IP Routing: LISP Configuration Guide, Cisco IOS Release 15SY

First Published: April 26, 2013

Last Modified: April 26, 2013

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

Locator ID Separation Protocol (LISP) Overview

Locator ID Separation Protocol (LISP) is a network architecture and protocol that implements the use of two namespaces instead of a single IP address:

- Endpoint identifiers (EIDs)—assigned to end hosts.
- Routing locators (RLOCs)—assigned to devices (primarily routers) that make up the global routing system.

Splitting EID and RLOC functions yields several advantages including improved routing system scalability, and improved multihoming efficiency and ingress traffic engineering.

LISP functionality requires LISP-specific configuration of one or more LISP-related devices, such as the LISP egress tunnel router (ETR), ingress tunnel router (ITR), proxy ETR (PETR), proxy ITR (PITR), map resolver (MR), map server (MS), and LISP alternative logical topology (ALT) device.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

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

Prerequisites for Configuring LISP

Before you can configure Locator/ID Separation Protocol (LISP), you will need to determine the type of LISP deployment you intend to deploy. The LISP deployment defines the necessary functionality of LISP devices, which, in turn, determines the hardware, software, and additional support from LISP mapping services and proxy services that are required to complete the deployment.

LISP configuration requires the datak9 license.

Restrictions for Configuring LISP

LISP is not supported on Tunnels.

Information About Configuring LISP

LISP Functionality Overview

Problem

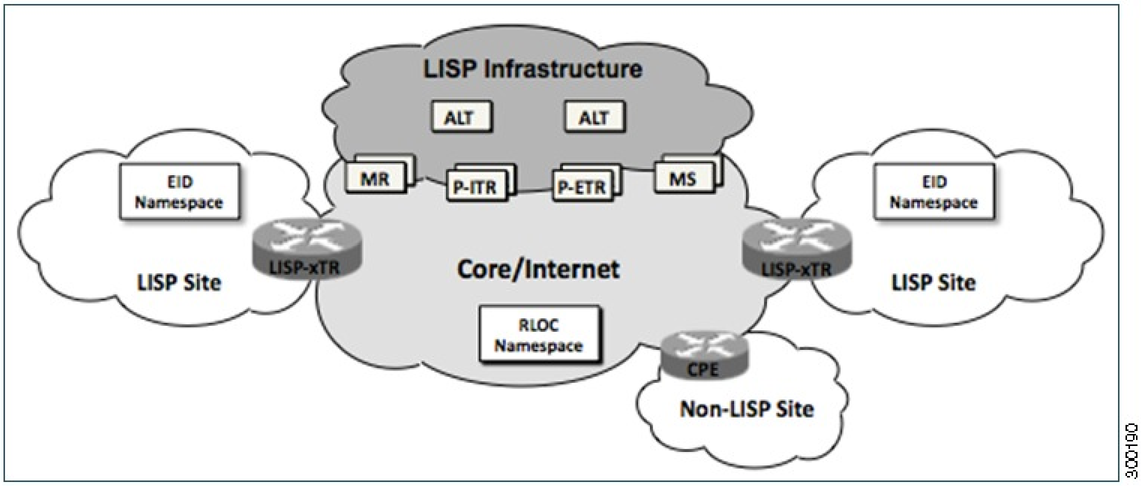
The continuous growth of the Internet presents a number of challenges. Among the most fundamental of these challenges is ensuring that the routing and addressing system continues to function efficiently even as the number of connected devices continues to increase. A basic observation during early network research and development work was that the single IP address, which includes both identity and location, leads to suboptimal route scaling and hinders multihoming and device mobility.

Solution

Locator ID Separation Protocol (LISP) provides improved routing scalability and facilitates flexible address assignment for multi-homing, provider independence, mobility, and virtualization. LISP offers an alternative to traditional Internet architecture by introducing two separate IP addresses: one to indicate routing locators (RLOCs) for routing traffic through the global Internet and a second address for endpoint identifiers (EIDs) used to identify network sessions between devices.

The figure below displays a general overview illustration of a LISP deployment environment, including the three essential environments that exist in a LISP environment: LISP sites (EID namespace), non-LISP sites (RLOC namespace), and LISP mapping service (infrastructure).

