-name** *cp-name*
	Example:
	`WC(config)# wireless fabric name APVlan92-IPV4 l2-vnid 8189`
	`l3-vnid 4097`
	`ip 10.92.1.1 255.255.255.0 control-plane-name`
	`default-control-plane`
		Configures the AP VLAN.

Step	Command	Purpose

7	**wlan** *wlan-name wlan-id SSID-name*
	Example:
	Create the following WLAN profiles:
	`wlan diy-psk_profile 17 diy-psk`
	` security ft over-the-ds`
	` security wpa psk set-key ascii 0 Cisco123`
	` no security wpa akm dot1x`
	` security wpa akm psk`
	` no shutdown`
	`!`
	`wlan diy_open_profile 18 diy_open`
	` no security ft adaptive`
	` no security wpa`
	` no security wpa wpa2`
	` no security wpa wpa2 ciphers aes`
	` no security wpa akm dot1x`
	` no shutdown`
	`!`
	`wlan diy-dot1x_profile 19 diy-dot1x`
	` security ft over-the-ds`
	` security dot1x authentication-list default`
	` security pmf optional`
	` no shutdown`
		Configures a WLAN.
		This example configures three WLANs with IDs 17, 18, 19 and SSID named diy-psk, diy_open , and diy-dot1x. It also enables the WLAN using the **no shutdown** command.

8	**wireless profile fabric** *profile-name*
	Example:
	Create the following fabric profiles:
	`wireless profile fabric diy-psk_profile`
	` description diy-psk_profile`
	` client-l2-vnid 8190    //Map to Layer 2 VNID 8190`
	` sgt-tag 22`
	
	`wireless profile fabric diy-dot1x_profile`
	` description diy-dot1x_profile`
	` client-l2-vnid 8191    //Map to Layer 2 VNID 8191`
	` sgt-tag 32`
	
	`wireless profile fabric diy-open_profile`
	` description diy-open_profile`
	` client-l2-vnid 8192 //Map to Layer 2 VNID 8192`
	` sgt-tag 42`
		Configures a fabric profile.
		This example configures three fabric profiles (*diy-psk_profile*, *diy_open_profile*, and *diy-dot1x_profile*), each mapped to a different Layer 2 VNI.

Step	Command	Purpose

9	**wireless profile policy** *profile-policy*
	Example:
	`wireless profile policy diy-psk_profile`
	` description diy-psk_profile`
	` no central dhcp    //specifies local DHCP mode`
	` no central switching  //configures WLAN for local switching`
	` dhcp-tlv-caching`
	` exclusionlist timeout 180`
	` fabric diy-psk_profile    //maps fabric profile with the policy profile`
	` http-tlv-caching`
	` service-policy input platinum-up`
	` service-policy output platinum`
	` no shutdown`
	`!`
	`wireless profile policy diy_open_profile`
	` description diy_open_profile`
	` no central dhcp`
	` no central switching`
	` dhcp-tlv-caching`
	` exclusionlist timeout 180`
	` fabric diy_open_profile    //maps fabric profile with the policy profile`
	` http-tlv-caching`
	` ip nbar protocol-discovery`
	` service-policy input platinum-up`
	` service-policy output platinum`
	` no shutdown`
	`!`
	
	`wireless profile policy diy-dot1x_profile`
	` description diy-dot1x_profile`
	` no central dhcp`
	` no central switching`
	` dhcp-tlv-caching`
	` exclusionlist timeout 180`
	` fabric diy-dot1x_profile    //maps fabric profile with the policy profile`
	` http-tlv-caching`
	` service-policy input platinum-up`
	` service-policy output platinum`
	` no shutdown`
		Configures a wireless policy profile for a given SSID and maps the fabric profile with this policy profile.
		This example configures three different wireless policy profiles, (*diy-psk_profile*, *diy_open_profile*, and *diy-dot1x_profile*) and maps the fabric profiles that were created earlier to these policy profiles.
		The wireless profile policy is mapped to a fabric profile using the **fabric** *profile-policy* command.

10	**wireless tag policy** *policy-tag-name* **Example**: `WC(config)# wireless tag policy wireless-policy-tag-psk`	Creates a Policy Tag and enters policy tag configuration mode. This example shows only one policy tag, namely *wireless-policy-tag-psk*. You can create more policy tags.

Step	Command	Purpose
11	**wlan** *wlan-name* **policy** *profile-policy-name* **Example**: `WC(config-policy-tag)# wlan diy-psk_profile policy diy-psk_profile`	Maps a policy profile to a WLAN profile. This example maps the profile policy *diy-psk_profile* that was created in Step 9 to the WLAN profile that was created in Step 7.

12	**end**
	Example:
	`WC(config-policy-tag)# end`
		Returns to privileged EXEC mode.

To see the GUI-based configurations of the wireless controller, click Configuring Wireless Controller for Fabric-Enabled Wireless (GUI).

Step 3 Integrate the wireless controller with the fabric control plane.

a) On the control plane node, define a locator set for the wireless controller.

Example:

```
router lisp
locator-set WLC
192.168.224.4  //IP address of the Wireless Management Interface
exit-locator-set
```

b) On the control plane node, configure open passive TCP sockets to listen for incoming connections. The wireless controller communicates with the control plane node on TCP port 4342.

Example:

```
map-server session passive-open WLC
```

c) On the control plane node, configure the LISP Site to accept EID prefixes.

Example:

```
 site site_uci
  description map-server1
  authentication-key some-key
  eid-record instance-id 4097 10.92.1.0/28 accept-more-specifics //AP subnet
  eid-record instance-id 4099 10.51.1.0/24 accept-more-specifics //New subnet for wireless
 clients
  eid-record instance-id 8189 any-mac
  eid-record instance-id 8190 any-mac
  eid-record instance-id 8191 any-mac
  exit-site
 !
exit-router-lisp
!
```

Step 4 On the border node, update the map cache with the AP subnets.

Example:

```
router lisp
 instance-id 4097  //Layer 3 instance-id for the default instance
```

```
  remote-rloc-probe on-route-change
  service ipv4
   eid-table default
   map-cache 10.92.1.0/28 map-request
   exit-service-ipv4
  !
  exit-instance-id
 !
exit-router-lisp
!
```

Step 5 Configure the fabric edge nodes to onboard the fabric APs. Do the following configurations on the fabric edge node.

a) Configure SVI interface for the wireless client VLAN.

Note

- Ensure that you assign the same MAC address for a given SVI, across all fabric edges within the fabric site. We recommend that you use a MAC address starting from the base range value of 0000.0C9F.F05F.

- IPv6 client address assignment through Stateless Address Auto-Configuration (SLAAC) depends on Router Solicitation (RS), Router Advertisement (RA), Neighbor Solicitation (NS), and Neighbor Discovery (ND) message sequences. A default RA interval of 200 seconds results in a longer duration for IP address resolution. To enable faster address convergence using SLAAC, we recommend that you configure a lower RA interval, such as 1000 milliseconds.

Example:

```
interface Vlan51
 description For Wirless Clients
 mac-address 0000.0c9f.f3b7   //Common MAC address
 vrf forwarding Campus
 ip address 10.51.1.1  255.255.255.0
 ip helper-address 192.168.136.1
 no ip redirects
 ip route-cache same-interface
 no lisp mobility liveness test
 lisp mobility wireless-Campus-IPV4
 lisp mobility wireless-Campus-IPV6
 ipv6 address 2001:192:168:166::1/96
 ipv6 enable
 ipv6 nd ra-interval msec 1000
 ipv6 nd dad attempts 0
 ipv6 nd managed-config-flag
 ipv6 nd other-config-flag
 ipv6 nd router-preference High
 ipv6 dhcp relay destination 2001:192:168:136::1
 ipv6 dhcp relay source-interface Vlan1023
 ipv6 dhcp relay trust
!
```

b) Configure SVI interface for the AP VLAN.

Note Ensure that you assign the same MAC address for a given SVI, across all fabric edges within the fabric site. We recommend that you use a MAC address starting from the base range value of 0000.0C9F.F05F.

Example:

```
interface Vlan92
 description For APs
 mac-address 0000.0c9f.ff39
 ip address 10.92.1.1 255.255.255.240
 no ip redirects
 no lisp mobility liveness test
 lisp mobility APVlan92-IPV4
end
!
```

c) Configure dynamic EID for the AP subnets in the default instance.

Example:

```
router lisp
  instance-id 4097
   remote-rloc-probe on-route-change
   dynamic-eid APVlan92-IPV4
    database-mapping 10.92.1.0/28 locator-set rloc_set
    exit-dynamic-eid
   !
  exit-instance-id
 !
```

d) Configure Layer 3 VNI for the wireless client subnet.

Example:

```
instance-id 4100
  remote-rloc-probe on-route-change
  dynamic-eid wireless-Campus-ipv4
   database-mapping 10.51.1.0/24 locator-set rloc_set
   exit-dynamic-eid
  !
  dynamic-eid wireless-Campus-ipv6
   database-mapping 2001:DB8:2051::/64 locator-set rloc_set
   exit-dynamic-eid
  !
  service ipv4
   eid-table vrf Campus
   map-cache 0.0.0.0/0 map-request
   exit-service-ipv4
  !
  service ipv6
   eid-table vrf Campus
   map-cache ::/0 map-request
   exit-service-ipv6
  !
  exit-instance-id
 !
```

e) Configure Layer 2 VNI for AP VLAN.

Example:

```
 instance-id 8189
  remote-rloc-probe on-route-change
  service ethernet
   eid-table vlan 92
   database-mapping mac locator-set rloc_set
   exit-service-ethernet
  !
  exit-instance-id
 !
```

f) Configure Layer 2 VNI for the wireless client VLAN.

Example:

```
 instance-id 8190
  remote-rloc-probe on-route-change
  service ethernet
   eid-table vlan 51
   database-mapping mac locator-set rloc_set
   exit-service-ethernet
  !
  exit-instance-id
 !
exit-router-lisp
!
```

g) Enable DHCP Snooping on the AP and Client VLANs.

Example:

```
ip dhcp snooping vlan 51,92
```

Configuring Wireless Controller for Fabric-Enabled Wireless (GUI)

Configuring a Fabric and its Control Plane (GUI)

Procedure

Step 1 Click **Configuration** > **Wireless** > **Fabric**.

Step 2 Under the **Control Plane** tab, click **Add**.

Step 3 In the **Add Control Plane** window, enter the name of the control plane and optionally a description. Click **Apply to Device** to save the control plane name.

Step 4 Under the **General** tab, click **Add**.

Step 5 In the **Add Client and AP VNID** window, enter the following values:

- Enter the name of the Fabric.
- Enter the Layer 2 virtual network ID (**L2 VNID**) for the wireless client and AP VLANs.
- Select a control plane node from the **Control Plane Name** drop down list.
- Enter the Layer 3 virtual network ID (**L3 VNID**) for the AP VLAN.
- Enter the **IP Address** and **Netmask** of the fabric control plane node.

Step 6 Click **Apply to Device** to save the configuration.

Configuring a Fabric Profile (GUI)

Procedure

Step 1 Choose **Configuration** > **Wireless** > **Fabric**.

Step 2 On the **Fabric** page, under the **Profiles** tab, click **Add**.

Step 3 In the **Add New Profile** window that is displayed, specify the following parameters:

- Profile name
- Description
- L2 VNID; valid range is between 0 and 16777215
- (Optional) SGT tag; valid range is between 2 and 65519

Step 4 Click **Apply to Device** to save the configuration.

Configuring a Wireless Profile Policy (GUI)

Procedure

Step 1 Choose **Configuration** > **Tags & Profiles** > **Policy**.

Step 2 On the **Policy Profile** page, click **Add**.

Step 3 In the **Add Policy Profile** window, under the **General** tab, enter a name and description for the policy profile. The name can be ASCII characters from 32 to 126, without leading and trailing spaces. Do not use spaces because it causes system instability.

Step 4 To enable the policy profile, set **Status** as **Enabled**.

Step 5 Use the slider to enable or disable **Passive Client** and **Encrypted Traffic Analytics**.

Step 6 n the **CTS Policy** section, choose the appropriate status for the following:

- Inline Tagging—a transport mechanism using which a controller or access point understands the source SGT.
- SGACL Enforcement.

Step 7 Specify a default **SGT**. The valid range is from 2 to 65519.

Step 8 In the WLAN Switching Policy section, enable **Central Authentication**. Central Authentication tunnels client data to the controller, as the controller handles client authentication.

Disable **Central Switching**, **Central DHCP**, and **Flex NAT/PAT**.

Step 9 Click **Apply to Device** to save the configuration.

Creating a WLAN Profile (GUI)

Procedure

Step 1 In the **Configuration** > **Tags & Profiles** > **WLANs** page, click **Add**.

The **Add WLAN** window is displayed.

Step 2 Under the **General** tab, enter the following information: .

a) In the **Profile Name** field, enter the name of the WLAN. The name can be ASCII characters from 32 to 126, without leading and trailing spaces
b) In the SSID field, enter a valid SSID for the WLAN. A valid SSID can be up to 32 characters and can contain spaces. A valid SSID can be ASCII characters from 0 to 31, with leading and trailing spaces. This is the broadcast name for your WLAN.
c) In the WLAN ID field, enter an ID for the WLAN.

Step 3 Enter a valid SSID for the WLAN. A valid SSID can be up to 32 characters and can contain spaces. A valid SSID can be ASCII characters from 0 to 31, with leading and trailing spaces. This is the broadcast name for your WLAN.

Step 4 Click **Apply to Device** to save the configuration.

Configuring WLAN Security (GUI)

An authentication method sets the method by which a client can access the WLAN and decides the level of security on the WLAN.

Set up the authentication configurations and filters for the WLAN depending on the method you have chosen. These include the keys, filters, ACLs, and parameter maps as applicable to the selected authentication method.

Procedure

Step 1 If you have selected **PSK** as the authentication method, configure the following:

a) In the **WLAN** > **Pre-Shared Key (PSK)** section, select the PSK format. Choose between ASCII and Hexadecimal formats.
b) From the **PSK type** drop-down list, choose if you want the key to be unencrypted or AES encrypted.
c) In the **Pre-Shared Key** field, enter the pass key for the WLAN.

Step 2 If you have selected **Dot1x** as the authentication method, configure the following:

a) In the **WLAN** > **AAA** tab, configure the AAA server list for the WLAN.
b) Select any of the available AAA servers to add to the WLAN.
c) To add a new AAA server to the list, click on Add New Server and enter the IP address and server-key.
d) To use an already configured AAA server list, click on **Use Existing** and select the appropriate list from the drop-down.

Step 3 If you have selected **Local Web Authentication** as the authentication method, configure the following:

a) In the **WLAN** > **Parameter Map** tab, configure the parameter map for the WLAN. A parameter map sets parameters that can be applied to subscriber sessions during authentication.

1. In the **Global Configuration** section, configure the global parameter map.
2. Enter an IPv4 or IPv6 address to configure a virtual IP address for redirecting the clients to the login page of the controller.
3. From the Trustpoint drop-down list, select the trustpoint for HTTPS login page. The trustpoint corresponds to the device certificate the controller will use in conjunction with the virtual IP and hostname.
4. In the **WLAN Specific Configuration** section, either create a new parameter map for the WLAN, or select an existing parameter map from the drop-down list.

b) In the **WLAN** > **Local Users** tab, enter the username in the local database to establish a username-based authentication system.

1. Enter the user name to be saved.
2. From the **Password Encryption** drop-down list, choose if you want the password to be unencrypted or encrypted.
3. In the **Password** field, specify the password the user must enter to gain access to the switch. The password must be from 1 to 25 characters and can contain embedded spaces.
4. Click on the **+** sign to add the credentials to the database. Add as many user credentials as required.

Step 4 If you have selected **External Web Authentication** as the authentication method, configure the following:

a) In the **WLAN** > **Parameter Map** tab, configure the parameter map for the WLAN.

1. In the **Global Configuration** section, configure the global parameter map.
2. Enter an IPv4 or IPv6 address to configure the virtual IP address of the external web authentication login page to which the guest users are redirected.
3. From the **Trustpoint** drop-down list, select the trustpoint for HTTPS login page. The trustpoint corresponds to the device certificate the controller will use in conjunction with the virtual IP and hostname.
4. In the **WLAN Specific Configuration** section, either create a new parameter map for the WLAN, or select an existing parameter map from the drop-down list.
5. To create a new parameter map, enter the parameter-map name.
6. In the **Redirect URL for login** field, enter the URL of the external server that will host the authentication page for login.
7. In the **Portal IPV4 Address** field, enter the IPv4 address of the external server to send redirects. If the external server uses an IPv6 address, in the **Portal IPV6 Address** field, enter the IPv6 address of the portal to send redirects.

b) In the **WLAN > ACL / URL Filter** tab, configure the ACL rules and the URL filter list.

1. In the **Pre Auth ACL** section, enter the name of the ACL.
2. In the **IP address** field, enter the source IP address and the destination IP address. This will configure the ACL to permit packet transfer from and to the specified IP address. You can add as many IP addresses as required.

3. In the **URL Filter** section, enter a name for the URL Filter list that you are creating.
4. Use the slider to set the list action to **Permit** or **Deny** the URLs.
5. Specify the URLs in the **URLs** box. Enter every URL on a new line.

Step 5 If you have selected Central Web Authentication as the authentication method, configure the following:

a) In the **WLAN > AAA/ACL** tab, configure the AAA server list and ACL for the WLAN.
b) In the **AAA Configuration** section, select any of the available AAA servers to add to the WLAN. This will be the server where the clients will get authenticated.
c) To add a new AAA server to the list, click on **Add New Server** and enter the IP address and server-key.
d) To use an already configured AAA server list, click on **Use Existing** and select the appropriate list from the drop-down.
e) In the **ACL List** section, enter the name of the ACL. This ACL will contain the rules regarding URLs that can be accessed by the client and should match the name configured on the RADIUS server.

Step 6 Click **Apply to Device** to save the configuration.

Configuring Policy Tag (GUI)

Procedure

Step 1 Choose **Configuration** > **Tags & Profiles** > **Tags** > **Policy**.

Step 2 Click **Add** to view the **Add Policy Tag** window.

Step 3 Enter a name and description for the policy tag. The name can be ASCII characters from 32 to 126, without leading and trailing spaces.

Step 4 Click **Add** to map WLAN and policy.

Step 5 Choose the WLAN profile to map with the appropriate policy profile, and click the tick icon.

Step 6 Click **Apply to Device** to save the configuration.

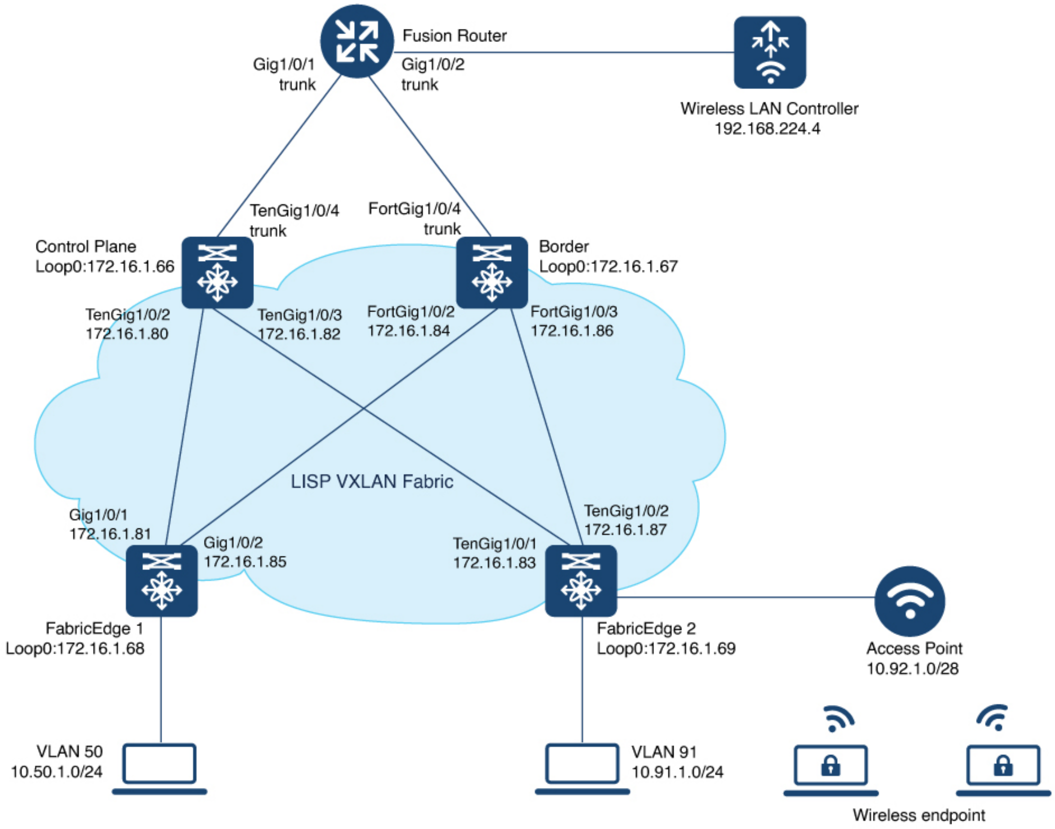
What to do next

Click Step 3 to continue the fabric configurations for integrating wireless.

Configuration Example for Fabric-Enabled Wireless

The example configurations described below are for the control plane node and the fabric edge node of a LISP VXLAN fabric shown in Figure 10: Fabric-enabled Wireless Topology. An upstream router connects the external border and the wireless controller. A fabric-enabled AP (10.92.1.0) is connected to Fabric Edge 2 (172.16.1.69) and is on VLAN 92. The wireless client IP subnets are 10.51.1.0/24 and 2001:DB8:2051::/64.

Figure 10: Fabric-enabled Wireless Topology



The example shows only the LISP configurations on the fabric nodes.

]

Control Plane Node Configuration	Fabric Edge Node Configuration

Control Plane Node Configuration

```
router lisp
 locator-table default
 locator-set WLC
  192.168.224.4
  exit-locator-set
 !
 service ipv4
  encapsulation vxlan
  sgt distribution
  sgt
  map-server
  map-resolver
  exit-service-ipv4
 !
 service ipv6
  encapsulation vxlan
  sgt distribution
  sgt
  map-server
  map-resolver
  exit-service-ipv6
 !
 service ethernet
  map-cache-limit 32768
  map-server
  map-resolver
  exit-service-ethernet
 !

 instance-id 4097
  service ipv4
   eid-table default
   route-export site-registrations
   distance site-registrations 250
   map-cache site-registration
   exit-service-ipv4
  !
  exit-instance-id
 !
 instance-id 4100
  service ipv4
   eid-table vrf Campus
   route-export site-registrations
   distance site-registrations 250
   map-cache site-registration
   exit-service-ipv4
  !
  service ipv6
   eid-table vrf Campus
   route-export site-registrations
   distance site-registrations 250
   map-cache site-registration
   exit-service-ipv6
  !
  exit-instance-id
 !
 instance-id 4101
  service ipv4
   eid-table vrf Guest
   route-export site-registrations
   distance site-registrations 250
   map-cache site-registration
```

Fabric Edge Node Configuration

```
router lisp
 locator-table default
 locator-set rloc_set
  IPv4-interface Loopback0 priority 10 weight 10
  exit-locator-set
 !
 locator default-set rloc_set
 service ipv4
  encapsulation vxlan
  itr map-resolver 192.168.94.1
  etr map-server 172.16.1.66 key some-key
  etr map-server 172.16.1.66 proxy-reply
  etr
  sgt
  no map-cache away-eids send-map-request
  use-petr 172.16.1.67
  proxy-itr 172.16.1.69
  exit-service-ipv4
 !
 service ipv6
  encapsulation vxlan
  itr map-resolver 192.168.94.1
  etr map-server 172.16.1.66 key some-key
  etr map-server 172.16.1.66 proxy-reply
  etr
  sgt
  no map-cache away-eids send-map-request
  use-petr 172.16.1.67
  proxy-itr 172.16.1.69
  exit-service-ipv6
 !
 service ethernet
  itr map-resolver 192.168.94.1
  itr
  etr map-server 172.16.1.66 key some-key
  etr map-server 172.16.1.66 proxy-reply
  etr
  exit-service-ethernet
 !
 instance-id 4097
  remote-rloc-probe on-route-change
  dynamic-eid AVlan91-IPV4
   database-mapping 10.91.1.0/24 locator-set rloc_set2
   exit-dynamic-eid
  !
  dynamic-eid APVlan92-IPV4
   database-mapping 10.92.1.0/28 locator-set rloc_set
   exit-dynamic-eid
  !
  service ipv4
   eid-table default
   exit-service-ipv4
  !
  exit-instance-id
 !
 instance-id 4099
  remote-rloc-probe on-route-change
  dynamic-eid AVlan50-IPV4
   database-mapping 10.50.1.0/24 locator-set rloc_set2
   exit-dynamic-eid
  !
  dynamic-eid AVlan50-IPV6
```

Control Plane Node Configuration

```
   exit-service-ipv4
  !
  exit-instance-id
 !
 map-server session passive-open WLC
 site site_uci
  description map-server
  authentication-key some-key
  eid-record instance-id 4097
        10.92.1.0/28 accept-more-specifics
  eid-record instance-id 4099
        10.51.1.0/24 accept-more-specifics
  eid-record instance-id 4099
        2001:DB8:2051::/64
accept-more-specifics
  eid-record instance-id 4097 0.0.0.0/0
                accept-more-specifics
  eid-record instance-id 4097 10.91.1.0/24
                accept-more-specifics
  eid-record instance-id 4099 0.0.0.0/0
                accept-more-specifics
  eid-record instance-id 4099 10.50.1.0/24
                   accept-more-specifics
  eid-record instance-id 4099 ::/0
                    accept-more-specifics
  eid-record instance-id 4099
2001:DB8:2050::/64

accept-more-specifics
  eid-record instance-id 8194 any-mac
  eid-record instance-id 8197 any-mac
  eid-record instance-id 8189 any-mac
  eid-record instance-id 8190 any-mac
  eid-record instance-id 8191 any-mac

  allow-locator-default-etr instance-id 4097
 ipv4
  allow-locator-default-etr instance-id 4099
 ipv4
  allow-locator-default-etr instance-id 4099
 ipv6
  exit-site
 !
 ipv4 source-locator Loopback0
 ipv6 source-locator Loopback0
 exit-router-lisp
```

Fabric Edge Node Configuration

```
   database-mapping 2001:DB8:2050::/64 locator-set rlo

   exit-dynamic-eid
  !
  service ipv4
   eid-table vrf VN3
   map-cache 0.0.0.0/0 map-request
   exit-service-ipv4
  !
  service ipv6
   eid-table vrf VN3
   map-cache ::/0 map-request
   exit-service-ipv6
  !
  exit-instance-id
 !
 instance-id 4100
  remote-rloc-probe on-route-change
  dynamic-eid wireless-Campus-ipv4
   database-mapping 10.51.1.0/24 locator-set rloc_se
   exit-dynamic-eid
  !
  dynamic-eid wireless-Campus-ipv6
   database-mapping 2001:DB8:2051::/64 locator-set rl

   exit-dynamic-eid
  !
  service ipv4
   eid-table vrf Campus
   map-cache 0.0.0.0/0 map-request
   exit-service-ipv4
  !
  service ipv6
   eid-table vrf Campus
   map-cache ::/0 map-request
   exit-service-ipv6
  !
  exit-instance-id
 !
 instance-id 4101 //guest
  remote-rloc-probe on-route-change
  dynamic-eid Campus-guest
   database-mapping 192.168.167.0/24 locator-set rlo

service ipv4
   eid-table vrf Guest
   map-cache 0.0.0.0/0 map-request
   exit-service-ipv4
  !
  exit-instance-id
 !
 instance-id 8194
  remote-rloc-probe on-route-change
  service ethernet
   eid-table vlan 91
   database-mapping mac locator-set rloc_set2
   exit-service-ethernet
  !
  exit-instance-id
 !
 !
 instance-id 8197
```

Control Plane Node Configuration	Fabric Edge Node Configuration

```
  remote-rloc-probe on-route-change
  service ethernet
   eid-table vlan 50
   database-mapping mac locator-set rloc_set2
   exit-service-ethernet
  !
  exit-instance-id
 !
 !
//APs in Global Instance
 instance-id 8189
  remote-rloc-probe on-route-change
  service ethernet
   eid-table vlan 92
   database-mapping mac locator-set rloc_set
   exit-service-ethernet
  !
  exit-instance-id
 !
//Wireless client in Custom VLAN
 instance-id 8190
  remote-rloc-probe on-route-change
  service ethernet
   eid-table vlan 51
   database-mapping mac locator-set rloc_set
   exit-service-ethernet
  !
  exit-instance-id
 !
//Guest VLAN
instance-id 8191
  remote-rloc-probe on-route-change
  service ethernet
   eid-table vlan 52
   database-mapping mac locator-set rloc_set
   exit-service-ethernet
  !
  exit-instance-id
 !
 ipv4 locator reachability minimum-mask-length 32
proxy-etr-only
 ipv4 source-locator Loopback0
 ipv6 locator reachability minimum-mask-length 128
proxy-etr-only
 ipv6 source-locator Loopback0
 exit-router-lisp
!
 vrf definition VN3
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 exit-address-family
!

 vrf definition Campus
 address-family ipv4
 exit-address-family
 !
ip dhcp relay information option
ip dhcp snooping vlan 50,91
ip dhcp snooping
```

Control Plane Node Configuration	Fabric Edge Node Configuration
	! device-tracking policy IPDT_POLICY tracking enable ! interface GigabitEthernet1/0/3 device-tracking attach-policy IPDT_POLICY ! vlan configuration 50 ipv6 nd raguard ipv6 dhcp guard ! vlan 50 name AVlan50 ! vlan 91 name AVlan91 ! interface Vlan50 description server1 mac-address 0000.0c9f.f18e vrf forwarding VN3 ip address 10.50.1.1 255.255.255.0 ip helper-address 172.16.2.2 no ip redirects ipv6 address 2001:DB8:2050::1/64 ipv6 enable ipv6 nd dad attempts 0 ipv6 nd prefix 2001:DB8:2050::/64 2592000 604800 no-autoconfig ipv6 nd managed-config-flag ipv6 nd other-config-flag ipv6 nd router-preference High ipv6 dhcp relay destination 2001:DB8:2::2 ipv6 dhcp relay source-interface Vlan50 ipv6 dhcp relay trust no lisp mobility liveness test lisp mobility AVlan50-IPV4 lisp mobility AVlan50-IPV6 ! interface Vlan91 description server2 mac-address 0000.0c9f.f984 ip address 10.91.1.1 255.255.255.0 ip helper-address 172.16.2.2 no ip redirects no lisp mobility liveness test lisp mobility AVlan91-IPV4 ! interface Vlan51 description For Wirless Clients mac-address 0000.0c9f.f3b7 vrf forwarding Campus ip address 10.51.1.1 255.255.255.0 ip helper-address 192.168.136.1. //DHCP IP no ip redirects no lisp mobility liveness test lisp mobility wireless-Campus-ipv4 lisp mobility wireless-Campus-ipv6 ipv6 address 2001:192:168:166::1/96 ipv6 enable

Control Plane Node Configuration	Fabric Edge Node Configuration
	ipv6 nd ra-interval msec 1000 ipv6 nd dad attempts 0 ipv6 nd managed-config-flag ipv6 nd other-config-flag ipv6 nd router-preference High ipv6 dhcp relay destination 2001:192:168:136::1 ipv6 dhcp relay source-interface Vlan51 ipv6 dhcp relay trust ! interface Vlan92 description For APs mac-address 0000.0c9f.ff39 ip address 10.92.1.1 255.255.255.240 no ip redirects no lisp mobility liveness test lisp mobility APVlan92-IPV4 ! ip dhcp snooping vlan 51,92

Fabric Wireless Controller Configuration

Fabric Wireless Controller Configuration
This table shows only those configurations on the wireless controller that are required to enable it for fabric operations. For complete configuration of a wireless controller, refer to the *Cisco Catalyst 9800 Wireless Controller Configuration Guide*.

```
wireless management interface Vlan224
wireless fabric control-plane default-control-plane
 ip address 192.168.94.1 key some-key
!
wireless fabric name wireless-Campus l2-vnid 8190
                       control-plane-name default-control-plane
wireless fabric name APVlan92-IPV4 l2-vnid 8189 l3-vnid 4097
ip 10.92.1.1 255.255.255.0 control-plane-name default-control-plane
!
wireless profile fabric diy-psk_profile
 client-l2-vnid 8190
 description diy-psk_profile
wireless profile fabric diy-dot1x_profile
 client-l2-vnid 8190
 description diy-dot1x_profile
wireless profile fabric diy-open_profile
 client-l2-vnid 8190
 description diy-open_profile
!
wlan diy-psk_profile 17 diy-psk
 security ft over-the-ds
 security wpa psk set-key ascii 0 Cisco123
 no security wpa akm dot1x
 security wpa akm psk
 no shutdown
!
wireless profile policy diy-psk_profile
 no central dhcp
 no central switching
 description diy-psk_profile
 dhcp-tlv-caching
 exclusionlist timeout 180
 fabric diy-psk_profile
 http-tlv-caching
 service-policy input platinum-up
 service-policy output platinum
 no shutdown
!

wlan diy-open_profile 18 diy-open
 radio policy dot11 24ghz
 radio policy dot11 5ghz
 no security wpa
 no security wpa wpa2
 no security wpa wpa2 ciphers aes
 no security wpa akm dot1x
 no shutdown
!
wireless profile policy diy-open_profile
 no central dhcp
 no central switching
 description diy-open_profile
 dhcp-tlv-caching
 exclusionlist timeout 180
 fabric diy-open_profile  <-- fabric wireless profile
 http-tlv-caching
```

Fabric Wireless Controller Configuration

```
 service-policy input platinum-up
 service-policy output platinum
 session-timeout 1800
 no shutdown
!
wlan diy-dot1x_profile 19 diy-dot1x
 security ft over-the-ds
 security dot1x authentication-list default
 security pmf optional
 no shutdown

wireless profile policy diy-dot1x_profile
 no central dhcp
 no central switching
 description diy-dot1x_profile
 dhcp-tlv-caching
 exclusionlist timeout 180
 fabric diy-dot1x_profile
 http-tlv-caching
 service-policy input platinum-up
 service-policy output platinum
 no shutdown
!
wireless tag policy wireless-policy-tag-psk
  wlan diy-psk_profile policy diy-psk_profile
!
wireless tag policy wireless-policy-tag-open
  wlan diy-open_profile policy diy-open_profile
!
wireless tag policy wireless-policy-tag-dot1x
  wlan diy-dot1x_profile policy diy-dot1x_profile
!
```

Verify the Fabric Enabled Wireless Configuration

You can verify the wireless fabric configurations using the show commands. This section provides the sample outputs for the show commands on the fabric wireless controller, control plane node and the fabric edge node in the topology shown Figure 10: Fabric-enabled Wireless Topology.