Figure 1: LISP Deployment Environment



As illustrated in the figure, the LISP EID namespace represents customer end sites in the same way that end sites are defined in non-LISP environments with one difference: The IP addresses used within these LISP sites are not advertised within the non-LISP Internet (RLOC namespace). Instead, end-customer LISP functionality is deployed exclusively on customer endpoint routers, which perform both the egress tunnel router (ETR) and ingress tunnel router (ITR) functions of a LISP device (abbreviated as xTR in the figure).

To fully implement LISP with support for mapping services and Internet interworking may require additional LISP infrastructure components as part of the deployment. As displayed in the figure above, these additional LISP infrastructure components include devices that function in the LISP roles of map resolver (MR), map server (MS), proxy egress tunnel router (PETR), proxy ingress tunnel router (PITR), and LISP alternative logical topology (ALT) device.

LISP Network Element Functions

The LISP architecture defines seven LISP-specific network infrastructure components. In some cases, a single physical device can implement more than one of these logical components. For more information, refer to the descriptions of the LISP components described in the following sections:

LISP Alternative Logical Topology

An alternative logical topology (ALT) device (not present in all mapping database deployments) connects through generic routing encapsulation (GRE) tunnels and border gateway protocol (BGP) sessions, map resolvers, map servers, and other ALT routers. The only purpose of ALT routers is to accept EID (Endpoint IDentifier) prefixes advertised by devices that form a hierarchically distinct part of the EID numbering space and then advertise an aggregated EID prefix that represents that distinct space to other parts of the ALT. Just as in the global Internet routing system, this aggregation is performed to reduce the number of prefixes that need to be propagated throughout the entire network. An MS or combined MR/MS may also be configured to perform the functions of an ALT router.

LISP Egress Tunnel Router

An ETR connects a site to the LISP-capable part of a core network (such as the Internet), publishes EID-to-RLOC mappings for the site, responds to Map-Request messages, and decapsulates and delivers LISP-encapsulated user data to end systems at the site. During operation, an ETR sends periodic Map-Register messages to all its configured map servers. The Map-Register messages contain all the EID-to-RLOC entries for the EID-numbered networks that are connected to the ETR's site.

An ETR that receives a Map-Request message verifies that the request matches an EID for which it is authoritative, constructs an appropriate Map-Reply message containing its configured mapping information, and sends this message to the ingress tunnel router (ITR) whose RLOCs are listed in the Map-Request message. An ETR that receives a LISP-encapsulated packet that is directed to one of its RLOCs decapsulates the packet, verifies that the inner header is destined for an EID-numbered end system at its site, and then forwards the packet to the end system using site-internal routing.

The ETR function is usually implemented in the customer premises equipment (CPE) router and does not require hardware changes on software-switched platforms, such as a Cisco Integrated Services Router (ISR). The same CPE router will often provide both ITR and ETR functions and, when doing so, is referred to as an xTR.

LISP Ingress Tunnel Router (ITR)

An ITR is responsible for finding EID-to-RLOC mappings for all traffic destined for LISP-capable sites. When the ITR receives a packet destined for an EID, it first looks for the EID in its mapping cache. If the ITR finds a match, it encapsulates the packet inside a LISP header with one of its RLOCs as the IP source address and one of the RLOCs from the mapping cache entry as the IP destination. The ITR then routes the packet normally.

If no entry is found in the ITR's mapping cache, the ITR sends a Map-Request message to one of its configured map resolvers and then discards the original packet. When the ITR receives a response to its Map-Request message, it creates a new mapping cache entry with the contents of the Map-Reply message. When another packet, such as a retransmission for the original and, now, discarded packet arrives, the new mapping cache entry is used for encapsulation and forwarding.

Note Sometimes the Map-Reply message will indicate that the destination is not an EID. When this happens, a negative mapping cache entry is created, which causes packets to either be discarded or forwarded natively when the packets match that cache entry.

Like the ETR, an ITR is usually implemented in a LISP site's customer premises equipment (CPE) router, which is typically configured as an xTR (performs functions of both ETR and ITR components).

LISP Map Resolver

Like an MS, a LISP MR connects to the ALT. The function of the LISP MR is to accept encapsulated Map-Request messages from ingress tunnel routers (ITRs), decapsulate those messages, and then forward the messages to the MS responsible for the egress tunnel routers (ETRs) that are authoritative for the requested EIDs.

When an MR is implemented concurrently with an MS in a private mapping system deployment, the concurrent MS forwards the encapsulated Map-Request messages to the authoritative ETRs. When a LISP ALT is present in the deployment, the MR forwards the Map-Request messages directly over the ALT to the MS responsible for the ETRs that are authoritative for the requested EIDs. An MR also sends Negative Map-Replies to ITRs in response to queries for non-LISP addresses.

LISP Map Server

An MS implements part of the distributed LISP mapping database by accepting registration requests from its client egress tunnel routers (ETRs), aggregating the successfully registered EID prefixes of those ETRs, and advertising the aggregated prefixes into the alternative logical topology (ALT) with border gateway protocol (BGP).

In a small private mapping system deployment, an MS may be configured to stand alone (or there may be several MSs) with all ETRs configured to register to each MS. If more than one, all MSs have full knowledge of the mapping system in a private deployment.

In a larger or public mapping system deployment, an MS is configured with a partial mesh of generic routing encapsulation (GRE) tunnels and BGP sessions to other map server systems or ALT routers. For these deployments, ETRs need to register to only one MS (or a few if redundancy is desired) and an ALT device is used to ensure that the entire LISP mapping system is available to all MS and MR devices.

Because an MS does not forward user data traffic—it handles only LISP control plane traffic—it does not require high performance switching capability and is well suited for implementation on a general purpose router, such as a Cisco Integrated Services Router (ISR). Both MS and MR functions are typically implemented on the same device, which is referred to as an MR/MS device.

LISP Proxy ETR

A LISP PETR implements ETR functions on behalf of non-LISP sites. A PETR is typically used when a LISP site needs to send traffic to non-LISP sites but the LISP site is connected through an access network of a service provider that does not accept nonroutable EIDs as packet sources.

When dual-stacked, a PETR may also serve as a way for EIDs and RLOCs to communicate in a LISP site that contains EIDs in one address family and RLOCs in a different address family. A dual-stacked PETR also provides multiaddress family support for LISP EIDs within one address family to be able to communicate with non-LISP destinations in the same address family over a core network within a different address family.

Example

A LISP site with IPv4-only RLOC connectivity can send IPv6 EIDs within an IPv4 LISP header across the IPv4 Internet to a dual-stacked PETR where the packets are decapsulated and then forwarded natively to non-LISP IPv6 Internet sites.

The PETR function is commonly configured on a device that also functions as a PITR. A device that functions as both a PETR and a PITR is known as a PxTR. Additionally, a PETR carries LISP data plane traffic and can be a high packet-rate device. To take advantage of this high packet-rate capability, deployments typically include hardware-switched platforms or high-end Cisco Integrated Services Routers (ISRs).

LISP Proxy ITR

A LISP PITR implements ITR mapping database lookups and LISP encapsulation functions on behalf of non-LISP-capable sites. PITRs are typically deployed near major Internet exchange points (IXPs) or in ISP networks to allow non-LISP customers from those networks to connect to LISP sites. In addition to implementing ITR functionality, a PITR also advertises some or all of the non-routable EID prefix space to the part of the non-LISP-capable Internet that it serves so that the non-LISP sites will route traffic toward the PITR for encapsulation and forwarding to LISP sites.

Note PITR advertising of nonroutable EID prefix space is intended to be highly aggregated with many EID prefixes represented by each prefix that is advertised by a PITR.

Like the PETR, when dual-stacked, the PITR also provides multiple-address family support. But the PITR supports transport of non-LISP traffic from one address family to LISP sites in the same address family over a core network within a different address family.

Example

A LISP site with IPv4-only RLOC connectivity can take advantage of a dual-stacked PITR to allow non-LISP IPv6 Internet users to reach IPv6 EIDs across the IPv4 Internet.

The PITR function is commonly configured on a device that also functions as a PETR. A device that functions as both a PETR and a PITR is known as a PxTR. Additionally, a PITR carries LISP data plane traffic and can be a high packet-rate device. To take advantage of this high packet-rate capability, deployments typically include hardware-switched platforms or high-end Cisco® Integrated Services Routers (ISRs).