Show Commands on the Fabric Wireless Controller

```
wlc# show wireless fabric summary

Fabric Status      : Enabled

Control-plane:
Name                             IP-address        Key                              Status
------------------------------------------------------------------------------------------
default-control-plane            172.16.1.66      a021544b825b420e                 Up

Fabric VNID Mapping:
  Name            L2-VNID    L3-VNID   IP Address      Subnet       Control plane name

--------------------------------------------------------------------------------------
wireless-Campus   8190        0          0.0.0.0                     default-control-plane

APVlan92-IPV4     8189      4097       10.92.1.1    255.255.255.0    default-control-plane
```

```
wlc# show fabric wlan summary

Number of Fabric wlan : 3

WLAN Profile Name                     SSID                          Status
----------------------------------------------------------------------------
17   diy-psk_profile                  diy-psk                         UP
18   diy-open_profile                 diy-open                        UP
19   diy-dot1x_profile                diy-dot1x                       UP


wlc# show fabric ap summary
Number of Fabric AP : 4
fabric
AP Name                            Slots    AP Model             Ethernet MAC    Radio MAC
       Location               Country    IP Address      State
------------------------------------------------------------------------------------------

AP0CD0.F894.6540                     2      C9117AXI-B            0cd0.f894.6540
0cd0.f897.f6c0  default location         US       192.168.156.11  Registered
AP24D7.9C8D.464C                     2      C9120AXI-B            24d7.9c8d.464c
24d7.9cbf.3fa0  default location         US       192.168.156.15  Registered
9115-ts325-9500H                     2      C9115AXE-B            7069.5a76.7a50
2c4f.5241.3540  Global/BLR/BL1/FL1       US       192.168.156.14  Registered
9115-ts340-katarxtr                  2      C9115AXI-B            70f0.966c.a0f0
a488.737f.0780  Global/BLR/BL1/FL2       US       192.168.156.13  Registered


wlc# show wireless client summary
Number of Clients: 1

MAC Address    AP Name                Type ID   State    Protocol Method     Role
------------------------------------------------------------------------------------------

4c34.889a.06be AP0CD0.F894.6540       WLAN 18   Run      11ac     None       Local


Number of Excluded Clients: 0


wlc# show wireless client mac-address 4c34.889a.06be details

Client MAC Address : 4c34.889a.06be
Client MAC Type : Universally Administered Address
Client DUID: NA
Client IPv4 Address : 10.51.1.12
Client IPv6 Addresses : fe80::311d:6e13:9d40:9dab
Client Username: N/A
AP MAC Address : 0cd0.f897.f6c0
AP Name: AP0CD0.F894.6540
AP slot : 1
Client State : Associated
Policy Profile : diy-open_profile
Flex Profile : default-flex-profile
Wireless LAN Id: 18
WLAN Profile Name: diy-open_profile
Wireless LAN Network Name (SSID): diy-open
BSSID : 0cd0.f897.f6ce
Connected For : 41 seconds
Protocol : 802.11ac
Channel : 140
Client IIF-ID : 0xa0000001
Association Id : 1
Authentication Algorithm : Open System
Idle state timeout : N/A
Session Timeout : 1800 sec (Remaining time: 1764 sec)
Session Warning Time : Timer not running
```

```
Input Policy Name  : None
Fabric status : Enabled    <--- displays status of the fabric and other details
  RLOC    : 172.16.1.69
  VNID    : 8190
  SGT     : 0
  Control plane name  : default-control-plane

<snip output>
…..
…..
<snip output>
wlc#
```

Show Commands on the Fabric Edge Node where the AP Joins

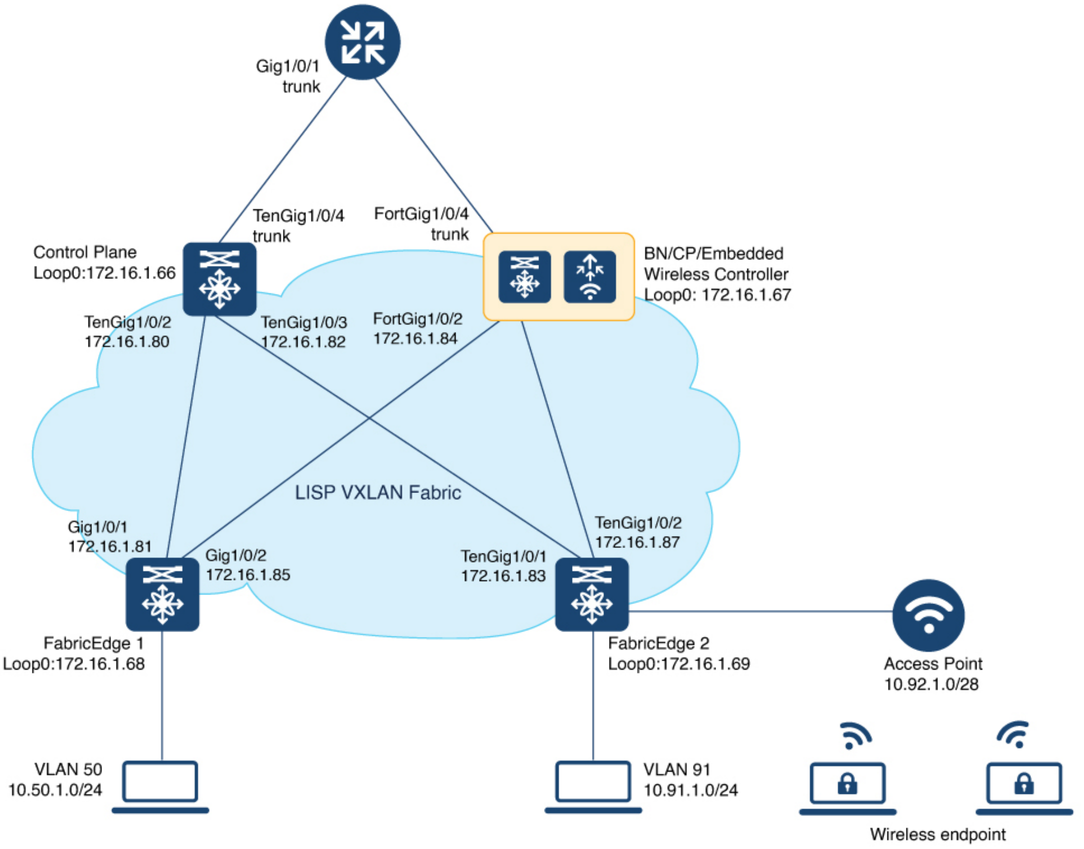
```
fabricedge# show access-tunnel summary

Access Tunnels General Statistics:
  Number of AccessTunnel Data Tunnels       = 2


Name    RLOC IP(Source)  AP IP(Destination)  VRF ID  Source Port  Destination Port
------  ---------------  ------------------  ------  -----------  ----------------
Ac0     172.16.1.69      192.168.156.15      0       N/A          4789
Ac1     172.16.1.69      192.168.156.11      0       N/A          4789


Name   IfId           Uptime
------ ---------- --------------------
Ac0    0x00000041 0 days, 00:10:24
Ac1    0x00000042 0 days, 00:03:24

fabricedge#
```

Configuration Example for Embedded Wireless in a LISP VXLAN Fabric

The example configurations described below are for the colocated control plane and border node, and the fabric edge node shown in the Figure 11: LISP VXLAN Fabric with Embedded Wireless to enable embedded wireless controller. The colocated control plane and border node has an loopback IP address of 172.16.1.67. A fabric enabled AP (10.92.1.0/24) is connected to Fabric Edge 2 (Loopback IP address 172.16.1.69) and is on VLAN 92. The wireless client IP subnet is 10.51.1.0/24.

For information on installing the embedded wireless controller, refer to List item..

Figure 11: LISP VXLAN Fabric with Embedded Wireless



This table only shows the LISP configurations on the fabric nodes, which are required to enable wireless operations.

Before you proceed, ensure that the you have configured the fabric for a wired network. For the sample configurations, refer to Configuration Example for Colocated Border Node and Configuration Example for LISP VXLAN Fabric Edge Node.

Control Plane, Border Node, and Embedded Wireless Controller	Fabric Edge Node

Control Plane, Border Node, and Embedded Wireless Controller	Fabric Edge Node

router lisp
 locator-table default
 locator-set WLC
  172.16.1.67
  exit-locator-set
 !
 locator-set rloc_set
  IPv4-interface Loopback0 priority 10 weight 10
  auto-discover-rlocs
  exit-locator-set
 !
 locator default-set rloc_set
 service ipv4
  encapsulation vxlan
  itr map-resolver 172.16.1.67
  etr map-server 172.16.1.67 key some-key
  etr map-server 172.16.1.67 proxy-reply
  etr
  sgt distribution
  sgt
  no map-cache away-eids send-map-request
  proxy-etr
  proxy-itr 172.16.1.67
  map-server
  map-resolver
  exit-service-ipv4
 !
 service ethernet
  map-cache-limit 65536
  itr map-resolver 172.16.1.67
  itr
  etr map-server 172.16.1.67 key 7 some-key
  etr map-server 172.16.1.67 proxy-reply
  etr
  map-server
  map-resolver
  exit-service-ethernet
 !
 instance-id 4097
  remote-rloc-probe on-route-change
  service ipv4
   eid-table default
   map-cache 10.92.1.0/24 map-request
   route-export site-registrations
   distance site-registrations 250
   map-cache site-registration
   exit-service-ipv4
  !
  exit-instance-id
 !
 instance-id 4099
  remote-rloc-probe on-route-change
  service ipv4
   eid-table vrf CLIENT_VN
   route-export site-registrations
   distance site-registrations 250
   map-cache site-registration
   exit-service-ipv4
  !
	router lisp
	 locator-table default
	 locator-set rloc_set2
	  IPv4-interface Loopback0 priority 10 weight 10
	  exit-locator-set
	 !
	 locator default-set rloc_set2
	 service ipv4
	  encapsulation vxlan
	  itr map-resolver 172.16.1.67
	  etr map-server 172.16.1.67 key some-key
	  etr map-server 172.16.1.67 proxy-reply
	  etr
	  sgt distribution
	  sgt
	  no map-cache away-eids send-map-request
	  use-petr 172.16.1.67
	  proxy-itr 172.16.1.69
	  exit-service-ipv4
	 !
	 service ethernet
	  itr map-resolver 172.16.1.67
	  itr
	  etr map-server 172.16.1.67 key some-key
	  etr map-server 172.16.1.67 proxy-reply
	  etr
	  exit-service-ethernet
	 !
	 instance-id 4097
	  remote-rloc-probe on-route-change
	  dynamic-eid APVlan92-IPv4
	   database-mapping 10.92.1.0/24 locator-set rloc_set2
	   exit-dynamic-eid
	  !
	  service ipv4
	   eid-table default
	   exit-service-ipv4
	  !
	  exit-instance-id
	 !
	 instance-id 4099
	  remote-rloc-probe on-route-change
	  dynamic-eid wireless-VN-IPV4
	   database-mapping 10.51.1.0/24 locator-set rloc_set2
	   exit-dynamic-eid
	  !
	  service ipv4
	   eid-table vrf CLIENT_VN
	   map-cache 0.0.0.0/0 map-request
	   exit-service-ipv4
	  !
	  exit-instance-id
	 !
	 instance-id 8190
	  remote-rloc-probe on-route-change
	  service ethernet
	   eid-table vlan 1023

Control Plane, Border Node, and Embedded Wireless Controller	Fabric Edge Node
exit-instance-id ! map-server session passive-open WLC site site_uci description map-server1 authentication-key some-key eid-record instance-id 4097 10.92.1.0/24 accept-more-specifics eid-record instance-id 4099 10.51.1.0/24 accept-more-specifics eid-record instance-id 8190 any-mac eid-record instance-id 8191 any-mac exit-site ! ipv4 locator reachability exclude-default ipv4 source-locator Loopback0 exit-router-lisp !! wireless profile fabric diy_open_profile client-l2-vnid 8191 description diy_open_profile wireless profile policy diy_open_profile no central dhcp no central switching description diy_open_profile dhcp-tlv-caching exclusionlist timeout 180 fabric diy_open_profile http-tlv-caching ip nbar protocol-discovery service-policy input platinum-up service-policy output platinum no shutdown wlan diy_open_profile 17 diy_open no security ft adaptive no security wpa no security wpa wpa2 no security wpa wpa2 ciphers aes no security wpa akm dot1x no shutdown ! wireless management interface Loopback0 wireless fabric wireless fabric name APVlan92 l2-vnid 8190 l3-vnid 4097 ip 10.92.1.0 255.255.255.0 control-plane-name default-control-plane wireless fabric name wireless-VN l2-vnid 8191 control-plane-name default-control-plane wireless fabric control-plane default-control-plane ip address 172.16.1.67 key 0 auth-key ! interface Loopback1023 description Loopback Border ip address 10.92.1.1 255.255.255.255 ! interface Loopback1024 description Loopback Border	database-mapping mac locator-set rloc_set2 exit-service-ethernet ! exit-instance-id ! instance-id 8191 remote-rloc-probe on-route-change service ethernet eid-table vlan 1024 database-mapping mac locator-set rloc_set2 exit-service-ethernet ! exit-instance-id ! ipv4 locator reachability minimum-mask-length 32 proxy-etr-only ipv4 source-locator Loopback0 exit-router-lisp snmp-server enable traps ! interface Vlan92 description AP SVI mac-address 0000.0c9f.fcae ip address 10.92.1.1 255.255.255.0 ip helper-address 192.168.132.1 no ip redirects no lisp mobility liveness test lisp mobility APVlan92-IPv4 end interface Vlan51 description Client SVI mac-address 0000.0c9f.fd96 vrf forwarding CLIENT_VN ip address 10.51.1.1 255.255.255.0 ip helper-address 192.168.132.1 no ip redirects no lisp mobility liveness test lisp mobility wireless-VN-IPV4 end ip dhcp snooping vlan 51,92 !

Control Plane, Border Node, and Embedded Wireless Controller	Fabric Edge Node
vrf forwarding CLIENT_VN ip address 10.51.1.1 255.255.255.255 ! ! router bgp 700 bgp router-id interface Loopback0 bgp log-neighbor-changes bgp graceful-restart ! address-family ipv4 bgp redistribute-internal bgp aggregate-timer 0 network 10.92.1.1 mask 255.255.255.255 exit-address-family ! address-family ipv4 vrf CLIENT_VN bgp aggregate-timer 0 network 10.51.1.1 mask 255.255.255.255 exit-address-family !	

CHAPTER 7

Configuring a Multi-Site Remote Border

Configure a multi-site remote border if you require a centralized gateway for a subset of the Virtual Networks (VNs) across multiple fabric sites. The traffic for those VNs will egress the fabric from the multi-site remote border at the central site.

This section describes how to configure a multi-site remote border.

Multi-Site Remote Border

A multi-site remote border enables the fabric network to isolate untrusted traffic to a central location like a firewall or a DMZ (demilitarized zone). For example, if the network has a guest virtual network (VN) that is stretched across multiple sites, all the guest traffic can be tunneled to a remote border at the DMZ, thus isolating the guest traffic from the enterprise traffic.

In a multi-site network deployment, you can designate a common border (multi-site remote border) to route the traffic to and from a particular VN that is stretched across multiple sites. This allows you to deploy a VN across multiple fabric sites but have a single subnet across all these sites. Preserving the subnets across multiple fabric sites helps in conserving the IP address space.

Here are some common terms that are used in the context of a multi-site remote border:

Anchor Virtual Network (VN): A virtual network that exists across multiple fabric sites in a network. The associated IP subnet and segment are common across these multiple sites.

Anchor Site: The fabric site that hosts the common border and control plane for an Anchor VN. Anchor Site handles the ingress and egress traffic for the Anchor VN.

Anchoring Sites: Fabric sites other than the Anchor Site where the Anchor VN is deployed.

Anchor Border Node or Multi-Site Remote Border: The fabric border node at the Anchor Site that provides the ingress and egress location for traffic to and from the Anchor VN.

Anchor Control Plane Node: The fabric control plane node at the Anchor Site that accepts registrations and responds to requests for endpoints in the Anchor VN.

A Use Case for a Multi-Site Remote Border

Different users and devices in an enterprise network require different levels of access on the network. A guest user connecting to a fabric site can be permitted to access the internet but should not be permitted to access business sensitive data or network resources like shared folders, storage devices, and so on. The guest users connecting to multiple fabric sites in an enterprise network must be handled in a secure and reliable manner.

In a typical case, an endpoint (which could be a guest user) in a fabric site is assigned an Endpoint Identifier (EID) address from the local EID subnet and its traffic is directed through the local border. This adds complexity to the policy enforcement and EID address management for guests across multiple sites. To achieve traffic isolation and better manage the guest traffic, you can direct all the guest traffic to a designated border node which is located in the DMZ site. (A DMZ site provides access to external network like the internet but prevents external users from accessing the resources or data of the fabric network.) The DMZ site will now be the ingress and egress site for traffic to and from the guest VN.

Guidelines for Configuring a Multi-Site Remote Border

- An Anchor VN can have only one Anchor Site.
- The path from the fabric edge node of the Anchoring Site to the multi-site remote border should support frames greater than 1500 bytes.
- We recommend a value of 1250 bytes for the Transmission Control Protocol (TCP) Maximum Segment Size (MSS) on the on the overlay SVI interfaces.

How to Configure a Multi-Site Remote Border

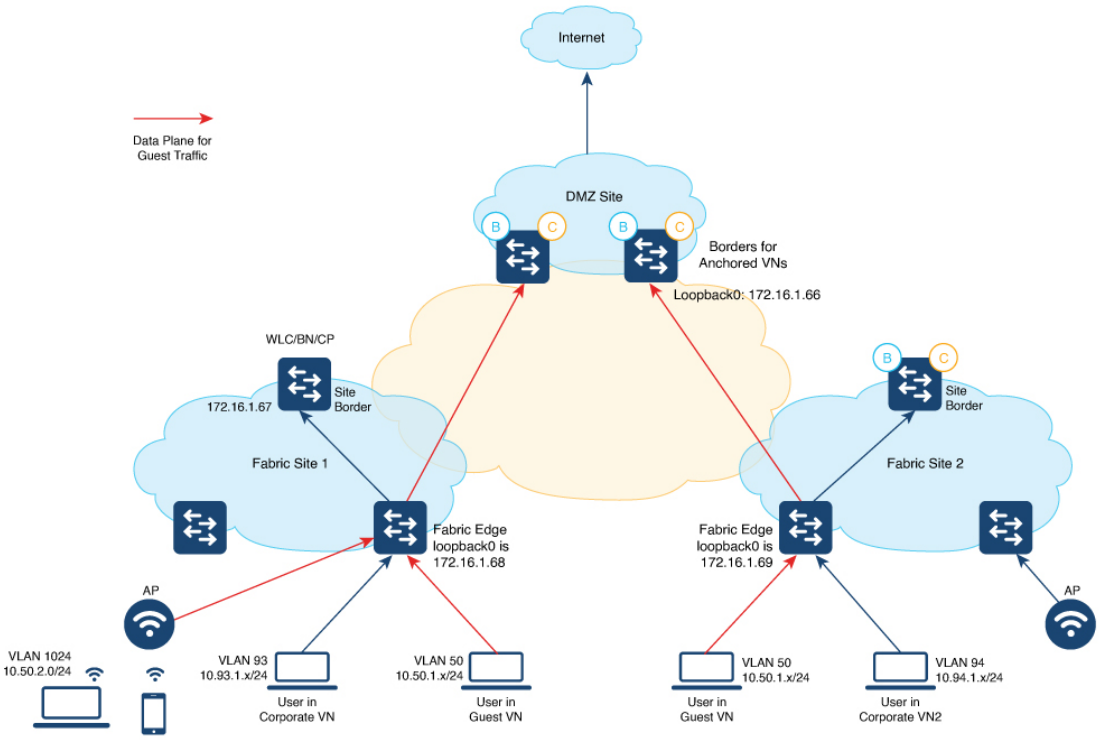
This section shows only the configurations on the Anchor Site and the Anchoring Sites for a multi-site remote border.

Before you begin, provision the fabric sites in the network. For a complete description of the fabric site configurations, refer the earlier chapters of this document.

To anchor a VN and configure a multi-site remote border, do the following:

- Configure the control plane node at the Anchor Site to act as the map-server and map-resolver for the requests from the Anchor VN.
- Configure the EID prefixes of the Anchor VN only on the control plane node at the Anchor Site. The control plane node of the Anchoring Sites should not be configured with the EID prefixes of the Anchor VN.

In the following topology, a Guest VN (Anchor VN) is spread across Fabric Site 1 and Fabric Site 2 (Anchoring Sites). Each of these fabric sites has its own control plane node and border nodes. The DMZ site (Anchor Site) has a colocated control plane node and border node (CPB), which is configured as the multi-site remote border.



Note The following is a snippet of the configurations on the fabric edge nodes and the DMZ control plane node. The snippet shows only the configurations that are required for a multi-site remote border functionality. For complete configurations on the fabric nodes, refer to the earlier chapters in the document.

Colocated Control Plane and Border Node at DMZ site	Fabric Edge Nodes at the Local Fabric Site

<table>
<tr>
<th>Colocated Control Plane and Border Node at DMZ site</th>
<th>Fabric Edge Nodes at the Local Fabric Site</th>
</tr>
<tr>
<td>

- Configure the LISP Site on the DMZ to accept the guest EID prefixes.
- If you have wireless guests, define a locator set for the wireless controller and configure open passive TCP sockets to listen for incoming connections.
- Define the Layer 3 instance ID for the guests.

<snip: only the relevant configuration is shown>

```
router lisp
 locator-table default
 locator-set WLC
  172.16.1.67
  exit-locator-set
 !
 locator default-set rloc_set
 service ipv4
  encapsulation vxlan
  itr map-resolver 172.16.1.66
  etr map-server 172.16.1.66 key 7
auth-key
  etr map-server 172.16.1.66 proxy-reply
  etr
  sgt
  no map-cache away-eids send-map-request

  proxy-etr
  proxy-itr 172.16.1.66
  map-server
  map-resolver
  exit-service-ipv4
 !
 service ethernet
  itr map-resolver 172.16.1.66
  itr
  etr map-server 172.16.1.66 key 7
auth-key
  etr map-server 172.16.1.66 proxy-reply
  etr
  map-server
  map-resolver
  exit-service-ethernet
 !
 instance-id 4100
  remote-rloc-probe on-route-change
  service ipv4
   eid-table vrf Guest
   database-mapping 10.52.2.8/30
locator-set rloc_set
   route-export site-registrations
   distance site-registrations 250
   map-cache site-registration
   exit-service-ipv4
  !
  exit-instance-id
```

</td>
<td>

Ensure that you use the same authentication key on the control plane node, fabric edge node, and wireless controller.

<snip: only the relevant configuration is shown>

```
router lisp
 locator-table default locator-set rloc_set
 IPv4-interface Loopback0 priority 10 weight 10
exit-locator-set
  !
 locator default-set rloc_set service ipv4
 encapsulation vxlan
  //Control plane is at the local Site
 itr map-resolver 172.16.1.67
 etr map-server 172.16.1.67 key some-key
 etr map-server 172.16.1.67 proxy-reply etr
 sgt
 proxy-itr 172.16.1.68 exit-service-ipv4
  !
 service ipv6 encapsulation vxlan
  //Control plane is at the local Site
 itr map-resolver 172.16.1.67
 etr map-server 172.16.1.67 key some-key
 etr map-server 172.16.1.67 proxy-reply etr
 sgt
 proxy-itr 172.16.1.68 exit-service-ipv6
  !
 service ethernet
  //Control plane is at the local Site
 itr
 itr map-resolver 172.16.1.67
 etr map-server 172.16.1.67 key some-key
 etr map-server 172.16.1.67 proxy-reply etr
 exit-service-ethernet
!
//Configurations for the Anchor VN with instance
 id 4099
instance-id 4099
 remote-rloc-probe on-route-change
 dynamic-eid AVlan50-IPV4
  database-mapping 10.50.1.0/24 locator-set
rloc_set
  exit-dynamic-eid
  !
 dynamic-eid AVlan50-IPV6
  database-mapping 2001:DB8:2050::/64 locator-set
 rloc_set
  exit-dynamic-eid
  !
 service ipv4
 eid-table vrf GuestVN
 map-cache 0.0.0.0/0 map-request
   //Control plane is at the DMZ Site
 itr map-resolver 172.16.1.66
 etr map-server 172.16.1.66 key auth-key
 etr map-server 172.16.1.66 proxy-reply
 etr
 proxy-itr 172.16.1.68
 exit-service-ipv4
!
```

</td>
</tr>
</table>

Colocated Control Plane and Border Node at DMZ site	Fabric Edge Nodes at the Local Fabric Site
! map-server session passive-open WLC site site_uci description mapserver authentication-key auth-key eid-record instance-id 4099 0.0.0.0/0 accept-more-specifics eid-record instance-id 4099 10.50.1.0/24 accept-more-specifics eid-record instance-id 4099 ::/0 accept-more-specifics eid-record instance-id 4099 2001:DB8:2050::/64 accept-more-specifics eid-record instance-id 16188 any-mac eid-record instance-id 4100 0.0.0.0/0 accept-more-specifics allow-locator-default-etr instance-id 4099 ipv4 allow-locator-default-etr instance-id 4099 ipv6 exit-site ! ipv4 locator reachability exclude-default ipv4 source-locator Loopback0 exit-router-lisp ! <snip>	service ipv6 eid-table vrf GuestVN map-cache ::/0 map-request // **Control plane is at the DMZ Site** itr map-resolver 172.16.1.66 etr map-server 172.16.1.66 key auth-key etr map-server 172.16.1.66 proxy-reply etr proxy-itr 172.16.1.68 exit-service-ipv6 ! exit-instance-id ! // **Associate Guest Layer 2 VNID (16188) with the** // **control plane node at the DMZ site** **(172.16.1.66)** instance-id 16188 remote-rloc-probe on-route-change service ethernet eid-table vlan 50 database-mapping mac locator-set eid_locator //**Control plane is at the DMZ Site** itr map-resolver 172.16.1.66 itr etr map-server 172.16.1.66 key auth-key etr map-server 172.16.1.66 proxy-reply etr exit-service-ethernet ! exit-instance-id ! //**Associate Guest Layer 3 VNID (4100) with the** // **control plane node at the DMZ site** **(172.16.1.66)** instance-id 4100 remote-rloc-probe on-route-change dynamic-eid guest-wireless-IPV4 database-mapping 10.50.2.0/24 locator-set rloc_set exit-dynamic-eid ! service ipv4 eid-table vrf Guest map-cache 0.0.0.0/0 map-request //**Control plane is at the DMZ Site** itr map-resolver 172.16.1.66 etr map-server 172.16.1.66 key 7 auth-key etr map-server 172.16.1.66 proxy-reply etr use-petr 172.16.1.66 proxy-itr 192.168.113.1 exit-service-ipv4 ! exit-instance-id ! exit-router-lisp ! <snip>

Wireless Controller at the Anchoring Site

- The wireless controller has LISP sessions with both the site control plane and the common control plane at the DMZ site.
- If you configure a guest SSID and associate it to a guest control plane node, the corresponding instance ID on the fabric edge also should associate with the same control plane node.

<snip: only the relevant configuration is shown>

```
//Configure the Guest SSID to use the control plane at the DMZ
wireless fabric control-plane anchor-vn-control-plane
 ip address 172.16.1.66 key 0 auth-key
!
wireless fabric name guest-wireless l2-vnid 16188 control-plane-name anchor-vn-control-plane

//Configure the wireless hosts and APs to use the control plane node at the local Site
wireless fabric control-plane default-control-plane
 ip address 172.16.1.67 key 0 some-key
!
wireless fabric name AP_VLAN l2-vnid 8188 l3-vnid 4097 ip 192.168.155.0 255.255.255.0
control-plane-name default-control-plane
wireless fabric name wireless-campus l2-vnid 8189 control-plane-name default-control-plane

//Configure the Guest SSID
wlan diy-guest_profile 18 diy-guest
 mac-filtering prof-cts-diy-gu-1f67e529
 no security ft adaptive
 no security wpa
 no security wpa wpa2
 no security wpa wpa2 ciphers aes
 no security wpa akm dot1x
 no shutdown

// Configure a Fabric Profile for the Guests
wireless profile fabric diy-guest_profile
 client-l2-vnid 16188
 description diy-guest_profile

// Configure a Policy Profile for the Guests
wireless profile policy diy-guest_profile
 aaa-override
 no central dhcp
 no central switching
 description diy-guest_profile
 dhcp-tlv-caching
 exclusionlist timeout 180
 fabric diy-guest_profile
 http-tlv-caching
 nac
 service-policy input silver-up
 service-policy output silver
 no shutdown

// Create a Policy Tag to map the WLAN Profile to the Policy Profile
wireless tag policy wireless-policy-tag-guest
wlan diy-guest_profile policy diy-guest_profile
```

<snip>

Verify Multi-Site Remote Border Configuration

Use the following show commands to verify the Multi-Site Remote Border configuration.

To see the LISP sessions that are established by the wireless controller, use the **show lisp session** command on the wireless controller.

```
wlc# show lisp session

Sessions for VRF default, total: 6, established: 4
Peer                 State      Up/Down        In/Out    Users
172.16.1.69:19360  Up         00:55:21       15/35     7
172.16.1.67:4342   Up         01:44:58       51/9      7
172.16.1.67:52937  Up         01:44:58        9/51     4
172.16.1.67:63963  Up         01:44:41        0/11     1
wlc#
```

To see the wireless fabric status and verify that the guest traffic is controlled at the Anchor Site, use the **show wireless fabric summary** command on the wireless controller.

```
wlc# show wireless fabric summary

Fabric Status      : Enabled


Control-plane:
Name                             IP-address        Key                              Status
------------------------------------------------------------------------------------------
anchor-vn-control-plane          192.168.102.1     7fb28b01b3e049ed                Up
default-control-plane            192.168.223.1     fbe1110d55b643cc                Up


Fabric VNID Mapping:
  Name               L2-VNID        L3-VNID        IP Address             Subnet
Control plane name
------------------------------------------------------------------------------------------

  AP_VLAN             8188           4097           192.168.155.0       255.255.255.0
default-control-plane
  guest-wireless      16188          0                                  0.0.0.0
anchor-vn-control-plane
  wireless-campus     8189           0                                  0.0.0.0
default-control-plane
```

To see the LISP sessions that are established by the fabric edge node at the local site, use the **show lisp session** command on the fabric edge node.