Feature Information for LISP Overview

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

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

Table 1: Feature Information for LISP Overview

Feature Name	Releases	Feature Information
LISP Overview	15.1(4)M Cisco IOS XE Release 3.3.0S	The LISP Overview feature provides a general overview of LISP and its components. The following LISP components are supported: • Egress tunnel router (ETR) • Ingress tunnel router (ITR) • LISP alternative logical topology (ALT) device • Map resolver (MR) • Map server (MS) • Proxy ETR (PETR) • Proxy ITR (PITR)

Feature Name	Releases	Feature Information
LISP, SHA-2 support for site registration	15.3(2)T Cisco IOS XE Release 3.9S	LISP can be configured to use SHA2-based HMAC algorithm for integrity-checking LISP site registration messages. Prior to this release, only SHA1-based HMAC algorithm was supported. The following commands were modified: • ipv4 etr map-server • ipv6 etr map-server

CHAPTER 2

Configuring LISP (Locator ID Separation Protocol)

This guide describes how to configure basic Locator ID Separation Protocol (LISP) functionality on all LISP-related devices, including the egress tunnel router (ETR), ingress tunnel router (ITR), proxy ETR (PETR), proxy ITR (PITR), map resolver (MR), map server (MS), and LISP-ALT device.

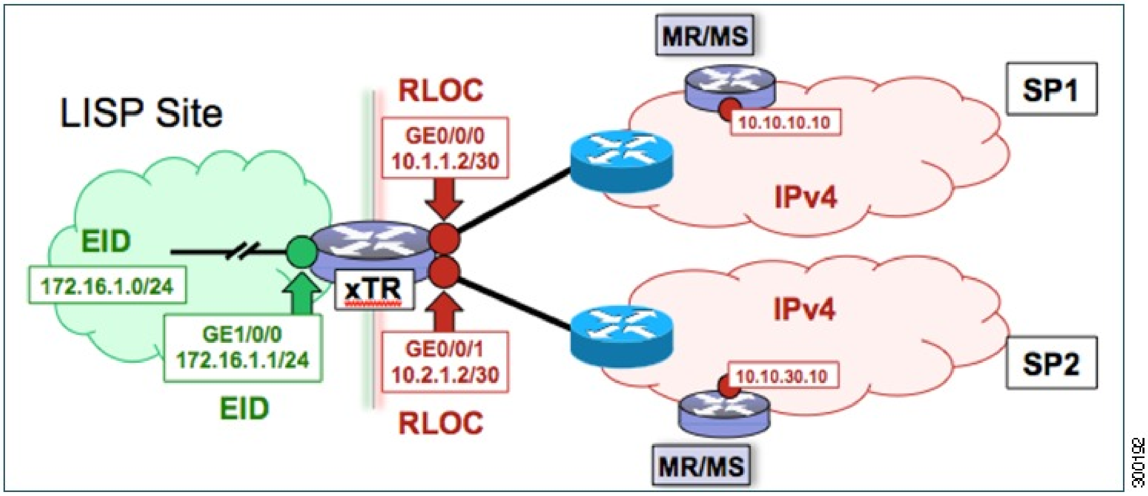
LISP is a network architecture and protocol that implements the use of two namespaces instead of a single IP address. These namespaces, known as endpoint identifiers (EIDs), are assigned to end-hosts and routing locators (RLOCs), which are assigned to devices (primarily routers) that make up the global routing system. Splitting EID and RLOC functions delivers improvements in routing system scalability, multi-homing efficiency, and ingress traffic engineering.

How to Configure LISP

Configure a Dual-Homed LISP Site with Two IPv4 RLOCs and an IPv4 EID

Perform this task to configure a dual-homed LISP site with two IPv4 RLOCs and an IPv4 EID. In this task, a LISP site uses a single edge router configured as both an ITR and an ETR (known as an xTR) with two connections to upstream providers. Both of the RLOCs and the EID prefix are IPv4. The LISP site registers to two map resolver/map server (MR/MS) devices in the network core. The topology used in this LISP configuration is shown in the figure below.