The command output shows that LISP sessions are established with the control plane node at the local fabric site as well as with the control plane node at the Anchor Site.

```
fabricEdge# show lisp session

Sessions for VRF default, total: 2, established: 2
Peer                 State      Up/Down        In/Out    Users
172.16.1.66:4342   Up         01:09:59       46/27     5
172.16.1.67:4342   Up         01:10:00       35/15     13
fabricEdge#
```

PART II

LISP VXLAN Fabric in a Branch

CHAPTER 8

Configuring Fabric In a Box for Wired Devices

A remote office or a branch office necessitates the design of a small fabric site. It could be a site with less than 200 endpoints and less than five virtual networks. In such cases, use a fabric in a box design. Fabric in a box is a single device that is configured as a border node, a control plane node and an edge node. This single device can be a switch with hardware stacking, or with StackWise Virtual deployment.

The following platforms support fabric in a box:

- Cisco Catalyst 9300 Series Switches
- Cisco Catalyst 9400 Series Switches
- Cisco Catalyst 9500 Series Switches

This section describes the configuration of a fabric in a box for small sites.

- How to Configure Fabric in a Box, on page 161
- Configuration Example for a Fabric in a Box Device, on page 190
- Verify Fabric in a Box, on page 195

How to Configure Fabric in a Box

Use the Fabric in a box construct for smaller sites or remote branch deployments.

Note Before you begin, ensure that the underlay network links are configured for routed access connectivity.

Step	Task	Purpose
Step 1	Configure VRFs	Configure a VRF to support IPv4 and IPv6 routing tables. VRF maintains the routing and forwarding information for devices within a virtual network. A VRF instance has its own IP routing table, a forwarding table, and one or more interfaces assigned to it. The VRF tables help the routing device reach the locator address space.

Step	Task	Purpose
Step 2	Configure Layer 3 Handoff	Configure the interface on the device for external connectivity and Layer 3 handoff.
Step 3	Configure Device Tracking	Configure Switch Integrated Security Features based (SISF-based) device tracking to track the presence, location, and movement of endpoints in the fabric. SISF snoops traffic received by the device, extracts device identity (MAC and IP address), and stores them in a binding table.
Step 4	Configure VLAN	Configure VLANs to segment your network and achieve traffic isolation between the segments.
Step 5	Configure SVI Interface	Configure an SVI interface for each VLAN. A Switched Virtual Interface (SVI) interface is a VLAN interface that allows traffic to be routed between the VLANs. DHCP Snooping on a VLAN enables DT-PROGRAMMATIC policy that supports onboarding of DHCPv4 hosts.
Step 6	Configure DHCP Relay and Snooping	Configure the fabric in a box device as a DHCP relay agent to relay the DHCP traffic between fabric endpoints and DHCP server.
Step 7	Configure LISP	• Set up the Ingress Tunnel Router (ITR) and Proxy Ingress Tunnel Router (PITR) functionalities for both IPv4 and IPv6 address families. An ITR or PITR encapsulates and forwards the incoming packets across the overlay either to a fabric edge node or to the external network, depending on the destination. • Set up the Egress Tunnel Router (ETR) and Proxy Egress Tunnel Router (PETR) functionalities for both IPv4 and IPv6 address families. An ETR or PETR decapsulates the LISP VXLAN-encapsulated packets and sends them to the endpoint. • Configure a Map Server to receive and store the endpoint registrations. • Configure a Map Resolver to resolve a lookup request for route to destination endpoints. • Define this border node as a default ETR and map the default route for each VRF.

Step	Task	Purpose
Step 8	Configure Layer 3 VNI and Segment for Default Instance Configure Layer 3 VNI for User-Defined VRF	In a LISP VXLAN fabric, the VXLAN-GPO header has a VNI field that serves as an identifier of a specific virtual network. VXLAN VNI helps carry the macro segmentation information within the fabric site. A Layer 3 VNI identifies a Layer 3 overlay segment. • Configure Layer 3 VNI for the Default Instance. The default instance is used to connect the network infrastructure elements like Access Points and Layer 2 switches to the fabric access layer. • Configure Layer 3 VNI for VLANs in user-defined VRF.
	Configure Layer 2 VNI for Default Instance, on page 184 Configure Layer 2 VNI for User-Defined VRF, on page 185	A Layer 2 VNI identifies a Layer 2 overlay segment. Configure Layer 2 VNI for the Default Instance. Configure Layer 2 VNI for the User-Defined VRF. Configuring Layer 2 VNI programmatically enables these first-hop-security policies on the VLANs: LISP-DT-GUARD-VLAN and LISP-AR-RELAY-VLAN. LISP-DT-GUARD-VLAN policy mitigates IP theft, MAC theft and DOS attacks. LISP-AR-RELAY policy helps in converting ARP broadcast and Neighbor Solicitation (NS) multicast packets to unicast.
Step 9	Configure BGP	Configure Border Gateway Protocol (BGP) for route exchange with the external network.
Step 10	Configure Route-Map	Configure a prefix list and route map for redistribution and route leaking between the global routing table (GRT) and the VRF.

Step	Task	Purpose
Step 11	Verify the configurations on the fabric in a box device using these **show** commands:	
	show lisp session	Displays the details of the LISP sessions that are established on the device.
	show lisp locator-set	Displays the locator set information.
	show ip interface brief	Displays the usability status of all the interfaces that are configured on the device. Filter the output to view the dynamically created LISP interfaces, using the **show ip interface brief | i LISP** command.
	show lisp instance-id *instance-id* **ipv4** **show lisp instance-id** *instance-id* **ipv6**	Displays the details of each of the LISP IPv4 or IPv6 instances that are configured on the device. Use this command to view the operational status of the IPv4 or the IPv6 address family under each instance-id. This includes the status of the database, map-cache, publication entries, site registration entries, and so on.
	show lisp instance-id *instance-id* **ethernet server**	Displays the LISP site registration information such as the site name, the node that registered last, status of the site, and the EID prefixes that are associated with the site.
	show lisp instance-id *instance-id* **ethernet database**	Displays the database mappings on the device Use this command to check EID table for a given VLAN
	show ip route vrf ***vrf***	Displays the route table that is created on the node for a given VRF.
	show lisp platform	Displays the limits of the given platform or the device. This command shows the LISP instance limits, Layer 3 limits, Layer 2 limits, and the supported configuration style on the device. Use this command to understand the limits of the device and plan its usage and role in the fabric.

Configure VRFs

To configure VRFs on the fabric in a box device, perform this task:

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	**vrf definition** *vrf-name* **Example:** `Device(config)# vrf definition VN3`	Configures a VRF table, and enters VRF configuration mode.
Step 4	**rd** *route-distinguisher* **Example:** `Device(config-vrf)# rd 1:4099`	Creates routing and forwarding tables for a VRF instance.

Step 5	**address-family** {**ipv4** | **ipv6**}
	Example:
	`Device(config-vrf)# address-family ipv4`
	`Device(config-vrf)# address-family ipv6`
		Specifies the address family, and enters address family configuration mode.
		• **ipv4**: Specifies the address family as IPv4.
		• **ipv6**: Specifies the address family as IPv6.

Step 6	**route-target export** *route-target-ext-community* **Example:** `Device(config-vrf-af)# route-target export 1:4099`	Creates a list of export route target communities for the specified VRF. Enter either an AS system number and an arbitrary number (xxx:y) or an IP address and an arbitrary number (A.B.C.D:y). The *route-target-ext-community* value should be the same as the *route-distinguisher* value entered in the earlier step.
Step 7	**route-target import** *route-target-ext-community* **Example:** `Device(config-vrf-af)# route-target import 1:4099`	Creates a list of import route target communities for the specified VRF.
Step 8	**exit-address-family** **Example:** `Device(config-vrf-af)# exit-address-family`	Exits address family configuration mode, and enters VRF configuration mode.
Step 9	**end** **Example:**	Returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-vrf)# end`	

Configure Layer 3 Handoff

To configure Layer 3 handoff on a fabric in a box device, perform this task:

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	**vlan** *vlan-id* **Example:** `Device(config)# vlan 222`	Places you into the VLAN configuration submode. If the VLAN does not exist, the system creates the specified VLAN and then enters the VLAN configuration submode.
Step 4	**name** *vlan-name* **Example:** `Device(config-vlan)# name 222`	Names the VLAN.
Step 5	**exit** **Example:** `Device(config)# exit`	Returns to global configuration mode.
Step 6	**interface** *interface-name* **Example:** `Device(config)# interface Vlan222`	Specifies the VLAN interface and enters the interface configuration mode.
Step 7	**description** *interface-description* **Example:** `Device(config-if)# description vrf-external`	Adds a description for the interface
Step 8	**vrf forwarding** *vrf-name* **Example:** `Device(config-if)# vrf forwarding VN3`	Associates the VRF instance with the interface.
Step 9	**ip address** *ip_address subnet_mask* **Example:**	Configures the IP address and IP subnet.

	Command or Action	Purpose
	`Device(config-if)# ip address 10.20.1.1 255.255.255.252`	
Step 10	**no ip redirects** **Example:** `Device(config-if)# no ip redirects`	Disables sending of Internet Control Message Protocol (ICMP) redirect messages.
Step 11	**ipv6 address** *address* **Example:** `Device(config-if)# ipv6 address 2001:DB8:20::1/126`	Configures an IPv6 address on the interface.
Step 12	**ipv6 enable** **Example:** `Device(config-if)# ipv6 enable`	Enables IPv6 on the interface.
Step 13	**exit** **Example:** `Device(config-if)# exit`	Returns to global configuration mode.
Step 14	**interface** *interface-number* **Example:** `Device(config)# interface TenGigabitEthernet1/0/4`	Specifies the interface and enters the interface configuration mode.
Step 15	**switchport mode trunk** **Example:** `Device(config-if)# switchport mode trunk`	Configures the interface as a VLAN trunk port. Configures the physical interface toward Fusion router.

Step 16	**end**
	Example:
	`Device(config-if)# end`
		Returns to privileged EXEC mode.

Configure Device Tracking

To configure device-tracking on a fabric in a box device, perform this task:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**device-tracking policy** *policy-name* **Example:** `Device(config)# device-tracking policy IPDT_POLICY`	Creates a device-tracking policy with the specified name, and enters the device-tracking configuration mode.
Step 4	**tracking enable** **Example:** `Device(config-device-tracking)# tracking enable`	Enables polling for the specified policy.
Step 5	**exit** **Example:** `Device(config-device-tracking)# exit`	Exits device-tracking configuration mode, and enters global configuration mode.
Step 6	**interface** *interface-id* **Example:** `Device(config)# interface TenGigabitEthernet1/0/5`	Specifies an interface and enters interface configuration mode.
Step 7	**device-tracking attach-policy** *policy-name* **Example:** `Device(config-if)# device-tracking attach-policy IPDT_POLICY`	Attaches the device tracking policy to the interface.

Step 8	**end**
	Example:
	`Device(config-if)# end`
		Returns to privileged EXEC mode.

Configure VLAN

To configure VLAN on a FiaB, perform this task:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 nd raguard** **Example:** `Device(config)# ipv6 nd raguard`	Configures the default Router Advertisement (RA) Guard policy on the VLAN. The RA Guard feature analyzes the RAs and filters out bogus RAs sent by unauthorized devices. In host mode, all router advertisement and router redirect messages are disallowed on the port.
Step 4	**ipv6 dhcp guard** **Example:** `Device(config)# ipv6 dhcp guard`	Configures the default DHCP Guard policy on the VLAN. The IPv6 DHCP Guard feature blocks reply and advertisement messages that come from unauthorized DHCPv6 servers and relay agents.
Step 5	**vlan** *vlan-id* **Example:** `Device(config)# vlan 50`	Specifies a VLAN ID, and enters VLAN configuration mode.
Step 6	**name** *vlan-name* **Example:** `Device(config-vlan)# name AVlan50`	Specifies a name for the VLAN.
Step 7	**exit** **Example:** `Device(config-vlan)# exit`	Exits VLAN configuration mode, and enters global configuration mode.
Step 8	**vlan** *vlan-id* **Example:** `Device(config)# vlan 91`	Specifies a VLAN ID, and enters VLAN configuration mode.
Step 9	**name** *vlan-name* **Example:** `Device(config-vlan)# name AVlan91`	Specifies a name for the VLAN.
Step 10	**exit** **Example:** `Device(config-vlan)# exit`	Exits VLAN configuration mode, and enters global configuration mode.
Step 11	**end** **Example:**	Returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config)# end`	

Configure SVI Interface

To configure SVI interface for a VLAN, perform this task.

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	**interface** *vlan-id* **Example:** `Device(config)# interface Vlan50`	Specifies the interface for which you are adding a description, and enters interface configuration mode.
Step 4	**description** *string* **Example:** `Device(config-if)# description conf-vrf`	Adds a description for an interface.
Step 5	**mac-address** *address* **Example:** `Device(config-if)# mac-address 0000.0c9f.f18e`	Specifies the MAC address for the VLAN interface (SVI). We recommend that you use a MAC address starting from the base range value of 0000.0C9F.F05F.
Step 6	**vrf forwarding** *name* **Example:** `Device(config-if)# vrf forwarding VN3`	Associates the VRF instance with the interface.
Step 7	**ip address** *ip_address subnet_mask* **Example:** `Device(config-if)# ip address 10.50.1.1 255.255.255.0`	Configures the IP address and IP subnet.
Step 8	**ip helper-address** *ip_address* **Example:** `Device(config-if)# ip helper-address 172.16.2.2`	Configures the IP helper address.

	Command or Action	Purpose
Step 9	**no ip redirects** **Example:** `Device(config-if)# no ip redirects`	Disables sending of Internet Control Message Protocol (ICMP) redirect messages.
Step 10	**ipv6 address** *address* **Example:** `Device(config-if)# ipv6 address 2001:DB8:2050::1/64`	Configures an IPv6 address on the interface.
Step 11	**ipv6 enable** **Example:** `Device(config-if)# ipv6 enable`	Enables IPv6 on the interface.

Step 12	**ipv6 nd** {**dad attempts** | **prefix** | **managed-config-flag** | **other-config-flag** | **router-preference** | }
	Example:
	`Device(config-if)# ipv6 nd dad attempts 0`
	`Device(config-if)# ipv6 nd prefix 2001:DB8:2050::/64 2592000 604800 no-autoconfig`
	`Device(config-if)# ipv6 nd managed-config-flag`
	`Device(config-if)# ipv6 nd other-config-flag`
	`Device(config-if)# ipv6 nd router-preference High`
		Configures IPv6 neighbor discovery on the interface.
		• **dad attempts**: Specifies the number of consecutive neighbor solicitation messages that are sent on an interface while duplicate address detection is performed on the unicast IPv6 addresses of the interface.
		• **prefix**: Specifies IPv6 prefixes that are included in IPv6 neighbor discovery router advertisements.
		• **managed-config-flag**: Specifies IPv6 interfaces neighbor discovery to allow the hosts to uses DHCP for address configuration.
		• **other-config-flag**: Specifies IPv6 interfaces neighbor discovery to allow the hosts to uses DHCP for non-address configuration.
		• **router-preference**: Specifies a default router preference (DRP) for the router on a specific interface.

Step 13	**ipv6 dhcp relay** {**destination** | **source-interface** | **trust**}
	Example:
	`Device(config-if)# ipv6 dhcp relay destination 2001:DB8:2::2`
	`Device(config-if)# ipv6 dhcp relay source-interface Vlan50`
	`Device(config-if)# ipv6 dhcp relay trust`
		Configures Dynamic Host Configuration Protocol (DHCP) for IPv6 relay service on the interface.
		• **destination**: Specifies a destination address to which client messages are forwarded.

	Command or Action	Purpose
		- **source-interface**: Specifies an interface to use as the source when relaying messages received on this interface. - **trust**: Specifies the interface to be trusted to process relay-replies.
Step 14	**no lisp mobility liveness test** **Example:** `Device(config-if)# no lisp mobility liveness test`	Removes mobility liveness settings discovered on this interface.

Step 15	**lisp mobility** *dynamic-eid-name*
	Example:
	`Device(config-if)# lisp mobility AVlan50-IPV4`
	`Device(config-if)# lisp mobility AVlan50-IPV6`
		Specifies the name of the LISP dynamic-EID policy to apply to this interface.

Step 16	**no autostate** **Example:** `Device(config-if)# no autostate`	Brings up the VLAN even if there is no trunk or physical link that is up on that device.

Step 17	**end**
	Example:
	`Device(config-if)# end`
		Returns to privileged EXEC mode.

Configure DHCP Relay and Snooping

To configure DHCP relay and snooping on a fabric in a box device, perform this task:

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	**ip dhcp relay information option** **Example:**	Enables the system to insert the DHCP relay agent information option (option-82 field) in

	Command or Action	Purpose
	`Device(config)# ip dhcp relay information option`	forwarded BOOTREQUEST messages to a DHCP server.
Step 4	**ip dhcp snooping vlan** {*vlan id* | *vlan range*} **Example:** `Device(config)# ip dhcp snooping vlan 50,91`	Enables DHCP snooping on a VLAN or VLAN range.
Step 5	**ip dhcp snooping** **Example:** `Device(config)# ip dhcp snooping`	Enables DHCP snooping globally.

Step 6	**end**
	Example:
	`Device(config)# end`
		Returns to privileged EXEC mode.

Configure LISP

To configure LISP on a fabric in a box device, perform this task:

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	**router lisp** **Example:** `Device(config)# router lisp`	Enters LISP configuration mode.
Step 4	**locator-table default** **Example:** `Device(config-router-lisp)# locator-table default`	Selects the default (global) routing table for association with the routing locator address space.
Step 5	**locator-set** *loc-set-name* **Example:** `Device(config-router-lisp)# locator-set default_etr_locator`	Specifies a locator-set, and enters the locator-set configuration mode.

	Command or Action	Purpose
Step 6	**ipv4-interface Loopback** *loopback-interface-id* **priority** *locator-priority* **weight** *locator-weight* **Example:** `Device(config-router-lisp-locator-set)# IPv4-interface Loopback0 priority 10 weight 10`	Configures the loopback IP address to ensure the device is reachable.
Step 7	**exit-locator-set** **Example:** `Device(config-router-lisp-locator-set)# exit-locator-set`	Exits locator-set configuration mode, and enters LISP configuration mode.
Step 8	**locator-set** *loc-set-name* **Example:** `Device(config-router-lisp)# locator-set rloc_set`	Specifies a locator-set, and enters the locator-set configuration mode. Ensure that this locator set is different from the default locator.
Step 9	**ipv4-interface Loopback** *loopback-interface-id* **priority** *locator-priority* **weight** *locator-weight* **Example:** `Device(config-router-lisp-locator-set)# IPv4-interface Loopback0 priority 10 weight 10`	Specifies that the IPv4 address of the loopback interface should be used to reach the locator.
Step 10	**auto-discover-rlocs** **Example:** `Device(config-router-lisp-locator-set)# auto-discover-rlocs`	Auto discover the locators registered by other ingress or egress tunnel routers (xTRs).
Step 11	**exit-locator-set** **Example:** `Device(config-router-lisp-locator-set)# exit-locator-set`	Exits locator-set configuration mode, and enters LISP configuration mode.
Step 12	**locator default-set** *loc-set-name* **Example:** `Device(config-router-lisp)# locator default-set rloc_set`	Specifies a default locator-set.

Step 13	**service** {**ipv4** | **ipv6**}
	Example:
	`Device(config-router-lisp)# service ipv4`
	`Device(config-router-lisp)# service ipv6`
		Enables network services for the default instance.
		service ipv4: Enables Layer 3 network services for the IPv4 address family.
		service ipv6: Enables Layer 3 network services for the IPv6 address family.

	Command or Action	Purpose

Step 14	**encapsulation vxlan**
	Example:
	`Device(config-router-lisp-serv-ipv4)# `**`encapsulation vxlan`**
	`Device(config-router-lisp-serv-ipv6)# `**`encapsulation vxlan`**
		Specifies VXLAN-based encapsulation.

Step 15	**map-cache publications**
	Example:
	`Device(config-router-lisp-serv-ipv4)# `**`map-cache publications`**
	`Device(config-router-lisp-serv-ipv6)# `**`map-cache publications`**
		Exports the publication entries to the map cache. These entries are used for forwarding the traffic.

Step 16	**import publication publisher** *publisher-address*
	Example:
	`Device(config-router-lisp-serv-ipv4)# `**`import publication publisher 172.16.1.68`**
	`Device(config-router-lisp-serv-ipv6)# `**`import publication publisher 172.16.1.68`**
		Imports the publications from the publisher that is specified by the *publisher-address*. *publisher-address* is the IP address of the Loopback 0 interface of the control plane node.

Step 17	**itr map-resolver** *map-resolver-address*
	Example:
	`Device(config-router-lisp-serv-ipv4)# `**`itr map-resolver 172.16.1.68`**
	`Device(config-router-lisp-serv-ipv6)# `**`itr map-resolver 172.16.1.68`**
		Configures a locator address for the LISP map resolver. To resolve the EID-to-RLOC mappings, this router sends map request messages to the map resolver.
		A control plane node is the LISP map resolver. Specify the IP address of the Loopback 0 interface on control plane node as the *map-resolver-address*.

Step 18	**etr map-server** *map-server-address* **key** *authentication-key*
	Example:
	`Device(config-router-lisp-serv-ipv4)# `**`etr map-server 172.16.1.68 key 7 auth-key`**
	`Device(config-router-lisp-serv-ipv6)# `**`etr map-server 172.16.1.68 key 7 auth-key`**
		Configures a map server to be used by the Egress Tunnel Router (ETR), and specifies the key type.
		A control plane node is the LISP map server. Specify the IP address of the Loopback 0 interface on control plane node as the *map-server-address*.

Step 19	**etr map-server** *map-server-address* **proxy-reply** **Example:** `Device(config-router-lisp-serv-ipv4)# `**`etr map-server 172.16.1.68 proxy-reply`**	Configures a locator address for the LISP map server and an authentication key. This device acting as a LISP ETR, uses the authetication key to register with the LISP mapping system.

	Command or Action	Purpose

	`Device(config-router-lisp-serv-ipv6)#`
	`etr map-server 172.16.1.68 proxy-reply`	

Step 20	**etr**
	Example:
	`Device(config-router-lisp-serv-ipv4)#`
	`etr`
	`Device(config-router-lisp-serv-ipv6)#`
	`etr`
		Configures the device as an Egress Tunnel Router (ETR).

Step 21	**sgt**
	Example:
	`Device(config-router-lisp-serv-ipv4)#`
	`sgt`
	`Device(config-router-lisp-serv-ipv6)#`
	`sgt`
		Enables the Security Group Tag (SGT) function for SGT tag propagation.

Step 22	**route-export publications**
	Example:
	`Device(config-router-lisp-serv-ipv4)#`
	`route-export publications`
	`Device(config-router-lisp-serv-ipv6)#`
	`route-export publications`
		Exports the LISP publications into the routing information base (RIB).

Step 23	**distance publications 250**
	Example:
	`Device(config-router-lisp-serv-ipv4)#`
	`distance publications 250`
	`Device(config-router-lisp-serv-ipv6)#`
	`distance publications 250`
		Specifies the administrative distance to RIB when the LISP publications are exported to the RIB.

Step 24	**proxy-etr**
	Example:
	`Device(config-router-lisp-serv-ipv4)#`
	`proxy-etr`
	`Device(config-router-lisp-serv-ipv6)#`
	`proxy-etr`
		Enables Proxy Egress Tunnel Router (PETR) functionality for the EIDs.

Step 25	**proxy-itr** *address*
	Example:
	`Device(config-router-lisp-serv-ipv4)#`
	`proxy-itr 172.16.1.68`
	`Device(config-router-lisp-serv-ipv6)#`
	`proxy-itr 172.16.1.68`
		Enables Proxy Ingress Tunnel Router (PITR) functionality for the EIDs.
		For *address*, specify the Loopback 0 IP address of this device.

Step 26	**map-server** **Example:**	Configures the locator address of the LISP map server.

	Command or Action	Purpose

	`Device(config-router-lisp-serv-ipv4)# map-server`
	`Device(config-router-lisp-serv-ipv6)# map-server`	

Step 27	**map-resolver**
	Example:
	`Device(config-router-lisp-serv-ipv4)# map-resolver`
	`Device(config-router-lisp-serv-ipv6)# map-resolver`
		Configures the locator address of the LISP map resolver.

Step 28	Do one of the following:
	- **exit-service-ipv4**
	- **exit-service-ipv6**
	Example:
	`Device(config-router-lisp-serv-ipv4)# exit-service-ipv4`
	`Device(config-router-lisp-serv-ipv6)# exit-service-ipv4`
		Exits service configuration mode, and enters LISP configuration mode.
		Use the appropriate command, depending on which service mode you are exiting from (IPv4 or IPv6 service mode).

Step 29	**service ethernet** **Example:** `Device(config-router-lisp)# service ethernet`	Enables Layer 2 network services for the default instance.
Step 30	**itr map-resolver** *map-resolver-address* **Example:** `Device(config-router-lisp-serv-eth)# itr map-resolver 172.16.1.68`	Configures a locator address for the LISP map resolver to which this router will send map request messages for IPv4 EID-to-RLOC mapping resolutions.
Step 31	**itr** **Example:** `Device(config-router-lisp-serv-eth)# itr`	Configures the device as an Ingress Tunnel Router (ETR).
Step 32	**etr map-server** *map-server-address* **key** *authentication-key* **Example:** `Device(config-router-lisp-serv-eth)# etr map-server 172.16.1.68 key 7 auth-key`	Configures a map server to be used by the Egress Tunnel Router (ETR), and specifies the key type. *map-server-address* is the IP address of the Loopback 0 interface on the control plane node. In this step, specify the Loopback 0 IP address of the device because the control plane node, border node, and edge node are all configured on a single device.
Step 33	**etr map-server** *map-server-address* **proxy-reply**	Configures a locator address for the LISP map server and an authentication key for which this

	Command or Action	Purpose

	Example:
	`Device(config-router-lisp-serv-eth)#`
	`etr map-server 172.16.1.68 proxy-reply`
		router, acting as an IPv4 LISP ETR, will use to register with the LISP mapping system.
		map-server-address is the IP address of the Loopback 0 interface on the control plane node. In this step, specify the Loopback 0 IP address of the device because the control plane node, border node, and edge node are all configured on a single device.

Step 34	**etr**
	Example:
	`Device(config-router-lisp-serv-eth)#`
	`etr`
		Configures the device as an Egress Tunnel Router (ETR).

Step 35	**map-server**
	Example:
	`Device(config-router-lisp-serv-eth)#`
	`map-server`
		Configures the device as a Map Server.

Step 36	**map-resolver**
	Example:
	`Device(config-router-lisp-serv-eth)#`
	`map-resolver`
		Configures the device as a Map Resolver.

Step 37	**exit**
	Example:
	`Device(config-router-lisp-serv-eth)#`
	`exit`
		Exits service Ethernet configuration mode and enters LISP configuration mode.

Step 38	**site** *site-name*
	Example:
	`Device(config-router-lisp)#` **`site`**
	`site_uci`
		Specifies a LISP site named *site-name* and enters LISP site configuration mode.
		A LISP site name is locally significant to the map server on which it is configured. It has no relevance anywhere else. This name is used solely as an administrative means of associating one or more EID prefixes with an authentication key and other site-related mechanisms

Step 39	**description** *description*
	Example:
	`Device(config-router-lisp-site)#`
	`desription map-server1`
		Provides a description for the LISP site.

Step 40	**authentication-key** {*key-type*} *authentication-key* **Example:**	Configures the authentication key associated with this site.

	Command or Action	Purpose

	`Device(config-router-lisp-site)#`
	`authentication-key 7 auth-key`	

Step 41	**eid-record instance-id** *instance-id* [*eid-prefix*] [**accept-more-specifics**]
	Example:
	`Device(config-router-lisp-site)#`
	`eid-record instance-id 4097 10.91.1.0/24 accept-more-specifics`
	`Device(config-router-lisp-site)#`
	`eid-record instance-id 8197 any-mac`
		Configures an IPv4 or IPv6 EID prefix associated with this LISP instance.
		eid-prefix can be IPv4 or IPv6 or MAC EID prefixes.
		accept-more-specifics allows the site to accept registrations for more specific EID prefixes
		• Repeat this step as necessary to configure additional EID prefixes under the LISP site.

Step 42	**allow-locator-default-etr instance-id** *instance-id* { **ipv4** | **ipv6** }
	Example:
	`Device(config-router-lisp-site)#`
	`allow-locator-default-etr instance-id 4097 ipv4`
		Configures the LISP site to accept default egress tunnel router (ETR) registrations for a particular instance-id and a given service level (IPv4 or IPv6) within that instance-id.
		A default ETR handles the unknown EID prefixes, which are the EID prefixes that are not present in the control plane database. A border node that registers with the control plane node as a default ETR tracks the unknown EID prefixes in each of their VRF tables (a given service level within an instance ID).

Step 43	**exit** **Example:** `Device(config-router-lisp-site)#` **`exit`**	Exits the LISP Site configuration mode, and enters LISP configuration mode.
Step 44	**ipv4 locator reachability minimum-mask-length** *length* **Example:** `Device(config-router-lisp)#` **`ipv4 locator reachability minimum-mask-length 32`**	Specifies the shortest mask prefix to accept when looking up a remote RLOC in the RIB. LISP checks the host reachability from the routing locator.
Step 45	**ipv4 source-locator Loopback** *loopback-interface-number* **Example:** `Device(config-router-lisp)#` **`ipv4 source-locator Loopback 0`**	Specifies the interface whose IPv4 address should be used as the source locator address for outbound LISP encapsulated packets.

Step 46	**exit-router-lisp**
	Example:
	`Device(config-router-lisp)#`
	`exit-router-lisp`
		Exits LISP configuration mode, and enters global configuration mode.

	Command or Action	Purpose

Step 47	**end**
	Example:
	`Device(config)# end`
		Returns to privileged EXEC mode.

Configure Layer 3 VNI and Segment for Default Instance

To configure Layer 3 VNI on fabric in a box device, perform this task:

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	**instance-id** *id* **Example:** `Device(config)# instance-id 4097`	Specifies the instance ID.
Step 4	**remote-rloc-probe on-route-change** **Example:** `Device(config-inst)# remote-rloc-probe on-route-change`	Configures parameters for probing of remote local routing locators (RLOCs).
Step 5	**dynamic-eid** *eid-name* **Example:** `Device(config-inst)# dynamic-eid AVlan91-IPV4`	Creates a dynamic End Point Identifier (EID) policy, and enters the dynamic-eid configuration mode on an xTR.
Step 6	**database-mapping** *eid-prefix/prefix-length* **locator-set** *RLOC_name* **Example:** `Device(config-inst-dynamic-eid)# database-mapping 10.91.1.0/24 locator-set rloc_set`	Configures an IPv4 endpoint identifier-to-routing locator (EID-to-RLOC) mapping relationship and an associated traffic policy for LISP.
Step 7	**exit-dynamic-eid** **Example:** `Device(config-inst-dynamic-eid)# exit-dynamic-eid`	Exits dynamic-eid configuration mode, and enters instance configuration mode.

	Command or Action	Purpose
Step 8	**dynamic-eid** *eid-name* **Example:** `Device(config-inst)# dynamic-eid CAMPUS-DATA-FZ3-IPV4`	Creates a dynamic End Point Identifier (EID) policy, and enters the dynamic-eid configuration mode on an xTR.

Step 9	**service** {**ipv4** | **ipv6**}
	Example:
	`Device(config-inst)# service ipv4`
		Enables Layer 3 network services for the IPv4 or IPv6 address family.

Step 10	**eid-table default** **Example:** `Device(config-inst-serv-ipv4)# eid-table default`	Configures the default (global) routing table for association with the configured instance-service.
Step 11	**map-cache** *address* **map-request** **Example:** `Device(config-inst-serv-ipv4)# map-cache 10.91.1.0/24 map-request`	Sends map-request for LISP destination EID.
Step 12	Do one of the following: • **exit-service-ipv4** • **exit-service-ipv6** **Example:** `Device(config-inst-serv-ipv4)# exit-service-ipv4`	Exits service configuration mode, and enters instance configuration mode.
Step 13	**exit-instance-id** **Example:** `Device(config-inst)# exit-instance-id`	Exits instance configuration mode, and enters global configuration mode.

Step 14	**end**
	Example:
	`Device(config)# end`
		Returns to privileged EXEC mode.