Figure 2: Dual-Homed LISP Site with Two IPv4 RLOCs and an IPv4 EID



The components illustrated in the topology shown in the figure are described below:

- **LISP site:**
  - The CPE functions as a LISP ITR and ETR (xTR).
  - The LISP xTR is authoritative for the IPv4 EID prefix of 172.16.1.0/24.
  - The LISP xTR has two RLOC connections to the core. The RLOC connection to SP1 is 10.1.1.2/30; the RLOC connection to SP2 is 10.2.1.2/30.
  - For this simple dual-homed configuration, the LISP site policy specifies equal load sharing between service provider (SP) links for ingress traffic engineering.

- **Mapping system:**
  - Two map resolver/map server (MR/MS) systems are assumed to be available for the LISP xTR to configure. The MR/MSs have IPv4 RLOCs 10.10.10.10 and 10.10.30.10.
  - Mapping Services are assumed to be provided as part of this LISP solution via a private mapping system or as a public LISP mapping system. From the perspective of the configuration of this LISP site xTR, there is no difference.

Note Map server and map resolver configurations are not shown here. See the "Configure a Private LISP Mapping System Using a Standalone Map Resolver/Map Server" section for information about map server and map resolver configuration.

This task shows how to enable and configure LISP ITR and ETR (xTR) functionality when using a LISP map server and map resolver for mapping services.

SUMMARY STEPS

1. **configure terminal**
2. **router lisp**
3. Do one of the following:
   - **database-mapping** *EID-prefix/prefix-length locator* **priority** *priority* **weight** *weight*
   - **database-mapping** *EID-prefix/prefix-length* **ipv4-interface** *locator* **priority** *priority* **weight** *weight*
4. Repeat one of the choices in Step 3 to configure a second RLOC.
5. **ipv4 itr**
6. **ipv4 etr**
7. **ipv4 itr map-resolver** *map-resolver-address*
8. **ipv4 etr map-server** *map-server-address* **key** *key-type authentication-key*
9. **exit**
10. **ip route** *ipv4-prefix next-hop*
11. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 2	**router lisp** **Example:** `Router(config)# router lisp`	Enters LISP configuration mode (software only).
Step 3	Do one of the following: • **database-mapping** *EID-prefix/prefix-length locator* **priority** *priority* **weight** *weight* • **database-mapping** *EID-prefix/prefix-length* **ipv4-interface** *locator* **priority** *priority* **weight** *weight* **Example:** `Router(config-router-lisp)# database-mapping 172.16.1.0/24 10.1.1.2 priority 1 weight 50`	Configures an EID-to-RLOC mapping relationship and its associated traffic policy for this LISP site. • In this step example, a single EID prefix, 172.16.1.0/24, is being associated with the single IPv4 RLOC 10.1.1.2 but the *weight* argument of 50 signifies that a second **database-mapping** command is to be configured in the next step. • In the second example, the configuration shows the use of the dynamic interface form of the **database-mapping** command. This form is useful when the RLOC address is obtained dynamically, such as via DHCP.

	Command or Action	Purpose
	Example: `Router(config-router-lisp)# database-mapping 172.16.1.0/24 ipv4-interface GigabitEthernet0/0/0 priority 1 weight 50`	
Step 4	Repeat one of the choices in Step 3 to configure a second RLOC.	—
Step 5	**ipv4 itr** **Example:** `Router(config-router-lisp)# ipv4 itr`	Enables LISP ITR functionality for the IPv4 address family.
Step 6	**ipv4 etr** **Example:** `Router(config-router-lisp)# ipv4 etr`	Enables LISP ETR functionality for the IPv4 address family.
Step 7	**ipv4 itr map-resolver** *map-resolver-address* **Example:** `Router(config-router-lisp)# ipv4 itr map-resolver 10.10.10.10`	Configures the locator address of the LISP map resolver to which this router will send Map-Request messages for IPv4 EID-to-RLOC mapping resolutions. • The locator address of the map resolver may be an IPv4 or IPv6 address. In this example, because each xTR has only IPv4 RLOC connectivity, the map resolver is reachable via its IPv4 locator address. (See the *LISP Command Reference* for more details.) **Note** Up to two map resolvers may be configured if multiple map resolvers are available. (See the *LISP Command Reference* for more details.)
Step 8	**ipv4 etr map-server** *map-server-address* **key** *key-type authentication-key* **Example:** `Router(config-router-lisp)# ipv4 etr map-server 10.10.10.10 key 0 some-key`	Configures the locator address of the LISP map server and the authentication key that this router, acting as an IPv4 LISP ETR, will use to register with the LISP mapping system. • The map server must be configured with EID prefixes matching those configured on this ETR and with an identical authentication key. **Note** The locator address of the map server may be an IPv4 or IPv6 address. In this example, because each xTR has only IPv4 RLOC connectivity, the map server is reachable via its IPv4 locator address. (See the *LISP Command Reference* for more details.) **Note** Up to two map servers may be configured if multiple map servers are available. (See the *LISP Command Reference* for more details.)