Configure Layer 3 VNI for User-Defined VRF

To configure a Layer 3 VNI for user-defined VRF, perform this task.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router lisp** **Example:** `Device(config)# router lisp`	Enters LISP configuration mode.
Step 4	**instance-id** *id* **Example:** `Device(config-router-lisp)# instance-id 4099`	Specifies the instance ID.
Step 5	**remote-rloc-probe on-route-change** **Example:** `Device(config-router-lisp-inst)# remote-rloc-probe on-route-change`	Configures parameters for probing of remote local routing locators (RLOCs).
Step 6	**dynamic-eid** *eid-name* **Example:** `Device(config-router-lisp-inst)# dynamic-eid AVlan50-IPV4`	Creates a dynamic End Point Identifier (EID) policy, and enters the dynamic-eid configuration mode on an xTR.
Step 7	**database-mapping** *eid-prefix/prefix-length* **locator-set** *RLOC_name* **Example:** `Device(config-router-lisp-inst-dynamic-eid)# database-mapping 10.50.1.0/24 locator-set rloc_set`	Configures an IPv4 endpoint identifier-to-routing locator (EID-to-RLOC) mapping relationship and an associated traffic policy for LISP.
Step 8	**exit-dynamic-eid** **Example:** `Device(config-router-lisp-inst-dynamic-eid)# exit-dynamic-eid`	Exits dynamic-eid configuration mode, and enters LISP instance configuration mode.
Step 9	**dynamic-eid** *eid-name* **Example:** `Device(config-router-lisp-inst)# dynamic-eid AVlan50-IPV6`	Creates a dynamic End Point Identifier (EID) policy, and enters the dynamic-eid configuration mode on an xTR.
Step 10	**database-mapping** *eid-prefix/prefix-length* **locator-set** *RLOC_name* **Example:** `Device(config-router-lisp-inst-dynamic-eid)# database-mapping 2001:DB8:2050::/64 locator-set rloc_set`	Configures an IPv4 endpoint identifier-to-routing locator (EID-to-RLOC) mapping relationship and an associated traffic policy for LISP.

	Command or Action	Purpose
Step 11	**exit-dynamic-eid** **Example:** `Device(config-router-lisp-inst-dynamic-eid)#` **`exit-dynamic-eid`**	Exits dynamic-eid configuration mode, and enters LISP instance configuration mode.
Step 12	**service ipv4** **Example:** `Device(config-router-lisp-inst)#` **`service ipv4`**	Enables Layer 3 network services for the IPv4 address family.
Step 13	**eid-table vrf** *vrf-name* **Example:** `Device(config-router-lisp-inst-serv-ipv4)#` **`eid-table vrf VN3`**	Configures the VRF table for association with the configured instance-service.
Step 14	**database-mapping** *eid-prefix/prefix-length* **locator-set** *RLOC_name* **default-etr local** **Example:** `Device(config-router-lisp-inst-serv-ipv4)#` **`database-mapping 0.0.0.0/0 locator-set default_etr_locator default-etr local`**	Configures an IPv4 endpoint identifier-to-routing locator (EID-to-RLOC) mapping relationship and an associated traffic policy for LISP.
Step 15	**exit-service-ipv4** **Example:** `Device(config-router-lisp-inst-serv-ipv4)#` **`exit-service-ipv4`**	Exits service IPv4 configuration mode, and enters LISP instance configuration mode.
Step 16	**service ipv6** **Example:** `Device(config-router-lisp-inst)#` **`service ipv6`**	Enables Layer 3 network services for the IPv6 address family.
Step 17	**eid-table vrf** *vrf-name* **Example:** `Device(config-router-lisp-inst-serv-ipv6)#` **`eid-table vrf VN3`**	Configures the VRF table for association with the configured instance-service.
Step 18	**database-mapping** *eid-prefix/prefix-length* **locator-set** *RLOC_name* **default-etr local** **Example:** `Device(config-router-lisp-inst-serv-ipv6)#` **`database-mapping ::/0 locator-set default_etr_locator default-etr local`**	Configures an IPv6 endpoint identifier-to-routing locator (EID-to-RLOC) mapping relationship and an associated traffic policy for LISP.
Step 19	**exit-service-ipv6** **Example:**	Exits service IPv6 configuration mode, and enters LISP instance configuration mode.

	Command or Action	Purpose
	`Device(config-router-lisp-inst-serv-ipv6)# exit-service-ipv6`	
Step 20	**exit-instance-id** **Example:** `Device(config-router-lisp-inst)# exit-instance-id`	Exits instance configuration mode, and enters LISP configuration mode.

Step 21	**end**
	Example:
	`Device(config-router-lisp)# end`
		Returns to privileged EXEC mode.

Configure Layer 2 VNI for Default Instance

To configure a Layer 2 VNI for a default instance on fabric in a box device, perform this task:

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	**router lisp** **Example:** `Device(config)# router lisp`	Enters LISP configuration mode.
Step 4	**instance-id** *id* **Example:** `Device(config-router-lisp)# instance-id 8194`	Specifies the instance ID.
Step 5	**remote-rloc-probe on-route-change** **Example:** `Device(config-router-lisp-inst)# remote-rloc-probe on-route-change`	Specifies that the probing of remote routing locators (RLOCs) should be done when there is a route change for the remote RLOCs.
Step 6	**service ethernet** **Example:** `Device(config-router-lisp-inst)# service ethernet`	Enables Layer 2 network services.

	Command or Action	Purpose
Step 7	**eid-table vlan** *vlan-id* **Example:** `Device(config-router-lisp-inst-serv-ethernet)# eid-table vlan 91`	Configures the specified VLAN table for association with the configured instance.
Step 8	**database-mapping** *eid-prefix/prefix-length* **locator-set** *RLOC_name* **Example:** `Device(config-inst-serv-ethernet-eid-table)# database-mapping mac locator-set rloc_set`	Configures an IPv4 endpoint identifier-to-routing locator (EID-to-RLOC) mapping relationship and an associated traffic policy for LISP.
Step 9	**exit** **Example:** `Device(config-inst-serv-ethernet-eid-table)# exit`	Exits EID table configuration mode.
Step 10	**exit-service-ethernet** **Example:** `Device(config-inst-serv-ethernet)# exit-service-ethernet`	Exits service Ethernet configuration mode, and enters instance configuration mode.
Step 11	**exit-instance-id** **Example:** `Device(config-inst)# exit-instance-id`	Exits instance configuration mode, and enters global configuration mode.
Step 12	**exit-router-lisp** **Example:** `Device(config-router-lisp)# exit-router-lisp`	Exits LISP configuration mode, and enters global configuration mode.

Step 13	**end**
	Example:
	`Device(config)# end`
		Returns to privileged EXEC mode.

Configure Layer 2 VNI for User-Defined VRF

To configure Layer 2 VNI for user-defined VRF on a fabric in a box device, perform this task:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router lisp** **Example:** `Device(config)# router lisp`	Enters LISP configuration mode.
Step 4	**instance-id** *id* **Example:** `Device(config-router-lisp)# instance-id 8197`	Specifies the instance ID of the user-defined instance.
Step 5	**remote-rloc-probe on-route-change** **Example:** `Device(config-router-lisp-inst)# remote-rloc-probe on-route-change`	Specifies that the probing of remote local routing locators (RLOCs) should be done when there are routing changes for remote RLOCs.
Step 6	**service ethernet** **Example:** `Device(config-router-lisp-inst)# service ethernet`	Enables Layer 2 network services.
Step 7	**eid-table vlan** *vlan-id* **Example:** `Device(config-router-lisp-inst-serv-ethernet)# eid-table vlan 50`	Configures the specified VLAN table for association with the configured instance.
Step 8	**database-mapping** *eid-prefix/prefix-length* **locator-set** *RLOC_name* **Example:** `Device(config-inst-serv-ethernet-eid-table)# database-mapping mac locator-set rloc_set`	Configures an IPv4 endpoint identifier-to-routing locator (EID-to-RLOC) mapping relationship and an associated traffic policy for LISP.
Step 9	**exit** **Example:** `Device(config-inst-serv-ethernet-eid-table)# exit`	Exits EID table configuration mode.
Step 10	**exit-service-ethernet** **Example:** `Device(config-router-lisp-inst-serv-ethernet)# exit-service-ethernet`	Exits service Ethernet configuration mode, and enters instance configuration mode.

	Command or Action	Purpose
Step 11	**exit-instance-id** **Example:** `Device(config-router-lisp-inst)# exit-instance-id`	Exits instance configuration mode, and enters global configuration mode.
Step 12	**exit-router-lisp** **Example:** `Device(config-router-lisp)# exit-router-lisp`	Exits LISP configuration mode, and enters global configuration mode.

Step 13	**end**
	Example:
	`Device(config)# end`
		Returns to privileged EXEC mode.

Configure BGP

To configure BGP on a fabric in a box device, perform this task:

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	**router bgp** *autonomous-system-number* **Example:** `Device(config)# router bgp 700`	Configures a BGP routing process, and enters router configuration mode for the specified routing process. Use the *autonomous-system-number* argument to specify an integer, from 0 and 65534, that identifies the device to other BGP speakers.
Step 4	**bgp router-id** *ip-address* **Example:** `Device(config-router)# bgp router-id interface Loopback0`	(Optional) Configures a fixed 32-bit router ID as the identifier of the local device running BGP. Use the *ip-address* argument to specify a unique router ID within the network. **Note** Configuring a router ID using the **bgp router-id** command resets all active BGP peering sessions.

	Command or Action	Purpose
Step 5	**bgp log-neighbor-changes** **Example:** `Device(config-router)# bgp log-neighbor-changes`	(Optional) Enables logging of BGP neighbor status changes (up or down) and neighbor resets. Use this command for troubleshooting network connectivity problems and measuring network stability. Unexpected neighbor resets might indicate high error rates or high packet loss in the network and should be investigated.
Step 6	**bgp graceful-restart** **Example:** `Device(config-router)# bgp graceful-restart`	Enables the BGP graceful restart capability globally for all BGP neighbors.

Step 7	**address-family** {**ipv4** | **ipv6**}
	Example:
	`Device(config-router)# address-family ipv4`
		Specifies the address family, and enters address family configuration mode.
		• **ipv4**: Specifies the address family as IPv4.
		• **ipv6**: Specifies the address family as IPv6.

Step 8	**bgp aggregate-timer** *seconds* **Example:** `Device(config-router-af)# bgp aggregate-timer 0`	Sets the interval at which BGP routes will be aggregated or to disable timer-based route aggregation.

Step 9	**network** *network-number* **mask** *network-mask*
	Example:
	`Device(config-router-af)# network 10.91.1.0 mask 255.255.255.0`
	`Device(config-router-af)# network 172.16.1.68 mask 255.255.255.255`
		Specifies a network as local to this autonomous system and adds it to the BGP routing table.

Step 10	**aggregate-address** *address mask* **summary-only** **Example:** `Device(config-router-af)# aggregate-address 10.91.1.0 255.255.255.0 summary-only`	Creates an aggregate entry in a BGP database. • **summary-only**: Filters all more-specific routes from updates.
Step 11	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits address family configuration mode, and enters router configuration mode.

	Command or Action	Purpose

Step 12	**address-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*]
	Example:
	`Device(config-router)# address-family ipv4 vrf VN3`
	`Device(config-router)# address-family ipv6 vrf VN3`
		Enters address family configuration mode to configure routing sessions that use address family-specific command configurations.
		Use the **vrf** option to specify the VRF instance with which the subsequent address family configuration commands are associated.

Step 13	**bgp aggregate-timer** *seconds* **Example:** `Device(config-router-af)# bgp aggregate-timer 0`	Configures the interval at which the BGP routes are aggregated. A value of 0 (zero) disables timer-based aggregation and starts aggregation immediately.

Step 14	**network** *network-number* [**mask** *network-mask*] [**route-map** *route-map-name*]
	Example:
	`Device(config-router-af)# network 10.20.1.0 mask 255.255.255.252`
	`Device(config-router-af)# network 10.50.1.0 mask 255.255.255.0`
	
	`Device(config-router-af)# network 2001:DB8:20::/126`
	`Device(config-router-af)# network 2001:DB8:2050::/64`
		Specifies the network to be advertised by BGP and adds it to the BGP routing table.
		• For exterior protocols, the **network** command controls which networks are advertised. Interior protocols use the **network** command to determine where to send updates.

Step 15	**aggregate-address** *address mask* **summary-only**
	Example:
	`Device(config-router-af)# aggregate-address 10.50.1.0 255.255.255.0 summary-only`
	`Device(config-router-af)# aggregate-address 2001:DB8:2050::/64 summary-only`
		Creates an aggregate entry in a BGP database.
		• **summary-only**: Filters all more-specific routes from updates.

Step 16	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits address family configuration mode, and enters router configuration mode.

Step 17	**end**
	Example:
	`Device(config-router)# end`
		Returns to privileged EXEC mode.

Configure Route-Map

To configure a route-map for a fabric in a box device, perform this task:

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	**route-map** *map-name* [**permit** | **deny**] [*sequence-number*] **Example:** `Device(config)# route-map LISP_TO_BGP permit 10`	Configures a route map for the BGP and enters route map configuration mode. Route map entries are read in order. You can identify the order using the *sequence_number* argument.
Step 4	**description** *description* **Example:** `Device(config-route-map)# description prefixes_learnt`	Adds a description for the route map.
Step 5	**set as-path tag** **Example:** `Device(config-route-map)# set as-path tag`	Modifies an autonomous system path for BGP routes.

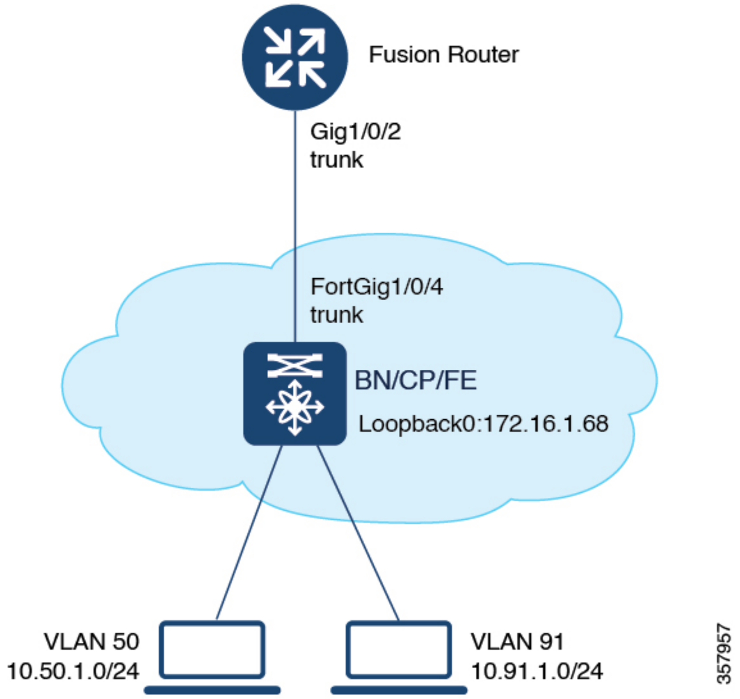
Step 6	**end**
	Example:
	`Device(config-route-map)# end`
		Returns to privileged EXEC mode.

Configuration Example for a Fabric in a Box Device

This example shows a sample configuration for a fabric in a box construct in the LISP VXLAN fabric depicted in the Figure 12: LISP VXLAN Topology for Fabric in a Box.

The topology has a fabric in a box containing an edge node, control plane node, and border node on the same device. The fabric in a box device connects to an upstream router.

Figure 12: LISP VXLAN Topology for Fabric in a Box



Fabric in a Box

```
vrf definition VN3
 rd 1:4099
 !
 address-family ipv4
  route-target export 1:4099
  route-target import 1:4099
 exit-address-family
 !
 address-family ipv6
  route-target export 1:4099
  route-target import 1:4099
 exit-address-family

vlan 222
 name 222
!
interface Vlan222
 description vrf-external
 vrf forwarding VN3
 ip address 10.20.1.1 255.255.255.252
 no ip redirects
 ipv6 address 2001:DB8:20::1/126
 ipv6 enable

!
interface TenGigabitEthernet1/0/4
 switchport mode trunk

device-tracking tracking
!
device-tracking policy IPDT_POLICY
```

```
 no protocol udp
 tracking enable
!

interface TenGigabitEthernet1/0/5
 device-tracking attach-policy IPDT_POLICY
!
 ipv6 nd raguard
 ipv6 dhcp guard
!
vlan 50
 name AVlan50
!
vlan 91
 name AVlan91
!
interface Vlan50
 description server1
 mac-address 0000.0c9f.f18e
 vrf forwarding VN3
 ip address 10.50.1.1 255.255.255.0
 ip helper-address 172.16.2.2
 no ip redirects
 ipv6 address 2001:DB8:2050::1/64
 ipv6 enable
 ipv6 nd dad attempts 0
 ipv6 nd prefix 2001:DB8:2050::/64 2592000 604800 no-autoconfig
 ipv6 nd managed-config-flag
 ipv6 nd other-config-flag
 ipv6 nd router-preference High
 ipv6 dhcp relay destination 2001:DB8:2::2
 ipv6 dhcp relay source-interface Vlan50
 ipv6 dhcp relay trust
 no lisp mobility liveness test
 lisp mobility AVlan50-IPV4
 lisp mobility AVlan50-IPV6
 no autostate
!
interface Vlan91
 description default-interface
 mac-address 0000.0c9f.f984
 ip address 10.91.1.1 255.255.255.0
 ip helper-address 172.16.2.2
 no ip redirects
 no lisp mobility liveness test
 lisp mobility AVlan91-IPV4
 no autostate
!
ip dhcp relay information option
ip dhcp snooping vlan 50,91
ip dhcp snooping

router lisp
 locator-table default
 locator-set default_etr_locator
  IPv4-interface Loopback0 priority 10 weight 10
  exit-locator-set
 !
 locator-set rloc_set
  IPv4-interface Loopback0 priority 10 weight 10
  auto-discover-rlocs
  exit-locator-set
 !
 locator default-set rloc_set
```

```
service ipv4
 encapsulation vxlan
 map-cache publications
 import publication publisher 172.16.1.68
 itr map-resolver 172.16.1.68
 etr map-server 172.16.1.68 key 7 auth-key
 etr map-server 172.16.1.68 proxy-reply
 etr
 sgt
 route-export publications
 distance publications 250
 proxy-etr
 proxy-itr 172.16.1.68
 map-server
 map-resolver
 exit-service-ipv4
!
service ipv6
 encapsulation vxlan
 map-cache publications
 import publication publisher 172.16.1.68
 itr map-resolver 172.16.1.68
 etr map-server 172.16.1.68 key 7 auth-key
 etr map-server 172.16.1.68 proxy-reply
 etr
 sgt
 route-export publications
 distance publications 250
 proxy-etr
 proxy-itr 172.16.1.68
 map-server
 map-resolver
 exit-service-ipv6
!
service ethernet
 itr map-resolver 172.16.1.68
 itr
 etr map-server 172.16.1.68 key 7 auth-key
 etr map-server 172.16.1.68 proxy-reply
 etr
 map-server
 map-resolver
 exit-service-ethernet
!

instance-id 4097
 remote-rloc-probe on-route-change
 dynamic-eid AVlan91-IPV4
  database-mapping 10.91.1.0/24 locator-set rloc_set
  exit-dynamic-eid
 !
 service ipv4
  eid-table default
  map-cache 10.91.1.0/24 map-request
  exit-service-ipv4
 !
 exit-instance-id
!


instance-id 4099
 remote-rloc-probe on-route-change
 dynamic-eid AVlan50-IPV4
  database-mapping 10.50.1.0/24 locator-set rloc_set
```

```
   exit-dynamic-eid
  !
  dynamic-eid AVlan50-IPV6
   database-mapping 2001:DB8:2050::/64 locator-set rloc_set
   exit-dynamic-eid
  !
  service ipv4
   eid-table vrf VN3
   database-mapping 0.0.0.0/0 locator-set default_etr_local default-etr local
   exit-service-ipv4
  !
  service ipv6
   eid-table vrf VN3
   database-mapping ::/0 locator-set default_etr_local default-etr local
   exit-service-ipv6
  !
  exit-instance-id
 !
 !


 instance-id 8194
  remote-rloc-probe on-route-change
  service ethernet
   eid-table vlan 91
   database-mapping mac locator-set rloc_set
   exit-service-ethernet
  !
  exit-instance-id
 !
 !
 instance-id 8197
  remote-rloc-probe on-route-change
  service ethernet
   eid-table vlan 50
   database-mapping mac locator-set rloc_set
   exit-service-ethernet
  !
  exit-instance-id
 !
 !
 site site_uci
  description map-server1
  authentication-key 7 auth-key
  eid-record instance-id 4097 0.0.0.0/0 accept-more-specifics
  eid-record instance-id 4097 10.91.1.0/24 accept-more-specifics
  eid-record instance-id 4099 0.0.0.0/0 accept-more-specifics
  eid-record instance-id 4099 10.50.1.0/24 accept-more-specifics
  eid-record instance-id 4099 ::/0 accept-more-specifics
  eid-record instance-id 4099 2001:DB8:2050::/64 accept-more-specifics
  eid-record instance-id 8194 any-mac
  eid-record instance-id 8197 any-mac
  allow-locator-default-etr instance-id 4097 ipv4
  allow-locator-default-etr instance-id 4099 ipv4
  allow-locator-default-etr instance-id 4099 ipv6
  exit-site
 !
 ipv4 locator reachability minimum-mask-length 32
 ipv4 source-locator Loopback0
 exit-router-lisp
!
router bgp 700
 bgp router-id interface Loopback0
 bgp log-neighbor-changes
 bgp graceful-restart
 !
 address-family ipv4
  bgp redistribute-internal
  bgp aggregate-timer 0
  network 10.91.1.0 mask 255.255.255.0
  network 172.16.1.68 mask 255.255.255.255
  aggregate-address 10.91.1.0 255.255.255.0 summary-only
  redistribute lisp metric 10 route-map LISP_TO_BGP
 exit-address-family
 !
 !
 address-family ipv4 vrf VN3
  bgp aggregate-timer 0
  network 10.20.1.0 mask 255.255.255.252
  network 10.50.1.0 mask 255.255.255.0
  aggregate-address 10.50.1.0 255.255.255.0 summary-only
  redistribute lisp metric 10 route-map LISP_TO_BGP
 exit-address-family
 !
 address-family ipv6 vrf VN3
  redistribute lisp metric 10 route-map LISP_TO_BGP
  bgp aggregate-timer 0
  network 2001:DB8:20::/126
  network 2001:DB8:2050::/64
  aggregate-address 2001:DB8:2050::/64 summary-only
 exit-address-family
!
!
route-map LISP_TO_BGP permit 10
 description prefixes_learnt
 set as-path tag
!
```

Verify Fabric in a Box

This section provides sample outputs for the **show** commands on the fabric edge nodes in the topology shown Figure 12: LISP VXLAN Topology for Fabric in a Box. In the topology, 172.16.1.68 is the loopback0 of the fabric in a box device. VLAN 50 has a subnet of 10.50.1.0/24 and VLAN 91 has a subnet of 10.91.1.0/24.

```
FabricInABox# show ip interface brief | i LISP
L2LISP0                172.16.1.68       YES unset  up                    up
L2LISP0.8194           172.16.1.68       YES unset  up                    up
L2LISP0.8197           172.16.1.68       YES unset  up                    up
LISP0                  unassigned        YES unset  up                    up
LISP0.4097             172.16.1.68       YES unset  up                    up
LISP0.4099             10.50.1.1         YES unset  up                    up
FabricInABox#


FabricInABox# show lisp session

Sessions for VRF default, total: 3, established: 2
Peer                           State      Up/Down        In/Out    Users
172.16.1.68:4342                  Up         03:37:52       38/23     11
172.16.1.68:24737


FabricInABox# show lisp session 172.16.1.68 port 4342

Peer address:     172.16.1.68:4342
Local address:    172.16.1.68:24737
```

```
Session Type:     Active
Session State:    Up (03:40:02)
Messages in/out:  38/23
Bytes in/out:     1830/1676
Fatal errors:     0
Rcvd unsupported: 0
Rcvd invalid VRF: 0
Rcvd override:    0
Rcvd malformed:   0
Sent deferred:    0
SSO redundancy:   N/A
Auth Type:        None

Accepting Users:  0
Users:            11
  Type                      ID                                      In/Out    State
  Pubsub subscriber         lisp 0 IID 4097 AFI IPv4                 3/2      Established
  ETR Reliable Registration lisp 0 IID 16777214 AFI IPv4             2/2      TCP
  ETR Reliable Registration lisp 0 IID 4099 AFI IPv4                 3/3      TCP
  Pubsub subscriber         lisp 0 IID 4099 AFI IPv4                 6/2      Established
  ETR Reliable Registration lisp 0 IID 4099 AFI IPv6                 3/3      TCP
  Pubsub subscriber         lisp 0 IID 4099 AFI IPv6                 6/2      Established
  ETR Reliable Registration lisp 0 IID 8194 AFI MAC                  2/4      TCP
  Pubsub subscriber         lisp 0 IID 8194 AFI MAC                  2/0      Off
  ETR Reliable Registration lisp 0 IID 8197 AFI MAC                  2/4      TCP
  Pubsub subscriber         lisp 0 IID 8197 AFI MAC                  2/0      Off
  Capability Exchange       N/A                                      1/1      waiting
FabricInABox#


FabricInABox#show lisp session 172.16.1.68 port 24737

Peer address:     172.16.1.68:24737
Local address:    172.16.1.68:4342
Session Type:     Passive
Session State:    Up (03:44:54)
Messages in/out:  23/38
Bytes in/out:     1676/1830
Fatal errors:     0
Rcvd unsupported: 0
Rcvd invalid VRF: 0
Rcvd override:    0
Rcvd malformed:   0
Sent deferred:    1
SSO redundancy:   synchronized
Auth Type:        None

Accepting Users:  1
Users:            9
  Type                      ID                                      In/Out    State
  Capability Exchange       N/A                                      1/1      waiting
  Pubsub publisher          lisp 0 IID 4097 AFI IPv4                 2/2      working
  Pubsub publisher          lisp 0 IID 4099 AFI IPv4                 2/5      working
  Pubsub publisher          lisp 0 IID 4099 AFI IPv6                 2/5      working
  MS Reliable Registration  lisp 0 IID 16777214 AFI IPv4             2/2      waiting
    WLC subscription received
  MS Reliable Registration  lisp 0 IID 4099 AFI IPv4                 2/3      waiting
    WLC subscription received
  MS Reliable Registration  lisp 0 IID 4099 AFI IPv6                 2/3      waiting
    WLC subscription received
  MS Reliable Registration  lisp 0 IID 8194 AFI MAC                  2/2      waiting
    WLC subscription received
  MS Reliable Registration  lisp 0 IID 8197 AFI MAC                  2/2      waiting
    WLC subscription received
```

```
FabricInABox#

FabricInABox# show lisp site
LISP Site Registration Information
* = Some locators are down or unreachable
# = Some registrations are sourced by reliable transport

Site Name      Last      Up     Who Last             Inst     EID Prefix
               Register         Registered           ID
site_uci       never     no     --                   4097     0.0.0.0/0
               never     no     --                   4097     10.91.1.0/24
               never     no     --                   4099     0.0.0.0/0
               never     no     --                   4099     10.50.1.0/24
               never     no     --                   4099     ::/0
               never     no     --                   4099     2001:DB8:2050::/64
FabricInABox#

FabricInABox# show lisp site name site_uci
Site name: site_uci
Description: <description>
Allowed configured locators: any
Allowed EID-prefixes:

  EID-prefix: 0.0.0.0/0 instance-id 4097
    First registered:     never
    Last registered:      never
    Routing table tag:    0
    Origin:               Configuration, accepting more specifics
    Merge active:         No
    Proxy reply:          No
    Skip Publication:     No
    Force Withdraw:       No
    TTL:                  00:00:00
    State:                unknown
    Extranet IID:         Unspecified
    Registration errors:
      Authentication failures:   0
      Allowed locators mismatch: 0
    No registrations.

  EID-prefix: 10.91.1.0/24 instance-id 4097
    First registered:     never
    Last registered:      never
    Routing table tag:    0
    Origin:               Configuration, accepting more specifics
    Merge active:         No
    Proxy reply:          No
    Skip Publication:     No
    Force Withdraw:       No
    TTL:                  00:00:00
    State:                unknown
    Extranet IID:         Unspecified
    Registration errors:
      Authentication failures:   0
      Allowed locators mismatch: 0
    No registrations.

  EID-prefix: 0.0.0.0/0 instance-id 4099
    First registered:     never
    Last registered:      never
    Routing table tag:    0
    Origin:               Configuration, accepting more specifics
    Merge active:         No
    Proxy reply:          No
    Skip Publication:     No
    Force Withdraw:       No
    TTL:                  00:00:00
    State:                unknown
    Extranet IID:         Unspecified
    Registration errors:
      Authentication failures:   0
      Allowed locators mismatch: 0
    No registrations.

  EID-prefix: 10.50.1.0/24 instance-id 4099
    First registered:     never
    Last registered:      never
    Routing table tag:    0
    Origin:               Configuration, accepting more specifics
    Merge active:         No
    Proxy reply:          No
    Skip Publication:     No
    Force Withdraw:       No
    TTL:                  00:00:00
    State:                unknown
    Extranet IID:         Unspecified
    Registration errors:
      Authentication failures:   0
      Allowed locators mismatch: 0
    No registrations.

  EID-prefix: ::/0 instance-id 4099
    First registered:     never
    Last registered:      never
    Routing table tag:    0
    Origin:               Configuration, accepting more specifics
    Merge active:         No
    Proxy reply:          No
    Skip Publication:     No
    Force Withdraw:       No
    TTL:                  00:00:00
    State:                unknown
    Extranet IID:         Unspecified
    Registration errors:
      Authentication failures:   0
      Allowed locators mismatch: 0
    No registrations.

  EID-prefix: 2001:DB8:2050::/64 instance-id 4099
    First registered:     never
    Last registered:      never
    Routing table tag:    0
    Origin:               Configuration, accepting more specifics
    Merge active:         No
    Proxy reply:          No
    Skip Publication:     No
    Force Withdraw:       No
    TTL:                  00:00:00
    State:                unknown
    Extranet IID:         Unspecified
    Registration errors:
      Authentication failures:   0
      Allowed locators mismatch: 0
    No registrations.
FabricInABox#
```

```
FabricInABox# show lisp instance-id 4099 ipv4 database
LISP ETR IPv4 Mapping Database for LISP 0 EID-table vrf VN3 (IID 4099), LSBs: 0x1
Entries total 2, no-route 1, inactive 0, do-not-register 1

0.0.0.0/0, locator-set DEFAULT_ETR_LOCATOR *** NO ROUTE TO EID PREFIX ***, default-ETR
  Uptime: 03:48:45, Last-change: 03:48:45
  Domain-ID: local
  Metric: -
  Service-Insertion: N/A
  Locator   Pri/Wgt  Source     State
  172.16.1.68   10/10   cfg-intf   site-self, reachable
10.50.1.1/32, dynamic-eid AVlan50-IPV4, do not register, inherited from default locator-set
 rloc_set1, auto-discover-rlocs
  Uptime: 03:33:23, Last-change: 03:33:23
  Domain-ID: local
  Service-Insertion: N/A
  Locator   Pri/Wgt  Source     State
  172.16.1.68   10/10   cfg-intf   site-self, reachable
FabricInABox#
```

```
FabricInABox# show lisp instance-id 4099 ipv4 map-cache
LISP IPv4 Mapping Cache for LISP 0 EID-table vrf VN3 (IID 4099), 4 entries

0.0.0.0/0, uptime: 00:00:00, expires: 00:00:59, via away, send-map-request
  Negative cache entry, action: send-map-request
10.0.0.0/11, uptime: 03:47:45, expires: 00:09:16, via map-reply, forward-native
  Negative cache entry, action: forward-native
10.50.1.0/24, uptime: 03:49:03, expires: never, via dynamic-EID, send-map-request
  Negative cache entry, action: send-map-request
128.0.0.0/1, uptime: 03:48:45, expires: 00:09:03, via map-reply, forward-native
  Negative cache entry, action: forward-native
FabricInABox#
```

```
FabricInABox# show lisp instance-id 8194 ethernet database
LISP ETR MAC Mapping Database for LISP 0 EID-table Vlan 91 (IID 8194), LSBs: 0x1
Entries total 2, no-route 0, inactive 0, do-not-register 2

0000.0c9f.f984/48, dynamic-eid Auto-L2-group-8194, do not register, inherited from default
 locator-set rloc_set1, auto-discover-rlocs
  Uptime: 03:39:05, Last-change: 03:39:05
  Domain-ID: local
  Service-Insertion: N/A
  Locator   Pri/Wgt  Source     State
  172.16.1.68   10/10   cfg-intf   site-self, reachable
ec1d.8b0a.b6d9/48, dynamic-eid Auto-L2-group-8194, do not register, inherited from default
 locator-set rloc_set1, auto-discover-rlocs
  Uptime: 03:39:07, Last-change: 03:39:07
  Domain-ID: local
  Service-Insertion: N/A
  Locator   Pri/Wgt  Source     State
  172.16.1.68   10/10   cfg-intf   site-self, reachable
FabricInABox#
```

```
FabricInABox# show lisp instance-id 8197 ethernet database
LISP ETR MAC Mapping Database for LISP 0 EID-table Vlan 50 (IID 8197), LSBs: 0x1
Entries total 2, no-route 0, inactive 0, do-not-register 2

0000.0c9f.f18e/48, dynamic-eid Auto-L2-group-8197, do not register, inherited from default
 locator-set rloc_set1, auto-discover-rlocs
  Uptime: 03:39:48, Last-change: 03:39:48
  Domain-ID: local
  Service-Insertion: N/A
  Locator   Pri/Wgt  Source     State
  172.16.1.68   10/10   cfg-intf   site-self, reachable
ec1d.8b0a.b6e8/48, dynamic-eid Auto-L2-group-8197, do not register, inherited from default
 locator-set rloc_set1, auto-discover-rlocs
  Uptime: 03:39:50, Last-change: 03:39:50
  Domain-ID: local
  Service-Insertion: N/A
  Locator   Pri/Wgt  Source     State
  172.16.1.68   10/10   cfg-intf   site-self, reachable
FabricInABox#
```

```
FabricInABox# show lisp vrf VN3 route
 Route prefix                             In RIB Sources
 10.50.1.1/32                             No     Dynamic EID
 2001:DB8:2050::1/128                     No     Dynamic EID
FabricInABox#
```

CHAPTER 9

Configuring Fabric In A Box With Embedded Wireless Controller

Fabric in a Box is a single device that is configured as a border node, a control plane node, an edge node. This single device also supports an embedded wireless controller.