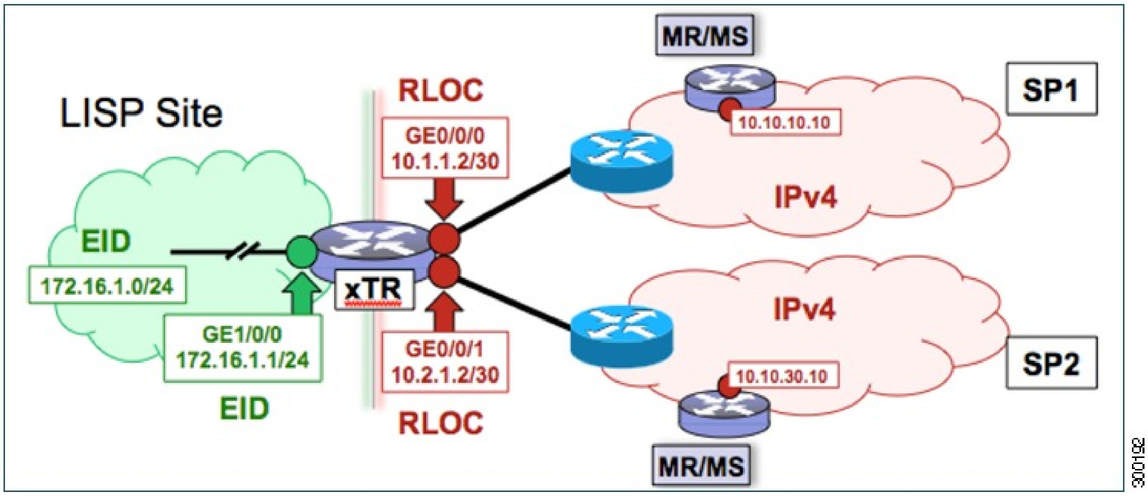
	Command or Action	Purpose
Step 9	**exit** **Example:** `Router(config-router-lisp)# exit`	Exits LISP configuration mode and returns to global configuration mode.

Step 10	**ip route** *ipv4-prefix next-hop*
	
	Example:
	
	`Router(config)# ip route 0.0.0.0 0.0.0.0 10.1.1.1`
		Configures a default route to the upstream next hop for all IPv4 destinations.
		• All IPv4 EID-sourced packets destined to both LISP and non-LISP sites are forwarded in one of two ways:
		  • LISP-encapsulated to a LISP site when traffic is LISP-to-LISP
		  • natively forwarded when traffic is LISP-to-non-LISP.
		• Packets are deemed to be a candidate for LISP encapsulation when they are sourced from a LISP EID and the destination matches one of the following entries:
		  • a current map-cache entry
		  • a default route with a legitimate next-hop
		  • no route at all
		
		In this configuration example, because the xTR has IPv4 RLOC connectivity, a default route to the upstream SP is used for all IPv4 packets to support LISP processing.