The following platforms support Cisco Catalyst 9800 Embedded Wireless Controller for a fabric in a box deployment:

- Cisco Catalyst 9300 Series Switches
- Cisco Catalyst 9400 Series Switches
- Cisco Catalyst 9500 Series Switches

This chapter describes only the configurations that are required to add the wireless functionality in an existing fabric in a box topology for wired endpoints.

Prerequisites for Configuring Fabric in a Box with Embedded Wireless

- Ensure that the Fabric in a Box device is already configured as edge, border, and control plane nodes for wired endpoints.

  For configuration details, refer to How to Configure Fabric in a Box.

- A Fabric in a Box device should operate in Install mode for a wireless package to be installed. You can install Cisco Catalyst 9800 Series Wireless Controller as a sub-package on top of the base image on the switch.

  Ensure that the wireless package is the same version as the base image on the switch (Cisco IOS XE) . For example, if the switch is operating on Cisco IOS XE 17.10.1, install the 17.10.1 version of the wireless package on the switch.

To download a wireless package, go to the Software Download page, navigate to the switch family, and select the **IOS XE Wireless Controller Software Package** Software Type.

For information on booting a switch in Install mode and installing a sub-package, refer to Cisco Catalyst 9800 Series Wireless Controller Software Configuration Guide.

How to Configure Fabric In A Box with Embedded Wireless

Perform the following procedure to enable wireless functionality in a fabric in a box.

Procedure

Step 1 Enable wireless controller on the switch. Configure the wireless management interface (WMI) as a loopback interface. The WMI is used for all the CAPWAP messages between the wireless controller and the fabric APs.

```
wireless-controller
wireless management interface Loopback0
```

Step 2 Configure a Switched Virtual Interface (SVI) for the AP VLAN.

Note Ensure that you assign the same MAC address for a given SVI, across all fabric edges within the fabric site. We recommend that you use a MAC address starting from the base range value of 0000.0C9F.F05F.

```
interface Vlan92
 description AP SVI
 mac-address 0000.0c9f.f42a   <--- Common MAC Address
 ip address 10.92.1.1 255.255.255.0
 ip helper-address 192.168.132.1
 no ip redirects
 no lisp mobility liveness test
 lisp mobility APVlan92-IPV4
 no autostate
!
```

Step 3 Configure an SVI for the Wireless Client VLAN.

Note Ensure that you assign the same MAC address for a given SVI, across all fabric edges within the fabric site. We recommend that you use a MAC address starting from the base range value of 0000.0C9F.F05F.

```
interface Vlan51
 description Client SVI
 mac-address 0000.0c9f.f7df    <-- Common MAC Address
 vrf forwarding VN4
 ip address 10.51.1.1 255.255.255.0
 ip helper-address 192.168.132.1
 no ip redirects
 no lisp mobility liveness test
 lisp mobility wireless-VN-IPV4
 no autostate
!
```

Step 4 Define a Locator set for the wireless controller.

```
router lisp
 ...
 locator-table default
 locator-set WLC
  192.168.99.1  //IP address of the WMI
  exit-locator-set
 !
```

Step 5 Configure open passive TCP sockets on the control plane node to listen for incoming connections.

```
map-server session passive-open WLC
```

Step 6 Configure the LISP Site to accept EID prefixes.

```
...
site site_uci
  description map-server1
  authentication-key 7 auth-key
  eid-record instance-id 4097 10.51.1.0/24 accept-more-specifics
  eid-record instance-id 4098 10.92.1.0/24 accept-more-specifics
  eid-record instance-id 8188 any-mac
  eid-record instance-id 8189 any-mac
  exit-site
 !
```

Step 7 Configure dynamic EID for the AP subnets in the default instance.

```
...
  instance-id 4097
  remote-rloc-probe on-route-change
  dynamic-eid APVlan92-IPV4
    database-mapping 10.92.1.0/24 locator-set rloc_set
   exit-dynamic-eid
  !
 exit-instance-id
 !
```

Step 8 Configure dynamic EID for the wireless client subnets in the user-defined instance that is mapped to a VRF.

```
...
 instance-id 4098
  remote-rloc-probe on-route-change
  dynamic-eid wireless-VN-IPV4
   database-mapping 10.51.1.0/24 locator-set rloc_set
   exit-dynamic-eid
  !
 exit-instance-id
 !
```

Step 9 Configure Layer 2 VNI for the wireless client VLAN.

```
...
 instance-id 8188
   remote-rloc-probe on-route-change
   service ethernet
     eid-table vlan 51
     database-mapping mac locator-set rloc_set
   exit-service-ethernet
  !
  exit-instance-id
 !
```

Step 10 Configure Layer 2 VNI for the AP VLAN.

```
...
 instance-id 8189
  remote-rloc-probe on-route-change
  service ethernet
   eid-table vlan 92
   database-mapping mac locator-set rloc_set
   exit-service-ethernet
  !
  exit-instance-id
 !
exit-router-lisp
!
```

Step 11 Enable fabric operations on the wireless controller. The following table describes the commands that configure an embedded wireless controller for fabric operations.

Step	Command or Action	Description
a)	**wireless fabric** **Example**: `Switch(config)# wireless fabric`	Enables the wireless functionality on the switch.

b)	**wireless fabric control-plane** *cp-name*
	Example:
	`Switch(config)# wireless fabric control-plane`
	`default-control-plane`
		Configures the name of the fabric control plane.
		You can assign a name of your choice to the control plane.

c)	**ip address** *cp-ip address* **key** *authentication-key*
	Example:
	`Switch(config-wireless-cp)# ip address 172.16.1.68 key 0 some-key`
	`Switch(config-wireless-cp)# end`
		Configures the IP address of the control plane and the authentication key shared with the control plane.

d)	**wireless fabric name** *fabric-name* **l2-vnid** *l2-vnid* **control-plane-name** *cp-name*
	Example:
	`Switch(config)# wireless fabric name wireless-VN-IPV4 l2-vnid`
	`8188`
	`control-plane-name default-control-plane`
		Registers the wireless client VLAN with the control plane.

e)	**wireless fabric name** *fabric-name* **l2-vnid** *l2-instance-id* **l3-vnid** *l3-instance-id* **control-plane-name** *cp-name*
	Example:
	`Switch(config)# wireless fabric name APVlan92-IPV4 l2-vnid 8189`
	` l3-vnid 4097`
	`ip 10.92.1.1 255.255.255.0 control-plane-name`
	`default-control-plane`
		Registers the AP VLAN with the control plane.

Step	Command or Action	Description

f)	**wlan** *wlan-name wlan-id SSID-name*
	Example:
	`Switch(config)# wlan kFiab-local-open_profile 17 kFiab-local-open`
	`Switch(config-wlan)# no shutdown`
	`Switch(config-wlan)#end`
		Configures a WLAN.
		This example configures a WLAN with an ID of 17 and an SSID named kFiab-local-open. It also enables the WLAN using the **no shutdown** command.

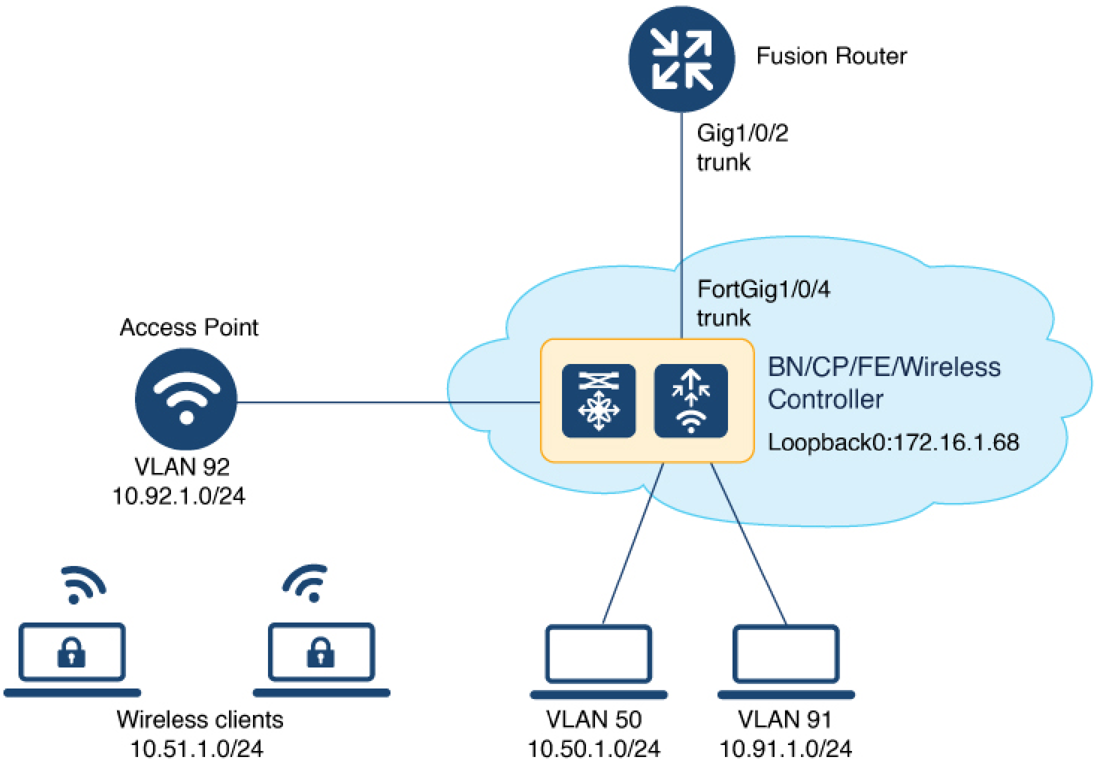
g)	**wireless profile fabric** *profile-policy*
	Example:
	`Switch(config)# wireless profile fabric kFiab-local-open_profile`
	`Switch(config-wireless-fabric)# description local-open-profile`
	`Switch(config-wireless-fabric)# client-l2-vnid 8188`
	`Switch(config-wireless-fabric)# end`
		Configures a fabric profile.
		This example creates a fabric profile named kFiab-local-open_profile and associates the Layer 2 VNI (8188) with the fabric profile.

h)	**wireless profile policy** *profile-policy*
	Example:
	`Switch(config)# wireless profile policy kFiab-local-open_profile`
	`// Specify local DHCP mode`
	` Switch(config-wireless-policy)# no central dhcp`
	`// Configure WLAN for local switching`
	` Switch(config-wireless-policy)# no central switching`
	`//Provide a description for the wireless policy`
	` Switch(config-wireless-policy)# description`
	`kFiab-local-open_profile`
	`//Map the fabric profile that was created in the previous step`
	` Switch(config-wireless-policy)# fabric kFiab-local-open_profile`
	`//Enable the profile policy`
	` Switch(config-wireless-policy)# no shutdown`
	`Switch(config-wireless-policy)# end`
		Configures a wireless policy profile and maps the fabric profile to it.
		The example configures a wireless profile policy named kFiab-local-open_profile and maps a fabric profile to it, using the **fabric** *profile-policy* command.
		You can configure more wireless and fabric profiles as shown in *Configuration Example for Fabric In A Box with Embedded Wireless*.

Configuration Example for Fabric In A Box with Embedded Wireless

This example shows a sample configuration for a fabric in a box construct in the LISP VXLAN fabric depicted in the topology. The fabric in a box device is a Cisco Catalyst 9000 Series switch that functions as a control plane node, border node, edge node, and wireless controller. The loopback IPv4 address of switch is 172.16.1.68. A fabric-capable Access Point (AP) with a subnet of 10.92.1.0/24 is connected to the fabric edge node interface.

Figure 13: LISP VXLAN Topology for Fabric in a Box with Embedded Wireless



Fabric in a Box

```
wireless-controller
wireless management interface Loopback0
!
vrf definition VN4
 rd 1:4098
 !
 address-family ipv4
  route-target export 1:4098
  route-target import 1:4098
 exit-address-family
!
interface Vlan92
 description AP SVI
 mac-address 0000.0c9f.f42a
 ip address 10.92.1.1 255.255.255.0
 ip helper-address 192.168.132.1
 no ip redirects
 no lisp mobility liveness test
 lisp mobility APVlan92-IPV4
 no autostate
!
interface Vlan51
 description Client SVI
 mac-address 0000.0c9f.f7df
 vrf forwarding VN4
 ip address 10.51.1.1 255.255.255.0
 ip helper-address 192.168.132.1
 no ip redirects
 no lisp mobility liveness test
 lisp mobility wireless-VN-IPV4
 no autostate
!

vrf definition VN3
 rd 1:4099
 !
 address-family ipv4
  route-target export 1:4099
  route-target import 1:4099
 exit-address-family
 !
 address-family ipv6
  route-target export 1:4099
  route-target import 1:4099
 exit-address-family

vlan 222
 name 222
!
interface Vlan222
 description vrf-external
 vrf forwarding VN3
 ip address 10.20.1.1 255.255.255.252
 no ip redirects
 ipv6 address 2001:DB8:20::1/126
 ipv6 enable

!
interface TenGigabitEthernet1/0/4
 switchport mode trunk

device-tracking tracking
!
device-tracking policy IPDT_POLICY
 no protocol udp
 tracking enable
!

interface TenGigabitEthernet1/0/5
 device-tracking attach-policy IPDT_POLICY
!
 ipv6 nd raguard
 ipv6 dhcp guard
!
vlan 50
 name AVlan50
!
vlan 91
 name AVlan91
!
interface Vlan50
 description server1
 mac-address 0000.0c9f.f18e
 vrf forwarding VN3
 ip address 10.50.1.1 255.255.255.0
 ip helper-address 172.16.2.2
 no ip redirects
 ipv6 address 2001:DB8:2050::1/64
 ipv6 enable
 ipv6 nd dad attempts 0
 ipv6 nd prefix 2001:DB8:2050::/64 2592000 604800 no-autoconfig
 ipv6 nd managed-config-flag
 ipv6 nd other-config-flag
 ipv6 nd router-preference High
 ipv6 dhcp relay destination 2001:DB8:2::2
 ipv6 dhcp relay source-interface Vlan50
 ipv6 dhcp relay trust
 no lisp mobility liveness test
 lisp mobility AVlan50-IPV4
 lisp mobility AVlan50-IPV6
 no autostate
!
interface Vlan91
 description default-interface
 mac-address 0000.0c9f.f984
 ip address 10.91.1.1 255.255.255.0
 ip helper-address 172.16.2.2
 no ip redirects
 no lisp mobility liveness test
 lisp mobility AVlan91-IPV4
 no autostate
!
ip dhcp relay information option
ip dhcp snooping vlan 50,91
ip dhcp snooping

router lisp
 locator-table default
 locator-set default_etr_locator
  IPv4-interface Loopback0 priority 10 weight 10
  exit-locator-set
 !
 locator-set rloc_set
  IPv4-interface Loopback0 priority 10 weight 10
  auto-discover-rlocs
  exit-locator-set
 !
 locator-set WLC
  192.168.99.1
  exit-locator-set
 !
 locator default-set rloc_set
 service ipv4
  encapsulation vxlan
  map-cache publications
  import publication publisher 172.16.1.68
  itr map-resolver 172.16.1.68
  etr map-server 172.16.1.68 key 7 auth-key
  etr map-server 172.16.1.68 proxy-reply
  etr
  sgt
  route-export publications
  distance publications 250
  proxy-etr
  proxy-itr 172.16.1.68
  map-server
  map-resolver
  exit-service-ipv4
 !
 service ipv6
  encapsulation vxlan
  map-cache publications
  import publication publisher 172.16.1.68
  itr map-resolver 172.16.1.68
  etr map-server 172.16.1.68 key 7 auth-key
  etr map-server 172.16.1.68 proxy-reply
  etr
  sgt
  route-export publications
  distance publications 250
  proxy-etr
  proxy-itr 172.16.1.68
  map-server
  map-resolver
  exit-service-ipv6
 !
 service ethernet
  itr map-resolver 172.16.1.68
  itr
  etr map-server 172.16.1.68 key 7 auth-key
  etr map-server 172.16.1.68 proxy-reply
  etr
  map-server
  map-resolver
  exit-service-ethernet
 !

 instance-id 4097
  remote-rloc-probe on-route-change
  dynamic-eid AVlan91-IPV4
   database-mapping 10.91.1.0/24 locator-set rloc_set
   exit-dynamic-eid
  !
  dynamic-eid APVlan92-IPV4
    database-mapping 10.92.1.0/24 locator-set rloc_set
   exit-dynamic-eid
  !
  service ipv4
   eid-table default
   map-cache 10.91.1.0/24 map-request
   exit-service-ipv4
  !
  exit-instance-id
 !

 instance-id 4099
  remote-rloc-probe on-route-change
  dynamic-eid AVlan50-IPV4
   database-mapping 10.50.1.0/24 locator-set rloc_set
   exit-dynamic-eid
  !
  dynamic-eid AVlan50-IPV6
   database-mapping 2001:DB8:2050::/64 locator-set rloc_set
   exit-dynamic-eid
  !
  dynamic-eid wireless-VN-IPV4
   database-mapping 10.51.1.0/24 locator-set rloc_set
   exit-dynamic-eid
  !
  service ipv4
   eid-table vrf VN3
   database-mapping 0.0.0.0/0 locator-set default_etr_local default-etr local
   exit-service-ipv4
  !
  service ipv6
   eid-table vrf VN3
   database-mapping ::/0 locator-set default_etr_local default-etr local
   exit-service-ipv6
  !
  exit-instance-id
 !
 !
 instance-id 8194
  remote-rloc-probe on-route-change
  service ethernet
   eid-table vlan 91
   database-mapping mac locator-set rloc_set
   exit-service-ethernet
  !
  exit-instance-id
 !
 !
 instance-id 8197
  remote-rloc-probe on-route-change
  service ethernet
   eid-table vlan 50
   database-mapping mac locator-set rloc_set
   exit-service-ethernet
  !
  exit-instance-id
 !
 instance-id 8188
  remote-rloc-probe on-route-change
  service ethernet
   eid-table vlan 92
   database-mapping mac locator-set rloc_set
   exit-service-ethernet
  !
  exit-instance-id
 !
 instance-id 8189
   remote-rloc-probe on-route-change
   service ethernet
     eid-table vlan 51
     database-mapping mac locator-set rloc_set
   exit-service-ethernet
  !
  exit-instance-id
 !
 !
 map-server session passive-open WLC
 site site_uci
  description map-server1
  authentication-key 7 auth-key
  eid-record instance-id 4097 0.0.0.0/0 accept-more-specifics
  eid-record instance-id 4097 10.91.1.0/24 accept-more-specifics
  eid-record instance-id 4097 10.51.1.0/24 accept-more-specifics
  eid-record instance-id 4098 10.92.1.0/24 accept-more-specifics
  eid-record instance-id 4099 0.0.0.0/0 accept-more-specifics
  eid-record instance-id 4099 10.50.1.0/24 accept-more-specifics
  eid-record instance-id 4099 ::/0 accept-more-specifics
  eid-record instance-id 4099 2001:DB8:2050::/64 accept-more-specifics
  eid-record instance-id 8194 any-mac
  eid-record instance-id 8197 any-mac
  eid-record instance-id 8188 any-mac
  eid-record instance-id 8189 any-mac
  allow-locator-default-etr instance-id 4097 ipv4
  allow-locator-default-etr instance-id 4099 ipv4
  allow-locator-default-etr instance-id 4099 ipv6
  exit-site
 !
```

```
 ipv4 locator reachability minimum-mask-length 32
 ipv4 locator reachability exclude-default
 ipv4 source-locator Loopback0
 exit-router-lisp
!
router bgp 700
 bgp router-id interface Loopback0
 bgp log-neighbor-changes
 bgp graceful-restart
 !
 address-family ipv4
  bgp redistribute-internal
  bgp aggregate-timer 0
  network 10.91.1.0 mask 255.255.255.0
  network 172.16.1.68 mask 255.255.255.255
  aggregate-address 10.91.1.0 255.255.255.0 summary-only
  redistribute lisp metric 10 route-map LISP_TO_BGP
 exit-address-family
 !
 !
 address-family ipv4 vrf VN3
  bgp aggregate-timer 0
  network 10.20.1.0 mask 255.255.255.252
  network 10.50.1.0 mask 255.255.255.0
  aggregate-address 10.50.1.0 255.255.255.0 summary-only
  redistribute lisp metric 10 route-map LISP_TO_BGP
 exit-address-family
 !
 address-family ipv6 vrf VN3
  redistribute lisp metric 10 route-map LISP_TO_BGP
  bgp aggregate-timer 0
  network 2001:DB8:20::/126
  network 2001:DB8:2050::/64
  aggregate-address 2001:DB8:2050::/64 summary-only
 exit-address-family
!
 address-family ipv4 vrf VN4
  bgp aggregate-timer 0
  network 10.51.1.0 mask 255.255.255.0
  aggregate-address 10.51.1.0 255.255.255.0 summary-only
  redistribute lisp metric 10 route-map LISP_TO_BGP
 exit-address-family
 !

!
route-map LISP_TO_BGP permit 10
 description prefixes_learnt
 set as-path tag
!
wireless fabric
wireless fabric name APVlan92-IPV4 l2-vnid 8189 l3-vnid 4097 ip 10.92.1.1 255.255.255.0
control-plane-name default-control-plane
wireless fabric name wireless-VN-IPV4 l2-vnid 8188 control-plane-name default-control-plane
wireless fabric control-plane default-control-plane ip address 172.16.1.68 key 7 auth-key

wlan kFiab-local-open_profile 17 kFiab-local-open
 radio policy dot11 24ghz
 radio policy dot11 5ghz
 no security wpa
 no security wpa wpa2
 no security wpa wpa2 ciphers aes
 no security wpa akm dot1x
 no shutdown
!
```

```
wireless profile policy kFiab-local-open_profile
 no central dhcp
 no central switching
 description kFiab-local-open_profile
 dhcp-tlv-caching
 exclusionlist timeout 180
 fabric kFiab-local-open_profile   // fabric wireless profile
 http-tlv-caching
 service-policy input platinum-up
 service-policy output platinum
 session-timeout 1800
 no shutdown
!
!
wireless profile fabric kFiab-local-open_profile  // configures wireless profile parameters

 client-l2-vnid 8188
 description kFiab-local-open_profile
!!
```

Verify Fabric in a Box with Embedded Wireless

You can verify the fabric in a box with embedded wireless configuration using the **show** commands. This section provides the sample outputs for the **show** commands on the fabric in a box device in the topology shown Figure 13: LISP VXLAN Topology for Fabric in a Box with Embedded Wireless.

```
fiab# show lisp session
Sessions for VRF default, total: 4, established: 3
Peer                 State      Up/Down        In/Out    Users
172.16.1.68:4342    Up         10:48:14      232/144    10
172.16.1.68:51283   Up         10:48:14      144/232    8
172.16.1.68:60947   Up         10:48:15       48/29     3
fiab#


fiab# show wlan summary

Number of WLANs: 1

ID  Profile Name                  SSID            Status   2.4GHz/5GHz Security  6GHz Security

------------------------------------------------------------------------------------------
17  kFiab-local-open_profile   kFiab-local-open    UP     [open]


fiab# show wireless fabric summary

Fabric Status      : Enabled

Control-plane:
Name                             IP-address        Key                   Status
------------------------------------------------------------------------------------------
default-control-plane            172.16.1.68      bcad25df225e410d        Up


Fabric VNID Mapping:
  Name           L2-VNID    L3-VNID     IP Address    Subnet        Control plane name
------------------------------------------------------------------------------------------

APVlan92-IPV4     8189      4097     10.92.1.1  255.255.255.0     default-control-plane
```

```
wireless-VN-IPV4   8188     0         0.0.0.0                     default-control-plane


fiab#

fiab# show wireless client summary
Number of Clients: 1

MAC Address    AP Name                Type ID   State    Protocol Method     Role
-------------------------------------------------------------------------------------------
4c34.889a.06be AP0CD0.F894.6540       WLAN 17   Run      11ac     None       Local


Number of Excluded Clients: 0

fiab# show wireless client mac-address 4c34.889a.06be details

Client MAC Address : 4c34.889a.06be
Client MAC Type : Universally Administered Address
Client DUID: NA
Client IPv4 Address : 10.51.1.12
Client IPv6 Addresses : fe80::311d:6e13:9d40:9dab
Client Username: N/A
AP MAC Address : 0cd0.f897.f6c0
AP Name: AP0CD0.F894.6540
AP slot : 1
Client State : Associated
Policy Profile : kFiab-local-open_profile
Flex Profile : default-flex-profile
Wireless LAN Id: 17
WLAN Profile Name: kFiab-local-open_profile
Wireless LAN Network Name (SSID): kFiab-local-open
BSSID : 0cd0.f897.f6ce
Connected For : 41 seconds
Protocol : 802.11ac
Channel : 140
Client IIF-ID : 0xa0000001
Association Id : 1
Authentication Algorithm : Open System
Idle state timeout : N/A
Session Timeout : 1800 sec (Remaining time: 1764 sec)
Session Warning Time : Timer not running
Input Policy Name  : None
Fabric status : Enabled    <--- displays status of the fabric and other details
  RLOC    : 172.16.1.68
  VNID    : 8190
  SGT     : 0
  Control plane name  : default-control-plane

<snip output>
…..
…..
<snip output>

fiab#
```

PART III

Multicast in LISP VXLAN Fabric

CHAPTER 10

Configuring Multicast in LISP VXLAN Fabric

Multicast traffic forwarding is used to simultaneously distribute copies of data to multiple network destinations. In a LISP VXLAN fabric, multicast traffic flow can be handled in the overlay or the underlay, depending on whether the underlay network supports multicast replication or not. This chapter describes how to configure overlay multicast in a LISP VXLAN Fabric.

LISP VXLAN Fabric Multicast Overview

Note This document assumes that the reader is familiar with the fundamentals of Multicast technology. To understand the basics of Multicast technology, refer IP Multicast Technology Overview.

LISP VXLAN Fabric supports the following:

- Layer 2 overlay Broadcast, Unknown Unicast, and Multicast (BUM) traffic to be transported over IP multicast in the underlay network
- Layer 3 overlay multicast

Layer 2 Overlay Broadcast, Unknown Unicast, and Multicast

Multidestination Layer 2 traffic in a network is typically referred to as broadcast, unknown unicast, and multicast (BUM) traffic. In a LISP VXLAN Fabric, the underlay network forwards the BUM traffic to all the endpoints connected to a common Layer 2 broadcast domain in the VXLAN overlay. The BUM functionality is achieved using the Any Source Multicast (ASM) model in the underlay network. The rendezvous points (RPs) are configured on the border nodes. The RLOC devices, which are the source and receivers, join the shared multicast group that is attached to the RPs. We recommend a dual border topology with the RPs configured on both the border nodes for redundancy.

Note Only IPv4 traffic is supported in the underlay.

Layer 3 Overlay Multicast

LISP VXLAN Fabric supports both PIM Any Source Multicast (ASM) and PIM Source Specific Multicast (SSM) in the overlay. Layer 3 overlay multicast supports only IPv4 multicast traffic.

The multicast source can either be outside the fabric site or can be in the fabric overlay, connected to the fabric edge node. Multicast receivers can be located outside the fabric site or be directly connected to the fabric edge nodes.

Multicast forwarding in the Layer 3 overlay uses two methods to distribute the traffic through the underlay: Headend Replication and Underlay Multicast. You can configure either Headend Replication or Underlay Multicast in a virtual network. Both cannot be configured together.

Note Bidirectional PIM (Bidir-PIM) is not supported in the overlay and the underlay network.

Any Source Multicast

Any Source Multicast (ASM) is a multicast distribution mode that requires the use of rendezvous points (RPs) to act as a shared root between sources and receivers of multicast data. You can configure a single RP or multiple RPs in the network.

To configure ASM mode in the Layer 3 overlay, you configure the RP selection method, where you indicate the distribution mode and assigns the range of multicast groups.

External RP

External devices can be designated as the RP for the multicast tree in a fabric. To function as an external RP, a device must be a router with PIM enabled. This device is located external to the fabric and is connected to the fabric through one or more border nodes. The External RP address must be reachable in the VRF routing table on the border nodes.

Note This release of LISP VXLAN Fabric supports only external RP for overlay multicast traffic.

Source Specific Multicast

Source Specific Multicast (SSM) creates an optimal path between the multicast source and receiver without the need for a rendezvous point.

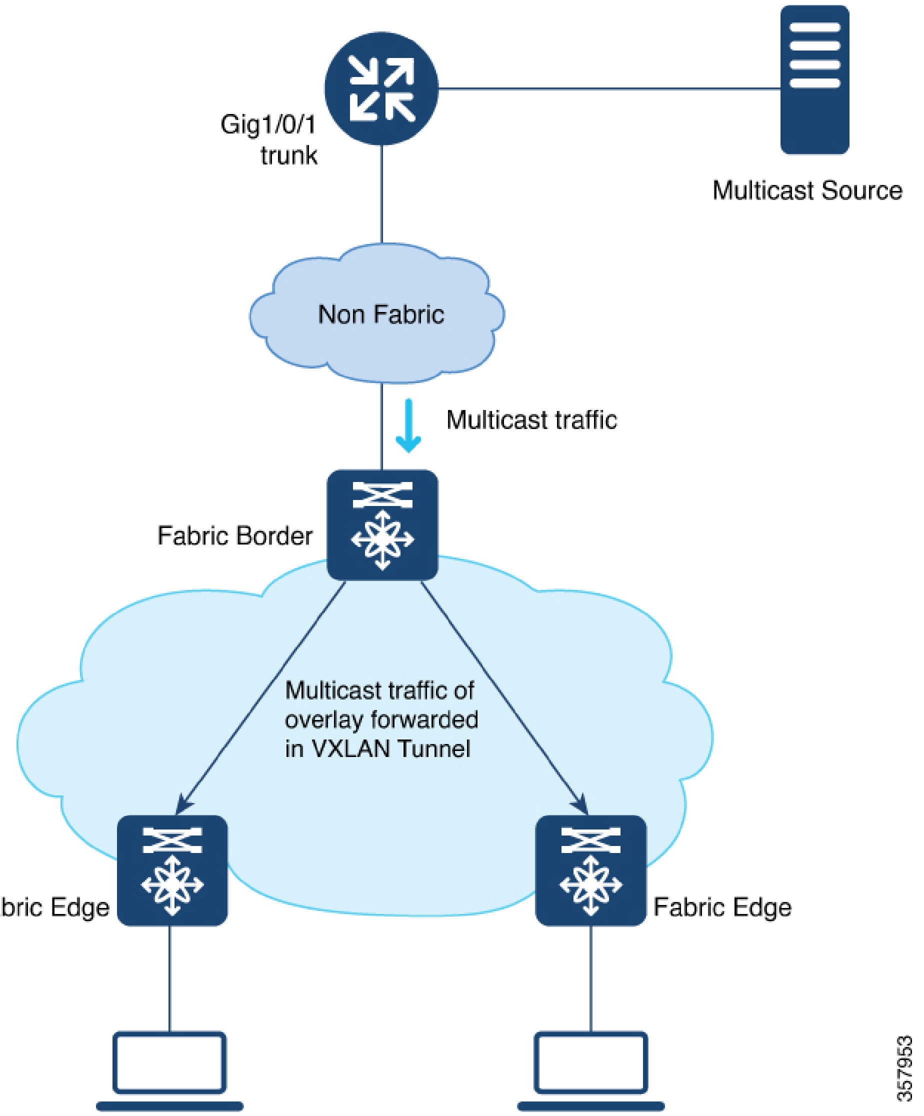
You can configure the SSM multicast range that can be supported by the fabric.

Headend Replication

Headend replication is performed by the multicast first-hop router. The first fabric node (FHR) that receives the multicast traffic replicates multiple copies of the VXLAN-encapsulated data packet and unicasts a copy to each of the remote fabric edge nodes where the multicast receivers are located.

The advantage of headend replication is that it does not require multicast in the underlay network to transport the overlay multicast packets. However, it can create a high overhead on the FHRs and result in high bandwidth and CPU utilization.

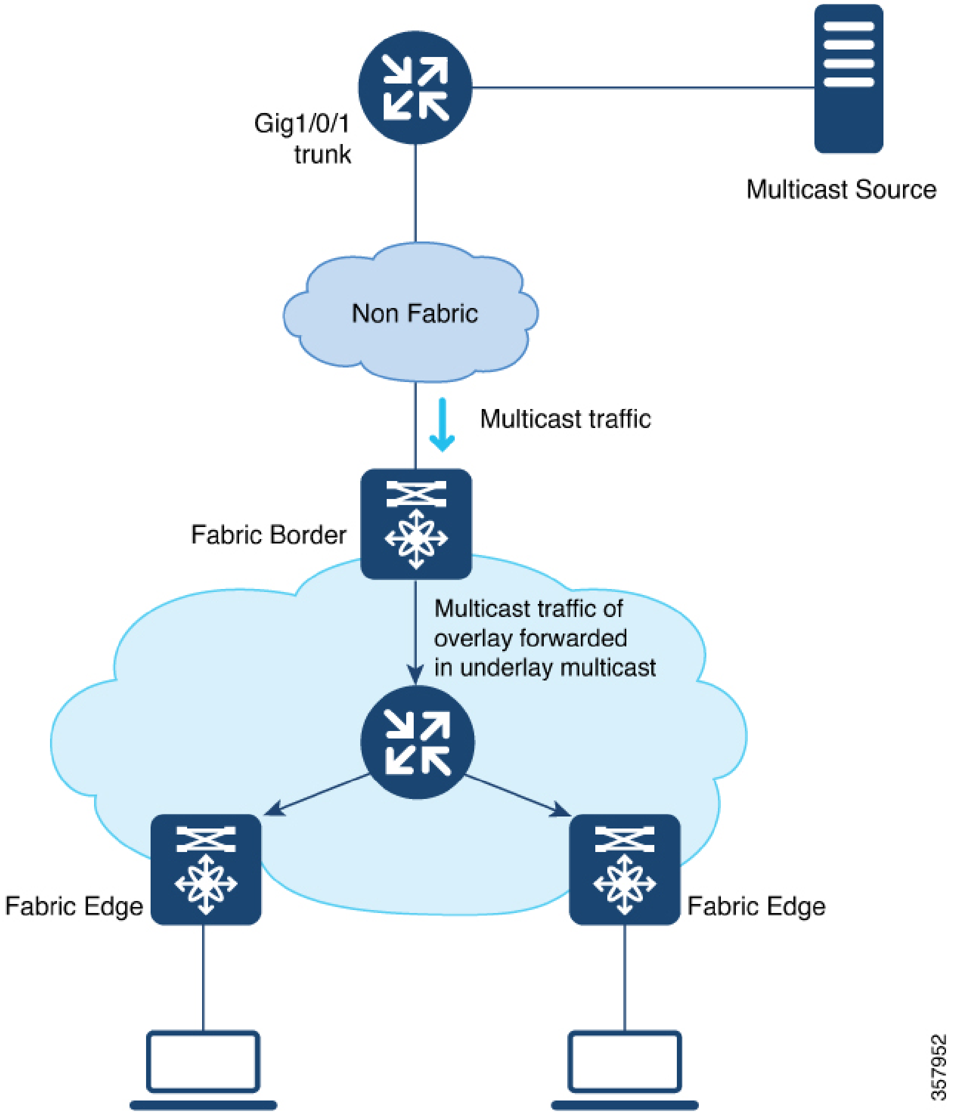
Figure 14: Headend Replication in a LISP VXLAN



Underlay Multicast

Underlay multicast works by performing multicast-in-multicast encapsulation. The multicast packets in the overlay network are transported as multicast in the underlay. The load of packet replication is shared across all the devices in the underlay network. To support underlay multicast, the FHRs, Last Hop Routers (LHRs), and all network infrastructure between them must be enabled for multicast. PIM SSM is used in the underlay for multicast transport.

Figure 15: Underlay Multicast Forwarding in LISP VXLAN



Layer 3 Overlay Multicast Support in LISP VXLAN Fabric

The following multicast methods are supported in this release of LISP VXLAN Fabric:

Layer 3 Overlay Multicast

- SSM with Underlay Multicast
- SSM with Headend Replication
- ASM with Underlay Multicast, External Rendezvous Point
- ASM with Headend Replication, External Rendezvous Point

How to Configure Broadcast, Unknown Unicast, Multicast

Layer 2 multicast supports only IPv4 multicast traffic in the underlay. Configure ASM mode in the underlay, with the RPs located on the border nodes. If the network has more than one border, configure the RPs on two border nodes with Multicast Source Discovery Protocol (MSDP) to provide redundancy in the network. Configure the fabric edge nodes or the RLOC devices as the multicast source and receivers.

Configure Layer 2 Overlay Broadcast, Unknown Unicast, and Multicast

Do the following configurations on the border node and edge node devices to configure Layer 2 overlay Broadcast, Unknown Unicast, and Multicast (BUM) traffic in the underlay network.

Before you begin

- Ensure that multicast is enabled in the underlay.
- Configure the border node device as the underlay rendezvous point.
- Ensure that Multicast Source Discovery Protocol (MSDP) is enabled between the border nodes in the underlay network.
- Ensure that PIM sparse-mode is enabled on Loopback 0 and all point-to-point interfaces.

Procedure

Step 1 Configure Multicast Source Discovery Protocol (MSDP) on the border nodes in the underlay.

If your fabric network has dual borders, configure MSDP on each of the borders to exchange multicast source information. MSDP also provides redundancy and load sharing between the two borders.

a) **ip msdp peer** *peer-address* **connect-source** *type* [*interface-path-id*]

Example:

```
Device(config)# ip msdp peer 172.16.1.67 connect-source Loopback0
```

Configures the MSDP peer and specifies the Loopback interface of the device as the source address for the MSDP connection. *peer-address* is the loopback0 address of the other border node.

b) **ip msdp cache-sa-state**

Example:

```
Device(config)# ip msdp cache-sa-state
```

Configures the Source-Active (SA) cache to store the SA messages that are received from the peer.

The SA cache holds the information for all sources learned through SA messages.

c) **ip msdp originator-id** *type* [*interface-path-id*]

Example:

```
Device(config)# ip msdp originator-id Loopback0
```

Allows an MSDP speaker that originates an SA message to use the loopback0 address of the interface as the RP address in the SA message.

Step 2 Configure the Loopback interface for the anycast RP on the border nodes and enable PIM sparse mode on it.

Example:

```
Device(config)# interface Loopback100
Device(config-if)# ip address 172.16.1.100 255.255.255.255
Device(config-if)# ip pim sparse-mode
```

Step 3 **ip multicast-routing**

Example:

```
Device(config)# ip multicast-routing
```

Enables IP multicast routing.

Step 4 **ip pim register-source** *interface*

Example:

```
Device(config)# ip pim register-source Loopback0
```

Configures the loopback address of the device as the source address of a PIM Register message.

Step 5 **ip pim rp-address** *address*

Example:

```
Device(config)# ip pim rp-address 172.16.1.100
```

Configures a static rendezvous point (RP) address.

Step 6 **ip pim ssm default**

Example:

```
Device(config)# ip pim ssm default
```

Defines a default range of SSM multicast address.

Step 7 Do the following configurations on the fabric edge node:

a) **router lisp**

Example:

```
Device(config)# router lisp
```

Enters LISP configuration mode.

b) **instance-id** *id*

Example:

```
Device(config-router-lisp)# instance-id 8188
```

Specifies the instance ID.

c) **service ethernet**

Example:

```
Device(config-router-lisp-inst)# service ethernet
```

Enables Layer 2 network services.

d) **eid-table vlan** *vlan-id*

Example:

```
Device(config-router-lisp-inst-serv-ethernet)# eid-table vlan 50
```

Associates the VLAN with this Layer 2 service instance.

e) **broadcast-underlay** *multicast-ip*

Example:

```
Device(config-router-lisp-inst-serv-ethernet)# broadcast-underlay 239.0.17.1
```

Enables the broadcast functionality on the fabric edge node.

f) **flood unknown-unicast**

Example:

```
Device(config-router-lisp-inst-serv-ethernet)# flood unknown-unicast
```

Floods the unknown broadcast, unicast packets in the Layer 2 domain.

g) **flood arp-nd**

Example:

```
Device(config-router-lisp-inst-serv-ethernet)# flood arp-nd
```

Enables Address Resolution Protocol (ARP) flooding in the Layer 2 domain.

h) **exit-service-ethernet**

Example:

```
Device(config-router-lisp-inst-serv-ethernet)# exit-service-ethernet
```

Exits service Ethernet configuration mode, and enters LISP instance configuration mode.

i) **exit-instance-id**

Example:

```
Device(config-router-lisp-inst)# exit-instance-id
```

Exits instance configuration mode, and enters LISP configuration mode.

j) **end**

Example:

```
Device(config-router-lisp)# end
```

Returns to privileged EXEC mode.

Repeat the steps to enable broadcast, unknown unicast, and multicast functionality for all the Layer 2 instances that were created while configuring the fabric edge node

Refer the How to Configure a Fabric Edge Node chapter to see the Layer 2 instances that are created.

Configuration Example for Layer 2 Overlay Broadcast, Unknown Unicast, Multicast

Here is a sample configuration for Layer 2 overlay BUM traffic. The fabric network has two colocated border and control plane nodes. The underlay anycast RP is configured on the dual border nodes.

Note that the table shows only the snippet of the configurations that are required to enable Layer 2 overlay BUM.

Table 4: Fabric Edge and Border Node Configurations for Layer 2 BUM

<table>
<tr><th>Border Node Configurations</th><th>Fabric Edge Node Configurations</th></tr>
<tr><td>

Border Node 1

```
interface Loopback0
 ip address 172.16.1.66 255.255.255.255
 ip pim sparse-mode
!
interface Loopback100
 ip address 172.16.1.100 255.255.255.255
 ip pim sparse-mode

ip multicast-routing
ip pim rp-address 172.16.1.100
ip pim register-source Loopback0
ip pim ssm default
!!
ip msdp peer 172.16.1.67 connect-source
Loopback0
ip msdp cache-sa-state
ip msdp originator-id Loopback0
```

Border Node 2

```
interface Loopback0
 ip address 172.16.1.67 255.255.255.255
 ip pim sparse-mode
!
interface Loopback100
 ip address 172.16.1.100 255.255.255.255
 ip pim sparse-mode
!
ip multicast-routing
ip pim rp-address 172.16.1.100
ip pim register-source Loopback0
ip pim ssm default
!
ip msdp peer 172.16.1.66 connect-source
Loopback0
ip msdp cache-sa-state
ip msdp originator-id Loopback0
```

</td><td>

```
instance-id 8197
  service ethernet
   eid-table vlan 50
   broadcast-underlay 239.0.17.1
   flood arp-nd
   flood unknown-unicast
   exit-service-ethernet
  !

ip multicast-routing
ip pim rp-address 172.16.1.100
ip pim register-source Loopback0
ip pim ssm default
```

</td></tr>
</table>

How to Configure Layer 3 Overlay Multicast in a LISP VXLAN Fabric

This section uses a single procedure to describe the configuration steps for the Headend Replication (ASM and SSM) and Underlay Multicast (ASM and SSM) forwarding methods. Some steps are applicable only to a particular method, either Headend Replication or Underlay Multicast. Such steps are called out clearly at the beginning of the respective step.

The configuration procedure is followed by configuration examples.

Configure Layer 3 Overlay Multicast in a LISP VXLAN Fabric

This task describes how to configure multicast in the overlay network, and assumes that multicast is already configured in the underlay network.

Note Unless otherwise noted, perform the following steps on both the border node and fabric edge node.

Before you begin

- Ensure that multicast is enabled in the underlay network.
- Ensure that the control plane node, border node, and edge nodes of the overlay are configured and virtual routing and forwarding (VRF) routing table instances are configured for unicast communication. Refer earlier chapters of this document for information on configuring the fabric.

Note Ensure that you configure PIM Sparse mode on all the core-facing fabric devices.

Procedure

Step 1 Enable multicast routing for the overlay network, using the **ip multicast-routing vrf** *vrf-name* command in the global configuration mode.

Example:

```
Device(config)# ip multicast-routing vrf VN3
```

Enables IP multicast routing on the specified VRF.

Step 2 Configure a Loopback interface for multicast segment.

a) **interface Loopback** *multicast-segment-interface*

Example:

```
Device(config)# interface Loopback4099
```

Configures the loopback interface and enters the interface configuration mode.

b) **vrf forwarding** *vrf-name*

Example:

```
Device(config-if)# vrf forwarding VN3
```

Enables VRF forwarding on the interface.

c) **ip address** *address mask*

Example:

```
Device(config-if)# ip address 10.22.1.1 255.255.255.255
```

Assigns an IP address to the interface.

d) **ip pim sparse-mode**

Example:

```
Device(config-if)#ip pim sparse-mode
```

Enables Protocol Independent Multicast (PIM) on the interface for sparse-mode operation.

e) **exit**

Example:

```
Device(config-if)# exit
Device(Config)#
```

Returns to the global configuration mode.

Step 3 Enable PIM on the LISP interface.

a) **interface** *interface-name*

Example:

```
Device(config)# interface LISP0.4099
```

Configures the LISP interface and enters the LISP interface configuration mode.

b) **Perform this step only for Underlay Multicast**: **ip pim lisp transport multicast**

Example:

```
Device(config-if)# ip pim lisp transport multicast
```

Enables multicast on the LISP interface.

c) **Perform this step only for Headend Replication**: **ip pim sparse-mode**

Example:

```
Device(config-if)# ip pim sparse-mode
```

Enables Protocol Independent Multicast (PIM) on the interface for sparse-mode operation.

Execute this step only if you are configuring Headend Replication.

d) **Perform this step only for Headend Replication with SSM**: **ip pim lisp core-group-range** *start-SSM-address range-size*

Example:

```
Device(config-if)# ip pim lisp core-group-range 232.0.0.1 1000
```

Configures the group of IP addresses for SSM on a LISP interface, to transport multicast traffic.

e) **exit**

Example:

```
Device(config-if)# exit
Device(config)#
```

Returns to the global configuration mode.

Step 4 On the border node, if Layer 3 handoff is configured, configure PIM on the Layer 3 overlay.

a) **interface** *interface-number*

Example:

```
Device(config)# interface Vlan222
```

Enters the Layer 3 overlay SVI configuration mode.

b) **ip pim sparse-mode**

Example:

```
Device(config-if)#ip pim sparse-mode
```

Enables Protocol Independent Multicast (PIM) on the SVI for sparse-mode operation.

c) **exit**

Example:

```
Device(config-if)# exit
Device(config)#
```

Returns to the global configuration mode.

Step 5 On the edge node, enable PIM and IGMP for the user-defined VRF.

a) **interface** *interface-number*

Example:

```
Device(config)# interface Vlan50
```

Enters the interface configuration mode for the user-defined VRF.

b) **vrf forwarding** *vrf-name*

Example:

```
Device(config-if)# vrf forwarding VN3
```

Enables VRF forwarding on the interface.

c) **ip pim passive**

Example:

```
Device(config-if)# ip pim passive
```

Configures a PIM passive interface.

A PIM passive interface does not send or receive any PIM control messages. However, a PIM passive interface acts as designated router (DR) and designated forwarder (DF)-election winner, and it can accept and forward multicast data.

d) **ip igmp version** {**1** | **2** | **3**}

Example:

```
Device(config-if)# ip igmp version 3
```

Configures the version of the Internet Group Management Protocol (IGMP) for the device to use.

e) **exit**

Example:

```
Device(config-if)# exit
Device(config)#
```

Returns to the global configuration mode.

Step 6 Map the multicast EID database to the instance ID of the VRF.

a) **router lisp**

Example:

```
Device(config)# router lisp
```

Enters LISP configuration mode.

b) **instance-id** *id*

Example:

```
Device(config-router-lisp)# instance-id 4099
```

Specifies the instance ID of the VRF.

c) **service ipv4**

Example:

```
Device(config-router-lisp-inst)# service ipv4
```

Enables Layer 3 network services for this instance-id.

d) **database-mapping** *eid-prefix/prefix-length* **locator-set** *RLOC_name*

Example:

```
Device(config-router-lisp-inst-serv-ipv4)# database-mapping 10.22.1.2/32 locator-set
eid_LOCATOR
```

Configures EID-to-RLOC relationship in the LISP database.

e) **exit-service-ipv4**

Example:

```
Device(config-router-lisp-inst-serv-ipv4)# exit-service-ipv4
```

Exits service IPv4 configuration mode, and enters LISP instance configuration mode

f) **exit-instance-id**

Example:

```
Device(config-router-lisp-inst)# exit-instance-id
```

Exits instance configuration mode, and enters LISP configuration mode.

Step 7 (Optional) On the border node, advertise the loopback interface of the multicast segment to the external domain, using the BGP routing process.

a) **router bgp** *autonomous-system-number*

Example:

```
Device(config)# router bgp 700
```

Configures a BGP routing process, and enters router configuration mode for the specified routing process.

b) **address-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*]

Example:

```
Device(config-router)# address-family ipv4 vrf VN3
```

Specifies the VRF instance with which the subsequent address family configuration commands are associated.

c) **network** *network-number* [**mask** *network-mask*] [**route-map** *route-map-name*]

Example:

```
Device(config-router-af)# network 10.22.1.1 mask 255.255.255.255
```

Specifies the network to be advertised by BGP and adds it to the BGP routing table.

d) **aggregate-address** *address mask* [**summary-only**]

Example:

```
Device(config-router-af)#  aggregate-address 10.22.1.0 255.255.255.0 summary-only
```

Generates an aggregate entry in the BGP database.

Use the optional **summary-only** keyword to create the aggregate route (for example, 10.*.*.*) and also suppresses advertisements of more-specific routes to all neighbors.

e) **exit-address-family**

Example:

```
Device(config-router-af)# exit-address-family
```

Exits the address family configuration mode.

f) **exit**

Example:

```
Device(config-router)# exit
Device(config)#
```

Returns to the global configuration mode.

Step 8 **Perform this step only for SSM**: Define the range of SSM multicast address.

a) **ip pim vrf** *vrf-name* **ssm range** *access-list*

Example:

```
Device(config)# ip pim vrf VN3 ssm range SSM_RANGE_VN3
```

Configures the SSM service for the IP address range defined by the access list.

b) **ip access-list standard** *access-list-name*

Example:

```
Device(config)# ip access-list standard SSM_RANGE_VN3
Device(config)# 10 permit 232.0.0.0 0.255.255.255
Device(config)# exit
```

Define the the access list for the SSM multicast IP address.

Step 9 **Perform this step only for ASM**: Create a loopback for PIM and configure a static rendezvous point.

a) **ip pim vrf** *vrf-name* **register-source** *interface-type interface number*

Example:

```
Device(config)# ip pim vrf VN3 register-source Loopback4099
```

Configures the loopback address of the VRF as the source address of a PIM Register message.

b) **ip pim vrf** *vrf-name* **rp-address** *rp-address* [*access-list*]

Example:

```
Device(config)# ip pim vrf VN3 rp-address 172.16.3.1 ASM_ACL_IPV4_VN3_172.16.3.1
```

Configures the IP address of the rendezvous point to be used for the static group-to-RP mapping and specifies the access list that defines the multicast groups to be statically mapped to the rendezvous point.

Step 10 Enable PIM sparse mode on all the core-facing interfaces of the underlay network.

Example:

```
Device(config)# interface Gigabitethernet1/0/1

Device(config)# ip pim sparse
```

Repeat this step for all the core-facing interfaces of the fabric devices.

Configuration Example for Underlay Multicast with SSM

This is a sample configuration for Underlay Multicast with SSM. In this sample, 10.22.1.0/24 is the multicast subnet. Multicast source is located outside the fabric. The multicast listeners are within the fabric overlay. This configuration assumes that multicast is already configured in the underlay and the LISP VXLAN fabric edge nodes, border node, and control plane node are also up and running.

Table 5: Configurations on the Fabric Edge and Border Nodes

Border Node Configurations	Edge Node Configurations
ip multicast-routing vrf VN3 ! interface Loopback4099 vrf forwarding VN3 ip address 10.22.1.1 255.255.255.255 ip pim sparse-mode ! interface LISP0.4099 ip pim lisp transport multicast ip pim lisp core-group-range 232.0.0.1 1000 ! interface Vlan222 ip pim sparse-mode router lisp instance-id 4099 service ipv4 database-mapping 10.22.1.1/32 locator-set eid_LOCATOR exit-service-ipv4 ! exit-instance-id ! router bgp 700 ! address-family ipv4 vrf VN3 network 10.22.1.1 mask 255.255.255.255 aggregate-address 10.22.1.0 255.255.255.0 summary-only exit-address-family ! ! ip pim vrf VN3 ssm range SSM_RANGE_VN3 ! ip access-list standard SSM_RANGE_VN3 10 permit 232.0.0.0 0.255.255.255 ! interface Gig/Tengig/Hunderxxx ip pim sparse	ip multicast-routing vrf VN3 ! interface Loopback4099 vrf forwarding VN3 ip address 10.22.1.2 255.255.255.255 ip pim sparse-mode ! interface LISP0.4099 ip pim lisp transport multicast ip pim lisp core-group-range 232.0.0.1 1000 ! interface Vlan50 vrf forwarding VN3 ip pim passive ip igmp version 3 ip igmp explicit-tracking router lisp instance-id 4099 service ipv4 database-mapping 10.22.1.2/32 locator-set eid_LOCATOR exit-service-ipv4 ! exit-instance-id ! ip pim vrf VN3 ssm range SSM_RANGE_VN3 ! ip access-list standard SSM_RANGE_VN3 10 permit 232.0.0.0 0.255.255.255 ! interface Gig/Tengig/Hunderxxx ip pim sparse

Configuration Example for Underlay Multicast with ASM, External RP

This is a sample configuration for Underlay Multicast with ASM. In this sample, 10.22.1.0/24 is the multicast subnet. Multicast source is located outside the fabric. The multicast listeners are within the fabric overlay. The rendezvous point (RP) is located external to the fabric. This configuration assumes that multicast is already configured in the underlay and the LISP VXLAN fabric edge nodes, border node, and control plane node are also up and running.

Table 6: Configurations on the Fabric Edge and Border Nodes

Border Node Configurations	Edge Node Configurations
ip multicast-routing vrf VN3 ! interface Loopback4099 vrf forwarding VN3 ip address 10.22.1.1 255.255.255.255 ip pim sparse-mode ! interface LISP0.4099 ip pim lisp transport multicast ip pim lisp core-group-range 232.0.0.1 1000 ! interface Vlan222 ip pim sparse-mode router lisp instance-id 4099 service ipv4 database-mapping 10.22.1.1/32 locator-set eid_LOCATOR exit-service-ipv4 ! exit-instance-id ! router bgp 700 ! address-family ipv4 vrf VN3 network 10.22.1.1 mask 255.255.255.255 aggregate-address 10.22.1.0 255.255.255.0 summary-only exit-address-family ! ! ip pim vrf VN3 rp-address 172.16.3.1 ASM_ACL_IPV4_VN3_172.16.3.1 ip pim vrf VN3 register-source Loopback4099 ! ip access-list standard ASM_ACL_IPV4_VN3_172.16.3.1 10 permit 229.1.1.0 0.0.0.255 ! interface Gig/Tengig/Hunderxxx ip pim sparse	ip multicast-routing vrf VN3 ! interface Loopback4099 vrf forwarding VN3 ip address 10.22.1.2 255.255.255.255 ip pim sparse-mode ! interface LISP0.4099 ip pim lisp transport multicast ip pim lisp core-group-range 232.0.0.1 1000 ! interface Vlan50 vrf forwarding VN3 ip pim passive ip igmp version 3 ip igmp explicit-tracking ipv6 mld explicit-tracking ipv6 pim passive router lisp instance-id 4099 service ipv4 database-mapping 10.22.1.2/32 locator-set eid_LOCATOR exit-service-ipv4 ! exit-instance-id ! ip pim vrf VN3 rp-address 172.16.3.1 ASM_ACL_IPV4_VN3_172.16.3.1 ip pim vrf VN3 register-source Loopback4099 ! ip access-list standard ASM_ACL_IPV4_VN3_172.16.3.1 10 permit 229.1.1.0 0.0.0.255 ! interface Gig/Tengig/Hunderxxx ip pim sparse

Configuration Example for Headend Replication with SSM

This is a sample configuration for Headend Replication with SSM. In this sample, 10.22.1.0/24 is the multicast subnet. Multicast source is located outside the fabric. The multicast listeners are within the fabric overlay.

This configuration assumes that multicast is already configured in the underlay and the LISP VXLAN fabric edge nodes, border node, and control plane node are also up and running.

Table 7: Configurations on the Fabric Edge and Border Nodes

Border Node Configurations	Edge Node Configurations
ip multicast-routing vrf VN3 ! ! interface Loopback4099 vrf forwarding VN3 ip address 10.22.1.1 255.255.255.255 ip pim sparse-mode! interface LISP0.4099 ip pim sparse-mode ! interface Vlan222 ip pim sparse-mode router lisp instance-id 4099 service ipv4 database-mapping 10.22.1.1/32 locator-set eid_LOCATOR exit-service-ipv4 ! exit-instance-id ! router bgp 700 ! address-family ipv4 vrf VN3 network 10.22.1.1 mask 255.255.255.255 aggregate-address 10.22.1.0 255.255.255.0 summary-only exit-address-family ! ! ip pim vrf VN3 ssm range SSM_RANGE_VN3 ! ip access-list standard SSM_RANGE_VN3 10 permit 232.0.0.0 0.255.255.255 !	 ip multicast-routing vrf VN3 ! interface Loopback4099 vrf forwarding VN3 ip address 10.22.1.2 255.255.255.255 ip pim sparse-mode ! interface LISP0.4099 ip pim sparse-mode ! interface Vlan50 vrf forwarding VN3 ip pim passive ip igmp version 3 ip igmp explicit-tracking router lisp instance-id 4099 service ipv4 database-mapping 10.22.1.2/32 locator-set eid_LOCATOR exit-service-ipv4 ! ip pim vrf VN3 ssm range SSM_RANGE_VN3 ! ip access-list standard SSM_RANGE_VN3 10 permit 232.0.0.0 0.255.255.255 ! ipv6 pim vrf VN3 register-source Loopback4099

Configuration Example for Headend Replication with ASM, External RP

This is a sample configuration for Headend Replication with ASM. In this sample, 10.22.1.0/24 is the multicast subnets. Multicast source is located outside the fabric. The multicast listeners are within the fabric overlay. The rendezvous point (RP) is located external to the fabric. This configuration assumes that multicast is already configured in the underlay and the LISP VXLAN fabric edge nodes, border node, and control plane node are also up and running.

Table 8: Configurations on the Fabric Edge and Border Nodes

Border Node Configurations	Edge Node Configurations
ip multicast-routing vrf VN3 ! interface Loopback4099 vrf forwarding VN3 ip address 10.22.1.1 255.255.255.255 ip pim sparse-mode ! interface LISP0.4099 ip pim sparse-mode ! interface Vlan222 ip pim sparse-mode router lisp instance-id 4099 service ipv4 database-mapping 10.22.1.1/32 locator-set eid_LOCATOR exit-service-ipv4 ! exit-instance-id ! router bgp 700 ! address-family ipv4 vrf VN3 network 10.22.1.1 mask 255.255.255.255 aggregate-address 10.22.1.0 255.255.255.0 summary-only exit-address-family ! ! ip pim vrf VN3 rp-address 172.16.3.1 ASM_ACL_IPV4_VN3_172.16.3.1 ip pim vrf VN3 register-source Loopback4099 ! ip access-list standard ASM_ACL_IPV4_VN3_172.16.3.1 10 permit 229.1.1.0 0.0.0.255 !	ip multicast-routing vrf VN3 ! ! interface Loopback4099 vrf forwarding VN3 ip address 10.22.1.2 255.255.255.255 ip pim sparse-mode! interface LISP0.4099 ip pim sparse-mode ! interface Vlan50 vrf forwarding VN3 ip pim passive ip igmp version 3 ip igmp explicit-tracking router lisp instance-id 4099 service ipv4 database-mapping 10.22.1.2/32 locator- exit-service-ipv4 ! ! ip pim vrf VN3 rp-address 172.16.3.1 ASM_ACL_IPV4_VN3_172 ip pim vrf VN3 register-source Loopback4 ! ip access-list standard ASM_ACL_IPV4_VN3 10 permit 229.1.1.0 0.0.0.255 !

Verify the Multicast Configuration in LISP VXLAN Fabric

This section provides sample outputs for the **show** commands to verify the multicast configuration on the fabric edge and border nodes.