Step 11	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

Example:

Figure 3: Dual-Homed LISP Site with Two IPv4 RLOCs and an IPv4 EID



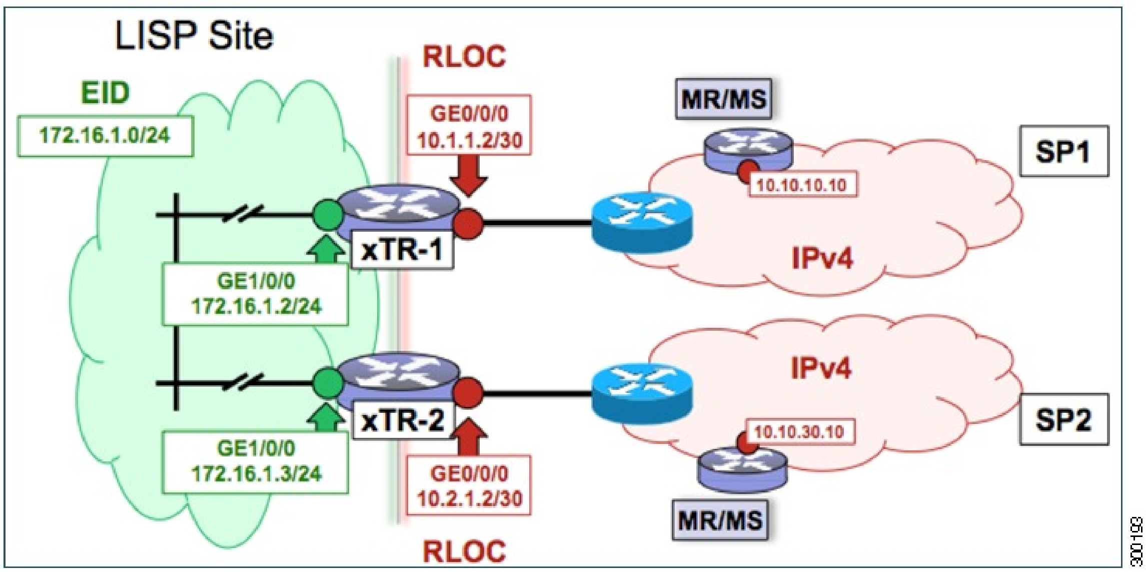
This example shows the complete configuration for the LISP topology illustrated in the figure above and in this task.

```
hostname xTR
!
no ip domain lookup
ip cef
!
interface Loopback0
 ip address 172.17.1.1 255.255.255.255
!
interface LISP0
!
interface GigabitEthernet0/0/0
 description Link to SP1 (RLOC)
 ip address 10.1.1.2 255.255.255.252
!
interface GigabitEthernet0/0/1
 description Link to SP2 (RLOC)
 ip address 10.2.1.2 255.255.255.252
!
interface GigabitEthernet1/0/0
 description Link to Site (EID)
 ip address 172.16.1.1 255.255.255.0
!
router lisp
 database-mapping 172.16.1.0/24 10.1.1.2 priority 1 weight 50
 database-mapping 172.16.1.0/24 10.2.1.2 priority 1 weight 50
 ipv4 itr
 ipv4 etr
 ipv4 itr map-resolver 10.10.10.10
 ipv4 itr map-resolver 10.10.30.10
 ipv4 etr map-server 10.10.10.10 key 0 some-key
 ipv4 etr map-server 10.10.30.10 key 0 some-key
 exit
!
ip route 0.0.0.0 0.0.0.0 10.1.1.1
ip route 0.0.0.0 0.0.0.0 10.2.1.1
```

Configure a Multihomed LISP Site with Two xTRs and Two IPv4 RLOCs and an IPv4 EID

Perform this task to configure a multihomed LISP site with two xTRs, two IPv4 RLOCs, and an IPv4 EID. In this task, a LISP site uses two edge routers. Each edge router is configured as an xTR (each performs as both an ITR and an ETR) and each also includes a single IPv4 connection to an upstream provider. (Two different providers are used in this example but the same upstream provider could be used for both connections.) Both of the RLOCs and the EID prefix are IPv4. The LISP site registers to two map resolver/map server (MR/MS) devices in the network core. The topology used in this typical multihomed LISP configuration is shown in the figure below.