Verify Layer 2 BUM

```
FabricEdge# show ip mfib 239.0.17.1
Entry Flags:    C - Directly Connected, S - Signal, IA - Inherit A flag,
                ET - Data Rate Exceeds Threshold, K - Keepalive
                DDE - Data Driven Event, HW - Hardware Installed
                ME - MoFRR ECMP entry, MNE - MoFRR Non-ECMP entry, MP - MFIB
                MoFRR Primary, RP - MRIB MoFRR Primary, P - MoFRR Primary
                MS  - MoFRR  Entry in Sync, MC - MoFRR entry in MoFRR Client,
                e   - Encap helper tunnel flag.
I/O Item Flags: IC - Internal Copy, NP - Not platform switched,
                NS - Negate Signalling, SP - Signal Present,
                A - Accept, F - Forward, RA - MRIB Accept, RF - MRIB Forward,
                MA - MFIB Accept, A2 - Accept backup,
                RA2 - MRIB Accept backup, MA2 - MFIB Accept backup

Forwarding Counts: Pkt Count/Pkts per second/Avg Pkt Size/Kbits per second
Other counts:      Total/RPF failed/Other drops
I/O Item Counts:   HW Pkt Count/FS Pkt Count/PS Pkt Count   Egress Rate in pps
Default
(*,239.0.17.1) Flags: C HW
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   HW Forwarding:   11/0/172/0, Other: 0/0/0
   TwentyFiveGigE1/0/15 Flags: A NS
   L2LISP0.8197, L2LISP Decap Flags: F NS
     Pkts: 0/0/0    Rate: 0 pps
   L2LISP0.8194, L2LISP Decap Flags: F NS
     Pkts: 0/0/0    Rate: 0 pps
(172.16.1.69,239.0.17.1) Flags: HW
   SW Forwarding: 2/0/154/0, Other: 0/0/0
   HW Forwarding:   4710/0/172/0, Other: 0/0/0
   TwentyFiveGigE1/0/15 Flags: A NS
   L2LISP0.8197, L2LISP Decap Flags: F NS
     Pkts: 0/0/2    Rate: 0 pps
   L2LISP0.8194, L2LISP Decap Flags: F NS
     Pkts: 0/0/2    Rate: 0 pps
(172.16.1.68,239.0.17.1) Flags: HW
   SW Forwarding: 2/0/154/0, Other: 762/762/0
   HW Forwarding:   4476/0/145/0, Other: 0/0/0
   Null0 Flags: A
FabricEdge#


FabricEdge# show lisp instance-id 8197 ethernet map-cache
LISP MAC Mapping Cache for LISP 0 EID-table Vlan 50 (IID 8197), 1 entries

000c.29c6.6069/48, uptime: 20:50:25, expires: 03:09:34, via map-reply, complete
  Locator    Uptime    State  Pri/Wgt     Encap-IID
  172.16.1.68  20:50:25  up      10/10        -



FabricEdge# show lisp instance-id 8197 ethernet database
LISP ETR MAC Mapping Database for LISP 0 EID-table Vlan 50 (IID 8197), LSBs: 0x1
Entries total 3, no-route 0, inactive 0, do-not-register 1

0000.0c9f.f18e/48, dynamic-eid Auto-L2-group-8197, do not register, inherited from default
 locator-set rloc_set2
  Uptime: 5d20h, Last-change: 5d20h
  Domain-ID: local
  Service-Insertion: N/A
  Locator    Pri/Wgt  Source     State
  172.16.1.69   10/10   cfg-intf   site-self, reachable
000c.2966.f195/48, dynamic-eid Auto-L2-group-8197, inherited from default locator-set
rloc_set2
  Uptime: 3d01h, Last-change: 3d01h
  Domain-ID: local
  Service-Insertion: N/A
  Locator    Pri/Wgt  Source     State
  172.16.1.69   10/10   cfg-intf   site-self, reachable
000c.2979.439d/48, dynamic-eid Auto-L2-group-8197, inherited from default locator-set
rloc_set2
  Uptime: 3d01h, Last-change: 3d01h
  Domain-ID: local
  Service-Insertion: N/A
  Locator    Pri/Wgt  Source     State
  172.16.1.69   10/10   cfg-intf   site-self, reachable
```

```
FabricEdge# show mac address-table vlan 50
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
  50    0000.0c9f.f18e    STATIC      Vl50
  50    000c.2966.f195    DYNAMIC     Gi1/0/31
  50    000c.2979.439d    DYNAMIC     Gi1/0/30
  50    6c03.09cb.7a68    STATIC      Vl50
  50    000c.29c6.6069    CP_LEARN    L2LI0
Total Mac Addresses for this criterion: 4
Total Mac Addresses installed by LISP: REMOTE: 1
```

```
Border# show ip msdp summary
MSDP Peer Status Summary
Peer Address     AS    State    Uptime/  Reset SA    Peer Name
                                Downtime Count Count
172.16.1.66      6502  Up       5d02h    0     0     ?
```

```
Check the multicast groups on RP
```

```
Border# show ip pim rp
Group: 239.0.17.1, RP: 172.16.1.100
```

View the IP Multicast Routing Table for the VRF:

```
FabricEdge# show ip mroute vrf VN3 summary
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group,
       G - Received BGP C-Mroute, g - Sent BGP C-Mroute,
       N - Received BGP Shared-Tree Prune, n - BGP C-Mroute suppressed,
       Q - Received BGP S-A Route, q - Sent BGP S-A Route,
       V - RD & Vector, v - Vector, p - PIM Joins on route,
       x - VxLAN group, c - PFP-SA cache created entry,
       * - determined by Assert, # - iif-starg configured on rpf intf,
       e - encap-helper tunnel flag, l - LISP decap ref count contributor
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
                          t - LISP transit group
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 229.1.1.1), 17:46:37/stopped, RP 10.22.1.1, OIF count: 1, flags: SJC
  (12.12.12.124, 229.1.1.1), 00:38:27/00:01:52, OIF count: 1, flags: JT

(*, 224.0.1.40), 17:47:10/00:02:51, RP 10.22.1.1, OIF count: 1, flags: SJ
```

View the multicast interfaces for the VRF:

```
Border# show ip pim vrf VN3 interface
```

```
Address          Interface                Ver/   Nbr    Query  DR         DR
                                          Mode   Count  Intvl  Prior
10.22.1.1        Loopback4099             v2/S   0      30     1          10.22.1.1
10.22.1.1        LISP0.4099               v2/S   0      30     1          10.22.1.1
10.20.1.1        Vlan222                  v2/S   0      30     1          10.20.1.1
Border#

Border# show ip pim vrf VN3 tunnel
Tunnel5
  Type       : PIM Encap
  RP         : 10.22.1.1*
  Source     : 10.22.1.1
  State      : UP
  Last event : Created (18:06:00)
Tunnel7*
  Type       : PIM Decap
  RP         : 10.22.1.1*
  Source     : -
  State      : UP
  Last event : Created (18:06:00
```

View the multicast groups in the VRFs:

```
FabricEdge# show ip mfib vrf VN3
Entry Flags:    C - Directly Connected, S - Signal, IA - Inherit A flag,
                ET - Data Rate Exceeds Threshold, K - Keepalive
                DDE - Data Driven Event, HW - Hardware Installed
                ME - MoFRR ECMP entry, MNE - MoFRR Non-ECMP entry, MP - MFIB
                MoFRR Primary, RP - MRIB MoFRR Primary, P - MoFRR Primary
                MS  - MoFRR  Entry in Sync, MC - MoFRR entry in MoFRR Client,
                e   - Encap helper tunnel flag.
I/O Item Flags: IC - Internal Copy, NP - Not platform switched,
                NS - Negate Signalling, SP - Signal Present,
                A - Accept, F - Forward, RA - MRIB Accept, RF - MRIB Forward,
                MA - MFIB Accept, A2 - Accept backup,
                RA2 - MRIB Accept backup, MA2 - MFIB Accept backup

Forwarding Counts: Pkt Count/Pkts per second/Avg Pkt Size/Kbits per second
Other counts:      Total/RPF failed/Other drops
I/O Item Counts:   HW Pkt Count/FS Pkt Count/PS Pkt Count   Egress Rate in pps
VRF VN5
(*,224.0.0.0/4) Flags: C HW
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   HW Forwarding:   NA/NA/NA/NA, Other: NA/NA/NA
(*,224.0.1.40) Flags: C HW
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   HW Forwarding:   NA/NA/NA/NA, Other: NA/NA/NA
   LISP0.4099 Flags: A NS
   Loopback4099 Flags: F IC NS
     Pkts: 0/0/0    Rate: 0 pps
(*,232.0.0.0/8) Flags: HW
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   HW Forwarding:   NA/NA/NA/NA, Other: NA/NA/NA
(12.12.12.124,232.1.1.1) Flags: HW
   SW Forwarding: 0/0/0/0, Other: 0/0/0
   HW Forwarding:   NA/NA/NA/NA, Other: NA/NA/NA
   LISP0.4099 Flags: A
   Vlan20 Flags: F NS
     Pkts: 0/0/0    Rate: 0 pps
```

Verify Underlay SSM Configuration

PART IV

LISP VXLAN Fabric Security

CHAPTER 11

Configuring Authentication Authorization and Accounting Services

The fabric network devices are configured with Authentication, Authorization, and Accounting (AAA) policies to provide secure fabric access to the endpoints. Authentication is the process of establishing and confirming the identity of a client requesting access to the network. Authorization is the process of authorizing access to some set of network resources. Accounting is process of recording what was done and accessed by the client. The AAA policies are enforced at the access layer of the network (the fabric edge node to which an endpoint connects), using SGTs for segmentation within the virtual network and dynamic VLAN assignments for mapping endpoints to the virtual networks.

Configure Username and Password on the Switch

You can configure username and password pairs, which are locally stored on the switch. These pairs are assigned to lines or ports and authenticate each user before that user can access the switch. If you have defined privilege levels, you can also assign a specific privilege level (with associated rights and privileges) to each username and password pair.

To configure a local username and password on the switch, perform this task:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. Enter your password, if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	**username** *name* [**privilege** *level*] {**password** { *encryption_type password* }
	Example:
	`Device(config)# username admin privilege 15 password 7 user-password`
		Sets the username, privilege level, and password for each user.
		• For *name*, specify the user ID as one word or the MAC address. Spaces and quotation marks are not allowed.
		• (Optional) For *level*, specify the privilege level the user has after gaining access. The range is 0 to 15. Level 15 gives privileged EXEC mode access. Level 1 gives user EXEC mode access.
		• For *encryption-type*, enter 0 to specify that an unencrypted password will follow. Enter 7 to specify that a hidden password will follow.
		• For password, specify the password the user must enter to gain access to the Switch. The password must be from 1 to 25 characters, can contain embedded spaces, and must be the last option specified in the **username** command.

Step 4	**enable secret** [**level** *level*] {*password* | *encryption-type encrypted-password*}
	Example:
	`Device(config)# enable secret level 1 secret-pwd`
		Defines a secret password, which is saved using a nonreversible encryption method.
		• (Optional) For *level*, the range is from 0 to 15. Level 1 is normal user EXEC mode privileges. The default level is 15 (privileged EXEC mode privileges).
		• For *password*, specify a string from 1 to 25 alphanumeric characters. The string cannot start with a number, is case sensitive, and allows spaces but ignores leading spaces. By default, no password is defined.
		• (Optional) For *encryption-type*, enter either 0, or 5, or 8, or 9. If you specify an encryption type, you must provide an encrypted password—an encrypted password that you copy from another switch configuration.

	Command or Action	Purpose
		• 0: Specifies an UNENCRYPTED password will follow • 5: Specifies a MD5 HASHED secret will follow • 8: Specifies a PBKDF2 HASHED secret will follow • 9: Specifies a SCRYPT HASHED secret will follow **Note** If you specify an encryption type and then enter a clear text password, you can not re-enter privileged EXEC mode. You cannot recover a lost encrypted password by any method.

Step 5	**end**
	Example:
	`Device(config)# end`
		Exits the configuration mode and returns to privileged EXEC mode.

Configure Login Authentication Using AAA

The AAA security services facilitate a variety of login authentication methods. Use the **aaa authentication login** command to enable AAA authentication no matter which of the supported login authentication methods you decide to use. With the **aaa authentication login** command, you create one or more lists of authentication methods that are tried at login. These lists are applied using the **login authentication** line configuration command.

To configure login authentication by using AAA, use the following commands beginning in global configuration mode:

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.

	Command or Action	Purpose
Step 3	**aaa new-model** **Example:** `Device(config)# aaa new-model`	Enables AAA.

Step 4	**aaa authentication login**{**default** | *list-name*} *method1*[*method2...*]
	Example:
	`Device(config)# aaa authentication login default local`
	`Device(config)# aaa authentication login cts-list group client-radius-group local`
		Creates a local authentication list.

Step 5	**line** [**aux** | **console** | **tty** | **vty**] **line-number** [**ending-line-number**] **Example:** `Device(config)# line vty 1`	Enters line configuration mode for the lines to which you want to apply the authentication list.
Step 6	**login local** **Example:** `Device(config-line)# login local`	Enables local password checking at login time. Authentication is based on the username password that is specified earlier.

Step 7	**end**
	Example:
	`Device(config-line)# end`
		Exits line configuration mode and returns to privileged EXEC mode.

Configure 802.1x Authentication Using AAA

To configure dot1x authentication by using AAA, use the following commands beginning in global configuration mode:

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	**aaa new-model** **Example:**	Enables AAA.

	Command or Action	Purpose
	`Device(config)# aaa new-model`	
Step 4	**aaa authentication dot1x** { **default**} *method1* **Example:** `Device(config)# aaa authentication dot1x default group client-radius-group`	Enables AAA accounting and to create method lists defining specific accounting methods on a per-line or per-interface basis for IEEE 802.1x sessions. Creates an IEEE 802.1x authentication method list. To create a default list that is used when a named list is not specified in the **authentication** command, use the **default** keyword followed by the method that is to be used in default situations. The default method list is automatically applied to all ports. For *method1*, enter the **group radius** keywords to use the list of all RADIUS servers for authentication. **Note** Though other keywords are visible in the command-line help string, only the **group radius** keywords are supported.
Step 5	**dot1x system-auth-control** **Example:** `Device(config)# dot1x system-auth-control`	Globally enables 802.1x port-based authentication.

Step 6	**end**
	Example:
	`Device(config)# end`
		Exits the configuration mode and returns to privileged EXEC mode.

Configure AAA Authorization Using Named Method Lists

To configure AAA authorization using named method lists, use the following commands beginning in global configuration mode:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	**aaa authorization** {**auth-proxy** | **network** | **exec** | **commands** *level* | **reverse-access** | **configuration** | **ipmobile**} {**default** | *list-name*} [*method1* [*method2...*]]
	Example:
	`Device(config)# aaa authorization exec default local`
	`Device(config)# aaa authorization network default group client-radius-group`
	`Device(config)# aaa authorization network cts-list group client-radius-group`
		Creates an authorization method list for a particular authorization type and enable authorization.

Step 4	Do one of the following:
	• **line** [**aux** | **console** | **tty** | **vty**] *line-number* [*ending-line-number*]
	• **interface** *interface-type interface-number*
	Example:
	`Device(config)# line 1`
	`Device(config)# interface gigabitethernet 0/1/1`
		Enters the line configuration mode for the lines to which you want to apply the authorization method list.
		Alternately, enters the interface configuration mode for the interfaces to which you want to apply the authorization method list.

Step 5	Do one of the following:
	• **authorization**{**arap** | **commands** *level* | **exec** | **reverse-access**} {**default** | *list-name*}
	• **ppp authorization**{**default** | *list-name*}
	Example:
	`Device(config-line)# authorization commands default`
	`Device(config-if)# ppp authorization default`
		Applies the authorization list to a line or set of lines.
		Alternately, applies the authorization list to an interface or set of interfaces.

Step 6	**end**
	Example:
	`Device(config-line)# end`
	`Device(config-if)# end`
		Exits line configuration mode and returns to privileged EXEC mode.
		Exits interface configuration mode and returns to privileged EXEC mode.

Configure AAA Accounting Using Named Method Lists

To configure AAA Accounting using named method lists, perform the following steps:

Note System accounting does not use named method lists. For system accounting, define only the default method list.

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	**aaa accounting identity** { *name* | **default** } **start-stop** { **broadcast group** { *name* | **radius** | **tacacs+**} [**group** { *name* | **radius** | **tacacs+**} ...] | **group** { *name* | **radius** | **tacacs+**} [**group** { *name* | **radius** | **tacacs+**} ...]}
	Example:
	`Device(config)# aaa accounting Identity default start-stop group client-radius-group`
	`Device(config)# aaa accounting update newinfo periodic 2880`
		Enables accounting for IEEE 802.1x, MAC authentication bypass (MAB), and web authentication sessions

Step 4	Do one of the following: • **line** [**aux** | **console** | **tty** | **vty**] *line-number* [*ending-line-number*] • **interface** *interface-type interface-number* **Example:** `Device(config)# line aux line1`	Enters the line configuration mode for the lines to which the accounting method list is applied. or Enters the interface configuration mode for the interfaces to which the accounting method list is applied.
Step 5	Do one of the following: • **accounting** {**arap** | **commands** *level* | **connection** | **exec**} {**default** | *list-name*} • **ppp accounting**{**default** | *list-name*} **Example:** `Device(config-line)# accounting arap default`	Applies the accounting method list to a line or set of lines. or Applies the accounting method list to an interface or set of interfaces.

Step 6	**end**
	Example:
	`Device(config-line)# end`
		(Optional) Exits line configuration mode and returns to privileged EXEC mode.

Configure CoA on the Device

Follow these steps to configure CoA on a device.

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	**aaa new-model** **Example:** `Device(config)# aaa new-model`	Enables AAA.
Step 4	**aaa server radius dynamic-author** **Example:** `Device(config)# aaa server radius dynamic-author`	Configures the device as an authentication, authorization, and accounting (AAA) server to facilitate interaction with an external policy server, and enters dynamic authorization local server configuration mode.
Step 5	**client** {*ip-address* | *name*} [**vrf** *vrfname*] [**server-key** *string*] **Example:** `Device(config-locsvr-da-radius)# client 172.16.2.1 server-key 7 server-pwd`	Specifies a RADIUS client from which a device will accept CoA and disconnect requests. Specify all the Policy Administration Nodes (PANs) or Policy Services Nodes (PSNs), if you have a multi-node deployment.

Step 6	**end**
	Example:
	`Device(config-locsvr-da-radius)# end`
		Exits dynamic authorization local server configuration mode and returns to privileged EXEC mode.

Identify the RADIUS Server Host

To apply these settings globally to all RADIUS servers communicating with the device, use the three unique global configuration commands: **radius-server timeout**, **radius-server retransmit**, and **key** *string*.

You can configure the device to use AAA server groups to group existing server hosts for authentication.

You also need to configure some settings on the RADIUS server. These settings include the IP address of the device and the key string to be shared by both the server and the device.

Follow these steps to configure per-server RADIUS server communication.

Before you begin

If you configure both global and per-server functions (timeout, retransmission, and key commands) on the device, the per-server timer, retransmission, and key value commands override global timer, retransmission, and key value commands.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. TEST Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**radius server** *server name* **Example:** `Device(config)# radius server radius_172.16.2.1`	Specifies the name for the RADIUS server configuration for Protected Access Credential (PAC) provisioning, and enters RADIUS server configuration mode.

Step 4	**address** {**ipv4** | **ipv6**}*ip address*{ **auth-port** *port number* | **acct-port** *port number*}
	Example:
	`Device(config-radius-server)# address ipv4 172.16.2.1 auth-port 1812 acct-port 1813`
		(Optional) Specifies the RADIUS server parameters.
		For **auth-port** *port-number*, specify the UDP destination port for authentication requests. The default is 1645. The range is 0 to 65536.
		For **acct-port** *port-number*, specify the UDP destination port for accounting requests. The default is 1646.

Step 5	**timeout** *seconds* **Example:** `Device(config-radius-server)# timeout 2`	(Optional) Specifies the time interval that the device waits for the RADIUS server to reply before sending a request again. The range is 1 to 1000. This setting overrides the **radius-server timeout** global configuration command setting. We recommend a timeout value of two seconds.
Step 6	**retransmit** *value* **Example:** `Device(config-radius-server)# retransmit 1`	(Optional) Specifies the number of times a RADIUS request is resent when the server is not responding or responding slowly. The range is 1 to 100. This setting overrides the **radius-server retransmit** global configuration command setting.

	Command or Action	Purpose

Step 7	**automate-tester username** *user* [**ignore-auth-port**] [**ignore-acct-port**] [**idle-time** *minutes*] **probe-on**
	Example:
	`Device(config-radius-server)#`
	`automate-tester username dummy`
	`ignore-acct-port probe-on`
		Enables RADIUS automated testing for a non-default VRF.

Step 8	**pac key** *encryption-key*
	Example:
	`Device(config-radius-server)#` **`pac key`**
	`7 pac-key`
		Specifies the Protected Access Credential (PAC) encryption key.

Step 9	**exit** **Example:** `Device(config-radius-server)#` **`exit`**	Exits RADIUS server configuration mode, and enters global configuration mode.

Step 10	**radius-server attribute** *attribute* {**on-for-login-auth** | **support-multiple** | **include-in-access-req** | **access-request include** | **mac format ietf upper-case** | **send nas-port-detail mac-only**}
	Example:
	`Device(config)#` **`radius-server attribute`**
	` 6 on-for-login-auth`
	`Device(config)#` **`radius-server attribute`**
	` 6 support-multiple`
	`Device(config)#` **`radius-server attribute`**
	` 8 include-in-access-req`
	`Device(config)#` **`radius-server attribute`**
	` 25 access-request include`
	`Device(config)#` **`radius-server attribute`**
	` 31 mac format ietf upper-case`
	`Device(config)#` **`radius-server attribute`**
	` 31 send nas-port-detail mac-only`
		Provides for the presence of the Service-Type attribute in RADIUS Access-Accept messages.

Step 11	**radius-server dead-criteria** [**time** *seconds*] [**tries** *number-of-tries*]
	Example:
	`Device(config)#` **`radius-server`**
	`dead-criteria time 5 tries 3`
		Forces one or both of the criteria, used to mark a RADIUS server as dead, to be the indicated constant.

Step 12	**radius-server deadtime** *minutes*
	Example:
	`Device(config)#` **`radius-server deadtime`**
	` 3`
		Improves RADIUS response times when some servers might be unavailable and causes the unavailable servers to be skipped immediately.

Step 13	**end** **Example:**	Exits global configuration mode and enters privileged EXEC mode.

	Command or Action	Purpose
	`Device(config)# end`	

Configure the Source Interface on RADIUS Server Group

Follow these steps to configure the source interface and for authentication and accounting on RADIUS server groups:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**aaa group server radius** *group_name* **Example:** `Device(config)# aaa group server radius client-radius-group`	Defines the RADIUS server group configuration and enters RADIUS server group configuration mode.
Step 4	**server name** *name* **Example:** `Device(config-sg-radius)# server name radius_172.16.2.1`	Associates the RADIUS server to the server group.
Step 5	{**ip** | **ipv6**} **radius source-interface** *type number* **Example:** `Device(config-sg-radius)# ip radius source-interface Loopback0`	Specifies an interface to use for the source address in RADIUS server.

Step 6	**end**
	Example:
	`Device(config-radius-server)# end`
		Exits RADIUS server mode and enters privileged EXEC mode.

Configure IBNS

To configure IBNS, perform the following tasks:

Configure a Control Class

A control class defines the conditions under which the actions of a control policy are executed. You define whether all, any, or none of the conditions must evaluate true to execute the actions of the control policy. Control classes are evaluated based on the event specified in the control policy.

Note This procedure shows all of the match conditions that you can configure in a control class. You must specify at least one condition in a control class to make it valid. All other conditions, and their corresponding steps, are optional (steps 4 through 18 below).

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	**class-map type control subscriber** {**match-all** | **match-any** | **match-none**} *control-class-name* **Example:** `Device(config)# class-map type control subscriber match-all DOT1X_NO_AGENT`	Creates a control class and enters control class-map filter mode. • **match-all**: All of the conditions in the control class must evaluate true. • **match-any**: At least one of the conditions in the control class must evaluate true. • **match-none**: All of the conditions in the control class must evaluate false.
Step 4	{**match** | **no-match**} **activated-service-template** *template-name* **Example:** `Device(config-filter-control-classmap)# match activated-service-template SVC_1`	(Optional) Creates a condition that evaluates true based on the service template activated on a session.

Step 5	{**match** | **no-match**} **authorization-status** {**authorized** | **unauthorized**}
	Example:
	`Device(config-filter-control-classmap)# match authorization-status authorized`
		(Optional) Creates a condition that evaluates true based on a session's authorization status.

Step 6	{**match** | **no-match**} **authorizing-method-priority** {**eq** | **gt** | **lt**} *priority-value*	(Optional) Creates a condition that evaluates true based on the priority of the authorization method.

	Command or Action	Purpose
	Example: `Device(config-filter-control-classmap)# match authorizing-method-priority eq 10`	• **eq**: Current priority is equal to *priority-value*. • **gt**: Current priority is greater than *priority-value*. • **lt**: Current priority is less than *priority-value*. • *priority-value*: Priority value to match. Range: 1 to 254, where 1 is the highest priority and 254 is the lowest.

Step 7	{**match** | **no-match**} **client-type** {**data** | **switch** | **video** | **voice**}
	Example:
	`Device(config-filter-control-classmap)# match client-type data`
		(Optional) Creates a condition that evaluates true based on an event's device type.

Step 8	{**match** | **no-match**} **current-method-priority** {**eq** | **gt** | **lt**} *priority-value* **Example:** `Device(config-filter-control-classmap)# match current-method-priority eq 10`	(Optional) Creates a condition that evaluates true based on the priority of the current authentication method.
Step 9	{**match** | **no-match**} **ip-address** *ip-address* **Example:** `Device(config-filter-control-classmap)# match ip-address 10.10.10.1`	(Optional) Creates a condition that evaluates true based on an event's source IPv4 address.
Step 10	{**match** | **no-match**} **ipv6-address** *ipv6-address* **Example:** `Device(config-filter-control-classmap)# match ipv6-address FE80::1`	(Optional) Creates a condition that evaluates true based on an event's source IPv6 address.
Step 11	{**match** | **no-match**} **mac-address** *mac-address* **Example:** `Device(config-filter-control-classmap)# match mac-address aabb.cc00.6500`	(Optional) Creates a condition that evaluates true based on an event's MAC address.

Step 12	{**match** | **no-match**} **method** {**dot1x** | **mab** | **webauth**}
	Example:
	`Device(config-filter-control-classmap)# match method dot1x`
		(Optional) Creates a condition that evaluates true based on an event's authentication method.

	Command or Action	Purpose

Step 13	{**match** | **no-match**} **port-type** {**l2-port** | **l3-port** | **dot11-port**}
	Example:
	`Device(config-filter-control-classmap)# match port-type l2-port`
		(Optional) Creates a condition that evaluates true based on an event’s interface type.

Step 14	{**match** | **no-match**} **result-type** [**method** {**dot1x** | **mab** | **webauth**}] *result-type* **Example:** `Device(config-filter-control-classmap)# match result-type agent-not-found`	(Optional) Creates a condition that evaluates true based on the specified authentication result. • To display the available result types, use the question mark (**?**) online help function.
Step 15	{**match** | **no-match**} **service-template** *template-name* **Example:** `Device(config-filter-control-classmap)# match service-template svc_1`	(Optional) Creates a condition that evaluates true based on an event’s service template.
Step 16	{**match** | **no-match**} **tag** *tag-name* **Example:** `Device(config-filter-control-classmap)# match tag tag_1`	(Optional) Creates a condition that evaluates true based on the tag associated with an event.
Step 17	{**match** | **no-match**} **timer** *timer-name* **Example:** `Device(config-filter-control-classmap)# match timer restart`	(Optional) Creates a condition that evaluates true based on an event’s timer.
Step 18	{**match** | **no-match**} **username** *username* **Example:** `Device(config-filter-control-classmap)# match username josmiths`	(Optional) Creates a condition that evaluates true based on an event’s username.

Step 19	**end**
	Example:
	`Device(config-filter-control-classmap)# end`
		(Optional) Exits control class-map filter configuration mode and returns to privileged EXEC mode.

Step 20	**show class-map type control subscriber** {**all** | **name** *control-class-name*} **Example:** `Device# show class-map type control subscriber all`	(Optional) Displays information about Identity-Based Networking Services control classes.

Example: Control Class

The following example shows a control class that is configured with two match conditions:

```
class-map type control subscriber match-all DOT1X_NO_AGENT
 match method dot1x
 match result-type agent-not-found
```

Configure a Control Policy

Control policies determine the actions that the system takes in response to specified events and conditions. The control policy contains one or more control policy rules that associate a control class with one or more actions. The actions that you can configure in a policy rule depend on the type of event that you specify.

Note This task includes all of the actions that you can configure in a control policy regardless of the event. All of these actions, and their corresponding steps, are optional (steps 6 through 21 below). To display the supported actions for a particular event, use the question mark (**?**) online help function.

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	**policy-map type control subscriber** *control-policy-name*
	Example:
	`Device(config)#` **`policy-map type control`**
	`PMAP_DefaultWiredDot1xClosedAuth_1X_MAB`
		Defines a control policy for subscriber sessions.

Step 4	**event** *event-name* [**match-all** | **match-first**]
	Example:
	`Device(config-event-control-policymap)#`
	`event session-started match-all`
		Specifies the type of event that triggers actions in a control policy if conditions are met.
		• **match-all** is the default behavior.
		• To display the available event types, use the question mark (**?**) online help function. For a complete description of event types, see the **event** command.

	Command or Action	Purpose
Step 5	*priority-number* **class** {*control-class-name* | **always**} [**do-all** | **do-until-failure** | **do-until-success**] **Example:** `Device(config-class-control-policymap)# 10 class always do-until-failure`	Associates a control class with one or more actions in a control policy. • A named control class must first be configured before specifying it with the *control-class-name* argument. • **do-until-failure** is the default behavior.
Step 6	*action-number* **activate** {**policy type control subscriber** *control-policy-name* [**child** [**no-propagation** | **concurrent**] | **service-template** *template-name* [**aaa-list** *list-name*] [**precedence** *number*] [**replace-all**]} **Example:** `Device(config-action-control-policymap)# 10 activate service-template DefaultCriticalAuthVlan_SRV_TEMPLATE`	(Optional) Activates a control policy or service template on a subscriber session.
Step 7	*action-number* **authenticate using** {**dot1x** | **mab** | **webauth**} [**aaa** {**authc-list** *authc-list-name* | **authz-list** *authz-list-name*]} [**merge**] [**parameter-map** *map-name*] [**priority** *priority-number*] [**replace** | **replace-all**] [**retries** *number* {**retry-time** *seconds*}] **Example:** `Device(config-action-control-policymap)# 20 authenticate using dot1x retries 2 retry-time 0 priority 10`	(Optional) Initiates the authentication of a subscriber session using the specified method.
Step 8	*action-number* **authentication-restart** *seconds* **Example:** `Device(config-action-control-policymap)# 20 authentication-restart 60`	(Optional) Sets a timer to restart the authentication process after an authentication or authorization failure.
Step 9	*action-number* **authorize** **Example:** `Device(config-action-control-policymap)# 30 authorize`	(Optional) Initiates the authorization of a subscriber session.
Step 10	*action-number* **clear-authenticated-data-hosts-on-port** **Example:** `Device(config-action-control-policymap)# 20 clear-authenticated-data-hosts-on-port`	(Optional) Clears authenticated data hosts on a port after an authentication failure.

	Command or Action	Purpose
Step 11	*action-number* **clear-session** **Example:** `Device(config-action-control-policymap)# 10 clear-session`	(Optional) Clears an active subscriber session.
Step 12	*action-number* **deactivate** {**policy type control subscriber** *control-policy-name* | **service-template** *template-name*} **Example:** `Device(config-action-control-policymap)# 20 deactivate service-template`	(Optional) Deactivates a control policy or service template on a subscriber session.
Step 13	*action-number* **err-disable** **Example:** `Device(config-action-control-policymap)# 10 err-disable`	(Optional)Temporarily disables a port after a session violation event.
Step 14	*action-number* **pause reauthentication** **Example:** `Device(config-action-control-policymap)# 40 pause reauthentication`	(Optional) Pauses reauthentication after an authentication failure.
Step 15	*action-number* **protect** **Example:** `Device(config-action-control-policymap)# 10 protect`	(Optional) Silently drops violating packets after a session violation event.
Step 16	*action-number* **replace** **Example:** `Device(config-action-control-policymap)# 10 replace`	(Optional) Clears the existing session and creates a new session after a violation event.
Step 17	*action-number* **restrict** **Example:** `Device(config-action-control-policymap)# 10 restrict`	(Optional) Drops violating packets and generates a syslog entry after a session violation event.
Step 18	*action-number* **resume reauthentication** **Example:** `Device(config-action-control-policymap)# 10 resume reauthentication`	(Optional) Resumes the reauthentication process after an authentication failure.
Step 19	*action-number* **set-timer** *timer-name seconds* **Example:** `Device(config-action-control-policymap)# 20 set-timer RESTART 60`	(Optional) Starts a named policy timer.

	Command or Action	Purpose

Step 20	*action-number* **terminate** {**dot1x** | **mab** | **webauth**}
	Example:
	`Device(config-action-control-policymap)# 10 terminate mab`
		(Optional) Terminates an authentication method on a subscriber session.

Step 21	*action-number* **unauthorize** **Example:** `Device(config-action-control-policymap)# 20 unauthorize`	(Optional) Removes all authorization data from a subscriber session.

Step 22	**end**
	Example:
	`Device(config-action-control-policymap)# end`
		(Optional) Exits control policy-map action configuration mode and returns to privileged EXEC mode.

Step 23	**show policy-map type control subscriber** {**all** | **name** *control-policy-name*} **Example:** `Device# show policy-map type control subscriber name PMAP_DefaultWiredDot1xClosedAuth_1X_MAB`	(Optional) Displays information about identity control policies.

Example: Control Policy

The following example shows a simple control policy with the minimum configuration necessary for initiating authentication:

```
policy-map type control subscriber POLICY_1
 event session-started match-all
  10 class always do-until-failure
   10 authenticate using dot1x
```

Configure Interface Templates

You can create an interface template using the **template** command in global configuration mode. In template configuration mode, enter the required commands. The following commands can be entered in template configuration mode:

Note

- System builtin templates are not displayed in the running configuration. These templates show up in the running configuration only if you edit them.
- When you configure an interface template, we recommend that you enter all the required dependent commands on the same template. we do not recommend to configure the dependent commands on two different templates.

Command	Description
access-session	Configures access session specific interface commands.
authentication	Configures authentication manager Interface Configuration commands.
carrier-delay	Configures delay for interface transitions.
dampening	Enables event dampening.
default	Sets a command to its defaults.
description	Configures interface-specific description.
dot1x	Configures interface configuration commands for IEEE 802.1X.
hold-queue	Sets hold queue depth.
ip	Configures IP template.
keepalive	Enables keepalive.
load-interval	Specifies interval for load calculation for an interface.
mab	Configures MAC authentication bypass Interface.
peer	Configures peer parameters for point to point interfaces.
service-policy	Configures CPL service policy.
source	Gets configurations from another source.
spanning-tree	Configures spanning tree subsystem.
storm-control	Configures storm control.
subscriber	Configures subscriber inactivity timeout value.
switchport	Sets switching mode configurations.
trust	Sets trust value for the interface.

To configure interface templates, perform this task:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	**template** *name*
	Example:
	`Device(config)# template DefaultWiredDot1xClosedAuth`
	
	`dot1x pae authenticator`
	` dot1x timeout supp-timeout 7`
	` dot1x max-req 3`
	` switchport mode access`
	` switchport voice vlan 2046`
	` mab`
	` access-session closed`
	` access-session port-control auto`
	` authentication periodic`
	` authentication timer reauthenticate server`
	` service-policy type control subscriber PMAP_DefaultWiredDot1xClosedAuth_1X_MAB`
		Creates a user template and enters template configuration mode.
		Note Builtin template are system-generated.

Step 4	**end**
	Example:
	`Device(config-template)# end`
		Returns to privileged EXEC mode.

Enabling Central Web Authentication

Web authentication allows users to get authenticated through a web browser on a client, with minimal configuration on the client side. Central web authentication is typically used for guest authentication. A RADIUS server (such as Cisco ISE) is mandatory when you enable central web authentication.

Perform the following task on the fabric edge node to redirect the clients based on the HTTP traffic.

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	**ip http server** **Example:**	Enables the HTTP server. The web-based authentication feature uses the HTTP server to

	Command or Action	Purpose
	`Device(config)# ip http server`	communicate with the hosts for user authentication.

Step 4	**end**
	Example:
	`Device(config)#end`
		Returns to privileged EXEC mode.

Create Extended Named ACLs

Follow these steps to create an extended ACL using names:

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	**ip access-list extended** *name* **Example:** `Device(config)# ip access-list extended ACL_WEBAUTH_REDIRECT`	Defines an extended IPv4 access list using a name, and enter access-list configuration mode. The name can be a number from 100 to 199.

Step 4	*sequence-number* {**deny** | **permit**} *protocol* {*source* [*source-wildcard*] | **host** *source* | **any**} {*destination* [*destination-wildcard*] | host *destination* | **any**} [**precedence** *precedence*] [**tos** *tos*] [**established**] [**log**] [**time-range** *time-range-name*]
	Example:
	`Device(config-ext-nacl)# 260 deny ip any host 172.16.2.1`
	`Device(config-ext-nacl)# 500 permit tcp any any eq www`
	`Device(config-ext-nacl)# 600 permit tcp any any eq 443`
	`Device(config-ext-nacl)# 700 permit tcp any any eq 8443`
	`Device(config-ext-nacl)# 800 deny udp any any eq domain`
	`Device(config-ext-nacl)# 900 deny udp any eq bootpc any eq bootps`
		In access-list configuration mode, specify the sequence number (1 to 32767) and the conditions that are to be allowed or denied. Use the **log** keyword to get access list logging messages, including violations.
		• **host** *source*: A source and source wildcard of *source* 0.0.0.0.
		• **host** *destintation*: A destination and destination wildcard of *destination* 0.0.0.0.
		• **any**: A source and source wildcard or destination and destination wildcard of 0.0.0.0 255.255.255.255.

	Command or Action	Purpose

Step 5	**end**
	Example:
	`Device(config-ext-nacl)# end`
		Exits access-list configuration mode and returns to privileged EXEC mode.

When you are creating extended ACLs, remember that, by default, the end of the ACL contains an implicit deny statement for everything if it did not find a match before reaching the end. For standard ACLs, if you omit the mask from an associated IP host address access list specification, 0.0.0.0 is assumed to be the mask.

After you create an ACL, any additions are placed at the end of the list. You cannot selectively add ACL entries to a specific ACL. However, you can use **no permit** and **no deny** access-list configuration mode commands to remove entries from a named ACL.

Being able to selectively remove lines from a named ACL is one reason you might use named ACLs instead of numbered ACLs.

What to do next

After creating a named ACL, you can apply it to interfaces or to VLANs .

Configure IPv6 ACLs

To filter IPv6 traffic, perform this procedure.

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	**ipv6 access-list** {*list-name* | **log-update threshold** | **role-based** *list-name*}
	Example:
	`Device(config)# ipv6 access-list IPV6_PRE_AUTH_ACL`
		Defines an IPv6 ACL name, and enters IPv6 access list configuration mode.

Step 4	*sequence-number* {**deny** | **permit**} protocol {*source-ipv6-prefix/* |*prefix-length* |**any threshold**| **host** *source-ipv6-address*} [operator [*port-number*]] { *destination-ipv6-prefix/ prefix-length* | **any** | **host** *destination-ipv6-address*} [operator	Specifies permit or deny conditions for an IPv6 ACL. • For protocol, enter the name or number of an IP: **ahp**, **esp**, **icmp**, **ipv6**, **pcp**, **stcp**, **tcp**, or **udp**, or an integer in the range 0 to 255 representing an IPv6 protocol number.

	Command or Action	Purpose

	[*port-number*]][**dscp** *value*] [**fragments**] [**log**] [**log-input**][**sequence** *value*] [**time-range** *name*]
	Example:
	`Device(config-ipv6-acl)# ` **`sequence 10 permit udp any any eq bootps`**
	`Device(config-ipv6-acl)# ` **`sequence 20 permit udp any any eq bootpc`**
	`Device(config-ipv6-acl)# ` **`sequence 30 permit udp any any eq domain`**
	`Device(config-ipv6-acl)# ` **`sequence 40 deny ipv6 any any`**
		• The *source-ipv6-prefix/prefix-length* or *destination-ipv6-prefix/ prefix-length* is the source or destination IPv6 network or class of networks for which to set deny or permit conditions, specified in hexadecimal and using 16-bit values between colons (see RFC 2373).
		• Enter **any** as an abbreviation for the IPv6 prefix ::/0.
		• For **host** *source-ipv6-address* or *destination-ipv6-address*, enter the source or destination IPv6 host address for which to set deny or permit conditions, specified in hexadecimal using 16-bit values between colons.
		• (Optional) For operator, specify an operand that compares the source or destination ports of the specified protocol. Operands are **lt** (less than), **gt** (greater than), **eq** (equal), **neq** (not equal), and **range**.
		If the operator follows the *source-ipv6-prefix/prefix-length* argument, it must match the source port. If the operator follows the *destination-ipv6-prefix/prefix-length* argument, it must match the destination port.
		• (Optional) The **port-number** is a decimal number from 0 to 65535 or the name of a TCP or UDP port. You can use TCP port names only when filtering TCP. You can use UDP port names only when filtering UDP.
		• (Optional) Enter **dscp** value to match a differentiated services code point value against the traffic class value in the Traffic Class field of each IPv6 packet header. The acceptable range is from 0 to 63.
		• (Optional) Enter **fragments** to check noninitial fragments. This keyword is visible only if the protocol is ipv6.
		• (Optional) Enter **log** to cause an logging message to be sent to the console about the packet that matches the entry. Enter **log-input** to include the input interface in

	Command or Action	Purpose
		the log entry. Logging is supported only for router ACLs. • (Optional) Enter **sequence** *value* to specify the sequence number for the access list statement. The acceptable range is from 1 to 4,294,967,295. • (Optional) Enter **time-range** name to specify the time range that applies to the deny or permit statement.

Step 5	**end**
	Example:
	`Device(config-ipv6-acl)# end`
		Exits IPv6 access list configuration mode and returns to privileged EXEC mode.

Step 6	**show ipv6 access-list** **Example:** `Device# show ipv6 access-list`	Verifies that IPv6 ACLs are configured correctly.

Configure Host Onboarding Interfaces

To configure host onboarding interfaces, perform this task:

Note The example configurations in this procedure are for Closed Authentication mode on the interface.

You can follow the same procedure for the Open Authentication and Low Impact authentication modes on the interface. Whatever interface configuration mode you deploy, ensure you use the respective dot1x interface template (DefaultWiredDot1xOpenAuth or DefaultWiredDot1xLowImpactAuth).

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	**interface** *type number* **Example:**	Specifies the interface type and number and enters interface configuration mode.

	Command or Action	Purpose
	`Device(config)# interface GigabitEthernet1/0/10`	
Step 4	**switchport access vlan** *vlan-id* **Example:** `Device(config-if)# switchport access vlan 50`	Assigns the port to a VLAN. Valid VLAN IDs are 1 to 4094.
Step 5	**switchport mode access** **Example:** `Device(config-if)# switchport mode access`	Defines the VLAN membership mode for the port (Layer 2 access port).
Step 6	**switchport voice vlan** *vlan-id* **Example:** `Device(config-if)# switchport voice vlan 51`	Configures the voice VLAN. Valid VLAN IDs are 1 to 4094.
Step 7	**device-tracking attach-policy** *policy_name* **Example:** `Device(config-if)# device-tracking attach-policy IPDT_POLICY`	Attaches the device tracking policy to the specified VLANs across all switch interfaces.
Step 8	**load-interval** *seconds* **Example:** `Device(config-if)# load-interval 30`	Changes the length of time for which data is used to compute load statistics. Value is a multiple of 30, from 30 to 600 (30, 60, 90, 120, and so on). The default is 300 seconds.
Step 9	**access-session inherit disable interface-template-sticky** **Example:** `Device(config-if)# access-session inherit disable interface-template-sticky`	Disables the Autoconf feature on a specific interface.
Step 10	**access-session inherit disable autoconf** **Example:** `Device(config-if)# access-session inherit disable autoconf`	Manually disables Autoconf at the interface level, even when Autoconf is enabled at the global level.
Step 11	**dot1x timeout tx-period** *seconds* **Example:** `Device(config-if)# dot1x timeout tx-period 7`	Configures the number of seconds between retransmission of EAP request ID packets (assuming that no response is received) to the client. The range is from 1 to 65535. The default is 30.
Step 12	**dot1x max-reauth-req** *number* **Example:**	Sets the maximum number of times the authenticator sends an Extensible Authentication Protocol (EAP) request/identity

	Command or Action	Purpose
	`Device(config-if)# dot1x max-reauth-req 3`	frame (assuming that no response is received) to the client. The range is 1 through 10. The default is 2.
Step 13	**no macro auto processing** **Example:** `Device(config-if)# no macro auto processing`	Disables Auto Smartports macros on an interface.
Step 14	**source template** *template* **Example:** `Device(config-if)# source template DefaultWiredDot1xClosedAuth`	Sources the interface template along with the other interface-specific commands for the desired ports. This example is for a Closed Authentication mode of 802.1x deployment. You can also use the Open Authentication or Low Impact authentication modes on the interface. Whatever authentication mode you deploy, ensure you use the correct dot1x interface template (DefaultWiredDot1xOpenAuth or DefaultWiredDot1xLowImpactAuth, which were defined earlier).
Step 15	**spanning-tree portfast** **Example:** `Device(config-if)# spanning-tree portfast`	Enables PortFast mode, where the interface is immediately put into the forwarding state upon linkup without waiting for the timer to expire.
Step 16	**spanning-tree bpduguard enable** **Example:** `Device(config-if)# spanning-tree bpduguard enable`	Enables bridge protocol data unit (BPDU) guard on the interface.

Step 17	**end**
	Example:
	`Device(config-if)# end`
		Returns to privileged EXEC mode.

Configuration Example for IEEE 802.1x on Fabric Edge

A fabric edge node is configured as an authenticator to interface with the AAA server or Cisco ISE and authenticate the endpoints. This is a sample configuration for IEEE 802.1x on a fabric edge node; Cisco ISE is configured with an IP address of 172.16.2.1

```
username admin privilege 15 password 7 user-password
enable secret level 1 secret-pwd
!
aaa new-model
dot1x system-auth-control
```

```
aaa session-id common
!
aaa authentication login default local
aaa authentication login cts-list group client-radius-group local
aaa authentication dot1x default group client-radius-group
aaa authorization exec default local
aaa authorization network default group client-radius-group
aaa authorization network cts-list group client-radius-group
aaa accounting Identity default start-stop group client-radius-group
aaa accounting update newinfo periodic 2880
!
aaa server radius dynamic-author
 client 172.16.2.1 server-key 7 server-pwd
!
!
radius server radius_172.16.2.1
 address ipv4 172.16.2.1 auth-port 1812 acct-port 1813
 timeout 2
 retransmit 1
 automate-tester username dummy ignore-acct-port probe-on
 pac key 7 pac-key
!
!
radius-server attribute 6 on-for-login-auth
radius-server attribute 6 support-multiple
radius-server attribute 8 include-in-access-req
radius-server attribute 25 access-request include
radius-server attribute 31 mac format ietf upper-case
radius-server attribute 31 send nas-port-detail mac-only
radius-server dead-criteria time 5 tries 3
radius-server deadtime 3
!
aaa group server radius client-radius-group
 server name radius_172.16.2.1
 ip radius source-interface Loopback0
!
!
!
!
!
ip radius source-interface Loopback0
Identify Based Networking Services(IBNS)
class-map type control subscriber match-all AAA_SVR_DOWN_AUTHD_HOST
 match authorization-status authorized
 match result-type aaa-timeout
!
class-map type control subscriber match-all AAA_SVR_DOWN_UNAUTHD_HOST
 match authorization-status unauthorized
 match result-type aaa-timeout
!
class-map type control subscriber match-all AUTHC_SUCCESS-AUTHZ_FAIL
 match authorization-status unauthorized
 match result-type success
!
class-map type control subscriber match-all DOT1X
 match method dot1x
!
class-map type control subscriber match-all DOT1X_FAILED
 match method dot1x
 match result-type method dot1x authoritative
!
class-map type control subscriber match-all DOT1X_MEDIUM_PRIO
 match authorizing-method-priority gt 20
!
```

```
class-map type control subscriber match-all DOT1X_NO_RESP
 match method dot1x
 match result-type method dot1x agent-not-found
!
class-map type control subscriber match-all DOT1X_TIMEOUT
 match method dot1x
 match result-type method dot1x method-timeout
!
class-map type control subscriber match-any IN_CRITICAL_AUTH
 match activated-service-template DefaultCriticalVoice_SRV_TEMPLATE
!
class-map type control subscriber match-any IN_CRITICAL_AUTH_CLOSED_MODE
 match activated-service-template DefaultCriticalAuthVlan_SRV_TEMPLATE
 match activated-service-template DefaultCriticalVoice_SRV_TEMPLATE
!
class-map type control subscriber match-all MAB
 match method mab
!
class-map type control subscriber match-all MAB_FAILED
 match method mab
 match result-type method mab authoritative
!
class-map type control subscriber match-none NOT_IN_CRITICAL_AUTH
 match activated-service-template DefaultCriticalVoice_SRV_TEMPLATE
!
class-map type control subscriber match-none NOT_IN_CRITICAL_AUTH_CLOSED_MODE
 match activated-service-template DefaultCriticalAuthVlan_SRV_TEMPLATE
 match activated-service-template DefaultCriticalVoice_SRV_TEMPLATE
!
policy-map type control subscriber PMAP_DefaultWiredDot1xClosedAuth_1X_MAB
 event session-started match-all
  10 class always do-until-failure
   10 authenticate using dot1x retries 2 retry-time 0 priority 10
 event authentication-failure match-first
  5 class DOT1X_FAILED do-until-failure
   10 terminate dot1x
   20 authenticate using mab priority 20
  10 class AAA_SVR_DOWN_UNAUTHD_HOST do-until-failure
   10 activate service-template DefaultCriticalAuthVlan_SRV_TEMPLATE
   20 activate service-template DefaultCriticalVoice_SRV_TEMPLATE
   30 authorize
   40 pause reauthentication
  20 class AAA_SVR_DOWN_AUTHD_HOST do-until-failure
   10 pause reauthentication
   20 authorize
  30 class DOT1X_NO_RESP do-until-failure
   10 terminate dot1x
   20 authenticate using mab priority 20
  40 class MAB_FAILED do-until-failure
   10 terminate mab
   20 authentication-restart 60
  50 class DOT1X_TIMEOUT do-until-failure
   10 terminate dot1x
   20 authenticate using mab priority 20
  60 class always do-until-failure
   10 terminate dot1x
   20 terminate mab
   30 authentication-restart 60
 event aaa-available match-all
  10 class IN_CRITICAL_AUTH_CLOSED_MODE do-until-failure
   10 clear-session
  20 class NOT_IN_CRITICAL_AUTH_CLOSED_MODE do-until-failure
   10 resume reauthentication
 event agent-found match-all
```

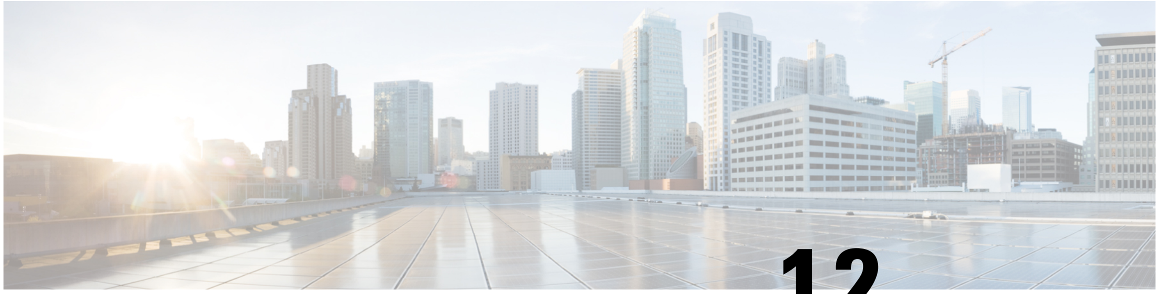
```
  10 class always do-until-failure
   10 terminate mab
   20 authenticate using dot1x retries 2 retry-time 0 priority 10
 event inactivity-timeout match-all
  10 class always do-until-failure
   10 clear-session
 event authentication-success match-all
 event violation match-all
  10 class always do-until-failure
   10 restrict
 event authorization-failure match-all
  10 class AUTHC_SUCCESS-AUTHZ_FAIL do-until-failure
   10 authentication-restart 60
!
policy-map type control subscriber PMAP_DefaultWiredDot1xClosedAuth_MAB_1X
 event session-started match-all
  10 class always do-until-failure
   10 authenticate using mab priority 20
 event authentication-failure match-first
  5 class DOT1X_FAILED do-until-failure
   10 terminate dot1x
   20 authentication-restart 60
  10 class AAA_SVR_DOWN_UNAUTHD_HOST do-until-failure
   10 activate service-template DefaultCriticalAuthVlan_SRV_TEMPLATE
   20 activate service-template DefaultCriticalVoice_SRV_TEMPLATE
   30 authorize
   40 pause reauthentication
  20 class AAA_SVR_DOWN_AUTHD_HOST do-until-failure
   10 pause reauthentication
   20 authorize
  30 class MAB_FAILED do-until-failure
   10 terminate mab
   20 authenticate using dot1x retries 2 retry-time 0 priority 10
  40 class DOT1X_NO_RESP do-until-failure
   10 terminate dot1x
   20 authentication-restart 60
  50 class DOT1X_TIMEOUT do-until-failure
   10 terminate dot1x
   20 authenticate using mab priority 20
  60 class always do-until-failure
   10 terminate mab
   20 terminate dot1x
   30 authentication-restart 60
 event aaa-available match-all
  10 class IN_CRITICAL_AUTH_CLOSED_MODE do-until-failure
   10 clear-session
  20 class NOT_IN_CRITICAL_AUTH_CLOSED_MODE do-until-failure
   10 resume reauthentication
 event agent-found match-all
  10 class always do-until-failure
   10 terminate mab
   20 authenticate using dot1x retries 2 retry-time 0 priority 10
 event inactivity-timeout match-all
  10 class always do-until-failure
   10 clear-session
 event authentication-success match-all
 event violation match-all
  10 class always do-until-failure
   10 restrict
 event authorization-failure match-all
  10 class AUTHC_SUCCESS-AUTHZ_FAIL do-until-failure
   10 authentication-restart 60
!
policy-map type control subscriber PMAP_DefaultWiredDot1xLowImpactAuth_1X_MAB
```

```
 event session-started match-all
  10 class always do-until-failure
   10 authenticate using dot1x retries 2 retry-time 0 priority 10
 event authentication-failure match-first
  5 class DOT1X_FAILED do-until-failure
   10 terminate dot1x
   20 authenticate using mab priority 20
  10 class AAA_SVR_DOWN_UNAUTHD_HOST do-until-failure
   10 activate service-template DefaultCriticalAuthVlan_SRV_TEMPLATE
   20 activate service-template DefaultCriticalVoice_SRV_TEMPLATE
   25 activate service-template DefaultCriticalAccess_SRV_TEMPLATE
   30 authorize
   40 pause reauthentication
  20 class AAA_SVR_DOWN_AUTHD_HOST do-until-failure
   10 pause reauthentication
   20 authorize
  30 class DOT1X_NO_RESP do-until-failure
   10 terminate dot1x
   20 authenticate using mab priority 20
  40 class MAB_FAILED do-until-failure
   10 terminate mab
   20 authentication-restart 60
  50 class DOT1X_TIMEOUT do-until-failure
   10 terminate dot1x
   20 authenticate using mab priority 20
  60 class always do-until-failure
   10 terminate dot1x
   20 terminate mab
   30 authentication-restart 60
 event aaa-available match-all
  10 class IN_CRITICAL_AUTH do-until-failure
   10 clear-session
  20 class NOT_IN_CRITICAL_AUTH do-until-failure
   10 resume reauthentication
 event agent-found match-all
  10 class always do-until-failure
   10 terminate mab
   20 authenticate using dot1x retries 2 retry-time 0 priority 10
 event inactivity-timeout match-all
  10 class always do-until-failure
   10 clear-session
 event authentication-success match-all
 event violation match-all
  10 class always do-until-failure
   10 restrict
 event authorization-failure match-all
  10 class AUTHC_SUCCESS-AUTHZ_FAIL do-until-failure
   10 authentication-restart 60
!
policy-map type control subscriber PMAP_DefaultWiredDot1xLowImpactAuth_MAB_1X
 event session-started match-all
  10 class always do-until-failure
   10 authenticate using mab priority 20
 event authentication-failure match-first
  5 class DOT1X_FAILED do-until-failure
   10 terminate dot1x
   20 authentication-restart 60
  10 class AAA_SVR_DOWN_UNAUTHD_HOST do-until-failure
   10 activate service-template DefaultCriticalAuthVlan_SRV_TEMPLATE
   20 activate service-template DefaultCriticalVoice_SRV_TEMPLATE
   25 activate service-template DefaultCriticalAccess_SRV_TEMPLATE
   30 authorize
   40 pause reauthentication
  20 class AAA_SVR_DOWN_AUTHD_HOST do-until-failure
```

```
   10 pause reauthentication
   20 authorize
  30 class MAB_FAILED do-until-failure
   10 terminate mab
   20 authenticate using dot1x retries 2 retry-time 0 priority 10
  40 class DOT1X_NO_RESP do-until-failure
   10 terminate dot1x
   20 authentication-restart 60
  50 class DOT1X_TIMEOUT do-until-failure
   10 terminate dot1x
   20 authenticate using mab priority 20
  60 class always do-until-failure
   10 terminate mab
   20 terminate dot1x
   30 authentication-restart 60
 event aaa-available match-all
  10 class IN_CRITICAL_AUTH do-until-failure
   10 clear-session
  20 class NOT_IN_CRITICAL_AUTH do-until-failure
   10 resume reauthentication
 event agent-found match-all
  10 class always do-until-failure
   10 terminate mab
   20 authenticate using dot1x retries 2 retry-time 0 priority 10
 event inactivity-timeout match-all
  10 class always do-until-failure
   10 clear-session
 event authentication-success match-all
 event violation match-all
  10 class always do-until-failure
   10 restrict
 event authorization-failure match-all
  10 class AUTHC_SUCCESS-AUTHZ_FAIL do-until-failure
   10 authentication-restart 60
!
policy-map type control subscriber PMAP_DefaultWiredDot1xOpenAuth_1X_MAB
 event session-started match-all
  10 class always do-until-failure
   10 authenticate using dot1x retries 2 retry-time 0 priority 10
 event authentication-failure match-first
  5 class DOT1X_FAILED do-until-failure
   10 terminate dot1x
   20 authenticate using mab priority 20
  10 class AAA_SVR_DOWN_UNAUTHD_HOST do-until-failure
   10 activate service-template DefaultCriticalAuthVlan_SRV_TEMPLATE
   20 activate service-template DefaultCriticalVoice_SRV_TEMPLATE
   30 authorize
   40 pause reauthentication
  20 class AAA_SVR_DOWN_AUTHD_HOST do-until-failure
   10 pause reauthentication
   20 authorize
  30 class DOT1X_NO_RESP do-until-failure
   10 terminate dot1x
   20 authenticate using mab priority 20
  40 class MAB_FAILED do-until-failure
   10 terminate mab
   20 authentication-restart 60
  50 class DOT1X_TIMEOUT do-until-failure
   10 terminate dot1x
   20 authenticate using mab priority 20
  60 class always do-until-failure
   10 terminate dot1x
   20 terminate mab
   30 authentication-restart 60
 event aaa-available match-all
  10 class IN_CRITICAL_AUTH do-until-failure
   10 clear-session
  20 class NOT_IN_CRITICAL_AUTH do-until-failure
   10 resume reauthentication
 event agent-found match-all
  10 class always do-until-failure
   10 terminate mab
   20 authenticate using dot1x retries 2 retry-time 0 priority 10
 event inactivity-timeout match-all
  10 class always do-until-failure
   10 clear-session
 event authentication-success match-all
 event violation match-all
  10 class always do-until-failure
   10 restrict
 event authorization-failure match-all
  10 class AUTHC_SUCCESS-AUTHZ_FAIL do-until-failure
   10 authentication-restart 60
!
policy-map type control subscriber PMAP_DefaultWiredDot1xOpenAuth_MAB_1X
 event session-started match-all
  10 class always do-until-failure
   10 authenticate using mab priority 20
 event authentication-failure match-first
  5 class DOT1X_FAILED do-until-failure
   10 terminate dot1x
   20 authentication-restart 60
  10 class AAA_SVR_DOWN_UNAUTHD_HOST do-until-failure
   10 activate service-template DefaultCriticalAuthVlan_SRV_TEMPLATE
   20 activate service-template DefaultCriticalVoice_SRV_TEMPLATE
   30 authorize
   40 pause reauthentication
  20 class AAA_SVR_DOWN_AUTHD_HOST do-until-failure
   10 pause reauthentication
   20 authorize
  30 class MAB_FAILED do-until-failure
   10 terminate mab
   20 authenticate using dot1x retries 2 retry-time 0 priority 10
  40 class DOT1X_NO_RESP do-until-failure
   10 terminate dot1x
   20 authentication-restart 60
  50 class DOT1X_TIMEOUT do-until-failure
   10 terminate dot1x
   20 authenticate using mab priority 20
  60 class always do-until-failure
   10 terminate mab
   20 terminate dot1x
   30 authentication-restart 60
 event aaa-available match-all
  10 class IN_CRITICAL_AUTH do-until-failure
   10 clear-session
  20 class NOT_IN_CRITICAL_AUTH do-until-failure
   10 resume reauthentication
 event agent-found match-all
  10 class always do-until-failure
   10 terminate mab
   20 authenticate using dot1x retries 2 retry-time 0 priority 10
 event inactivity-timeout match-all
  10 class always do-until-failure
   10 clear-session
 event authentication-success match-all
 event violation match-all
  10 class always do-until-failure
```

```
   10 restrict
 event authorization-failure match-all
  10 class AUTHC_SUCCESS-AUTHZ_FAIL do-until-failure
   10 authentication-restart 60
!
!
template DefaultWiredDot1xClosedAuth
 dot1x pae authenticator
 dot1x timeout supp-timeout 7
 dot1x max-req 3
 switchport mode access
 switchport voice vlan 2046
 mab
 access-session closed
 access-session port-control auto
 authentication periodic
 authentication timer reauthenticate server
 service-policy type control subscriber PMAP_DefaultWiredDot1xClosedAuth_1X_MAB
!
template DefaultWiredDot1xLowImpactAuth
 dot1x pae authenticator
 dot1x timeout supp-timeout 7
 dot1x max-req 3
 switchport mode access
 switchport voice vlan 2046
 mab
 access-session port-control auto
 authentication periodic
 authentication timer reauthenticate server
 service-policy type control subscriber PMAP_DefaultWiredDot1xLowImpactAuth_1X_MAB
!
template DefaultWiredDot1xOpenAuth
 dot1x pae authenticator
 dot1x timeout supp-timeout 7
 dot1x max-req 3
 switchport mode access
 switchport voice vlan 2046
 mab
 access-session port-control auto
 authentication periodic
 authentication timer reauthenticate server
 service-policy type control subscriber PMAP_DefaultWiredDot1xOpenAuth_1X_MAB
!
!
ip access-list extended ACL_WEBAUTH_REDIRECT
 260 deny ip any host 172.16.2.1
 500 permit tcp any any eq www
 600 permit tcp any any eq 443
 700 permit tcp any any eq 8443
 800 deny udp any any eq domain
 900 deny udp any eq bootpc any eq bootps
ip access-list extended IPV4_CRITICAL_AUTH_ACL
 10 permit ip any any
ip access-list extended IPV4_PRE_AUTH_ACL
 10 permit udp any any eq bootps
 20 permit udp any any eq bootpc
 30 permit udp any any eq domain
 40 deny ip any any
!
!
ipv6 access-list IPV6_CRITICAL_AUTH_ACL
 sequence 10 permit ipv6 any any
!
ipv6 access-list IPV6_PRE_AUTH_ACL
```

```
 sequence 10 permit udp any any eq bootps
 sequence 20 permit udp any any eq bootpc
 sequence 30 permit udp any any eq domain
 sequence 40 deny ipv6 any any
!
Host onboarding interfaces
interface GigabitEthernet1/0/10
 switchport access vlan 50
 switchport mode access
 switchport voice vlan 51
 device-tracking attach-policy IPDT_POLICY
 load-interval 30
 access-session inherit disable interface-template-sticky
 access-session inherit disable autoconf
 dot1x timeout tx-period 7
 dot1x max-reauth-req 3
 no macro auto processing
 source template DefaultWiredDot1xClosedAuth
 spanning-tree portfast
 spanning-tree bpduguard enable
!
interface GigabitEthernet1/0/11
 switchport access vlan 50
 switchport mode access
 switchport voice vlan 51
 device-tracking attach-policy IPDT_POLICY
 load-interval 30
 access-session inherit disable interface-template-sticky
 access-session inherit disable autoconf
 dot1x timeout tx-period 7
 dot1x max-reauth-req 3
 no macro auto processing
 source template DefaultWiredDot1xOpenAuth
 spanning-tree portfast
 spanning-tree bpduguard enable
!
interface GigabitEthernet1/0/12
 switchport access vlan 50
 switchport mode access
 switchport voice vlan 51
 device-tracking attach-policy IPDT_POLICY
 ip access-group IPV4_PRE_AUTH_ACL in
 load-interval 30
 ipv6 traffic-filter IPV6_PRE_AUTH_ACL in
 access-session inherit disable interface-template-sticky
 access-session inherit disable autoconf
 dot1x timeout tx-period 7
 dot1x max-reauth-req 3
 no macro auto processing
 source template DefaultWiredDot1xLowImpactAuth
 spanning-tree portfast
 spanning-tree bpduguard enable
!
interface GigabitEthernet1/0/13
 switchport access vlan 50
 switchport mode access
 switchport voice vlan 51
 device-tracking attach-policy IPDT_POLICY
 load-interval 30
 access-session inherit disable interface-template-sticky
 access-session inherit disable autoconf
 cts manual
  policy static sgt 15
  no propagate sgt
 no macro auto processing
 spanning-tree portfast
 spanning-tree bpduguard enable
!
interface GigabitEthernet1/0/14
 device-tracking attach-policy IPDT_POLICY
!
```



CHAPTER 12

Configuring Group-based Policy on a Fabric Edge

Provisioning a group-based policy secures your network by providing group-based access control and secure communication between the devices in the network. For information, see Cisco TrustSec Switch Configuration Guide.

Enabling SGACL Policy Enforcement

To enable SGACL policy enforcement, perform this task:

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	**cts authorization list** *server-list* **Example:** `Device(config)# cts authorization list cts-list`	Configures a AAA server to be used by the seed device.

Step 4	**cts role-based sgt-map vlan-list** *vlan-id* **sgt** *sgt-number*
	Example:
	`Device(config)# cts role-based sgt-map vlan-list 50 sgt 4`
	`Device(config)# cts role-based sgt-map vlan-list 30 sgt 8`
		Binds an SGT with a specified VLAN or a set of VLANs.

	Command or Action	Purpose
	`Device(config)# cts role-based sgt-map vlan-list 51 sgt 15`	
Step 5	**cts role-based enforcement** **Example:** `Device(config)# cts role-based enforcement`	Enables security group access control list (SGACL) policy enforcement on routed interfaces.
Step 6	**cts role-based enforcement vlan-list** *vlan-list* **Example:** `Device(config)# cts role-based enforcement vlan-list 30,40,50-51,91`	Enables SGACL policy enforcement on the VLAN or VLAN list.

Step 7	**end**
	Example:
	`Device(config)# end`
		Returns to privileged EXEC mode.

Configuration Example for Group-based Policy on Fabric Edge

This sample configuration shows how to manually map an SGT to VLANs and enforce the SGACL policy on the VLANs.

```
CTS role-based enforcement

cts authorization list cts-list
cts role-based sgt-map vlan-list 50 sgt 4
cts role-based sgt-map vlan-list 30 sgt 8
cts role-based sgt-map vlan-list 51 sgt 15
cts role-based enforcement
cts role-based enforcement vlan-list 30,40,50-51,91
```

PART V

Feature History for LISP VXLAN Fabric

CHAPTER 13

Feature History for LISP VXLAN Fabric

Feature History for LISP VXLAN Fabric

This table provides release and related information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Cupertino 17.9.3	LISP VXLAN Fabric for a Wired Network	A LISP VXLAN fabric is an enterprise solution that enables policy-based segmentation over a LISP-based fabric overlay across a Campus and Branch network. It uses a LISP-based control plane and VXLAN-based data plane. In this release, a LISP VXLAN-based fabric supports macro segmentation and micro segmentation, Layer 3 handoffs, Layer 2 BUM traffic, overlay multicast (both Headend Replication and Native Multicast), and access-side security. Access-side security includes port-based IEEE 802.1X, DHCP Snooping, Device Tracking, and so on. Optionally, Cisco Identity Services Engine can be integrated for security policy enforcement.
Cisco IOS XE Cupertino 17.9.4	Wireless Support in a LISP VXLAN Fabric	LISP VXLAN Fabric supports Over-the-Top Centralized Wireless and Fabric-enabled Wireless.

Use the Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to Cisco Feature Navigator.