Figure 4: Typical Multihomed LISP Site with Two xTRs and Two IPv4 RLOCs and an IPv4 EID



The components illustrated in the topology shown in the figure are described below:

- **LISP site:**
  - Two CPE routers make up the LISP site: xTR-1 and xTR-2.
  - Both CPE routers function as LISP xTRs (that is, an ITR and an ETR).
  - The LISP site is authoritative for the IPv4 EID prefix of 172.16.1.0/24.
  - Each LISP xTR has a single IPv4 RLOC connection to the core: the RLOC connection for xTR-1 to SP1 is 10.1.1.2/30; the RLOC connection for xTR-2 to SP2 is 10.2.1.2/30.
  - For this multihomed case, the LISP site policy specifies equal load-sharing between service provider (SP) links for ingress traffic engineering.
- **Mapping system:**

- Two map resolver/map server (MR/MS) systems are assumed to be available for the LISP xTR to configure. The MR/MSs have IPv4 RLOCs 10.10.10.10 and 10.10.30.10.
- Mapping services are assumed to be provided as part of this LISP solution via a private mapping system or as a public LISP mapping system. From the perspective of the configuration of these LISP site xTRs, there is no difference.

Note Map server and map resolver configurations are not shown here. See the "Configure a Private LISP Mapping System Using a Standalone Map Resolver/Map Server" section for information about map server and map resolver configuration.

Perform the steps in this task (once through for each xTR in the LISP site) to enable and configure LISP ITR and ETR (xTR) functionality when using a LISP map server and map resolver for mapping services. The example configurations at the end of this task show the full configuration for configuring two xTRs (xTR1 and xTR2).

SUMMARY STEPS

1. **configure terminal**
2. **router lisp**
3. **database-mapping** *EID-prefix/prefix-length locator* **priority** *priority* **weight** *weight*
4. Repeat Step 3 to configure a second RLOC for the same xTR.
5. **ipv4 itr**
6. **ipv4 etr**
7. **ipv4 itr map-resolver** *map-resolver-address*
8. Repeat Step 7 to configure a second locator address for the map resolver.
9. **ipv4 etr map-server** *map-server-address* **key** *key-type authentication-key*
10. Repeat Step 9 to configure a second locator address for the map server.
11. **exit**
12. **ip route** *ipv4-prefix next-hop*
13. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 2	**router lisp** **Example:** `Router(config)# router lisp`	Enters LISP configuration mode (software only).
Step 3	**database-mapping** *EID-prefix/prefix-length locator* **priority** *priority* **weight** *weight* **Example:** `Router(config-router-lisp)# database-mapping 172.16.1.0/24 10.1.1.2 priority 1 weight 50`	Configures an EID-to-RLOC mapping relationship and its associated traffic policy for this LISP site. • In this step example, a single EID prefix, 172.16.1.0/24, is being associated with a LISP site that contains two separate xTRs. Each xTR has a single IPv4 RLOC connection to the core. In this example, xTR-1 has an IPv4 RLOC connection to SP1 at 10.1.1.2 but the *weight* argument of 50 signifies that a second **database-mapping** command is to be configured in the next step. **Note** Two **database-mapping** commands are required on each xTR to indicate to the mapping system that this LISP site is reachable via these two IPv4 RLOCs. In this example, one RLOC is local (connected) to one xTR and the other is local (connected) to the other xTR.
Step 4	Repeat Step 3 to configure a second RLOC for the same xTR. **Example:** `Router(config-router-lisp)# database-mapping 172.16.1.0/24 10.2.1.2 priority 1 weight 50`	Configures an EID-to-RLOC mapping relationship and its associated traffic policy for an xTR on this LISP site. • In this step example, the second RLOC connection for xTR-1 has an IPv4 RLOC connection to SP2 (10.2.1.2). **Note** When a LISP site contains multiple xTRs, all xTRs must be configured with identical **database-mapping** commands to provide the mapping system with consistent information about EID-to-RLOC mappings.
Step 5	**ipv4 itr** **Example:** `Router(config-router-lisp)# ipv4 itr`	Enables LISP ITR functionality for the IPv4 address family.
Step 6	**ipv4 etr** **Example:** `Router(config-router-lisp)# ipv4 etr`	Enables LISP ETR functionality for the IPv4 address family.
Step 7	**ipv4 itr map-resolver** *map-resolver-address* **Example:** `Router(config-router-lisp)# ipv4 itr map-resolver 10.10.10.10`	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. • The locator address of the map resolver may be an IPv4 or IPv6 address. In this example, because each xTR has only IPv4 RLOC connectivity, the map resolver is reachable via its IPv4 locator address. (See the *LISP Command Reference* for more details.)

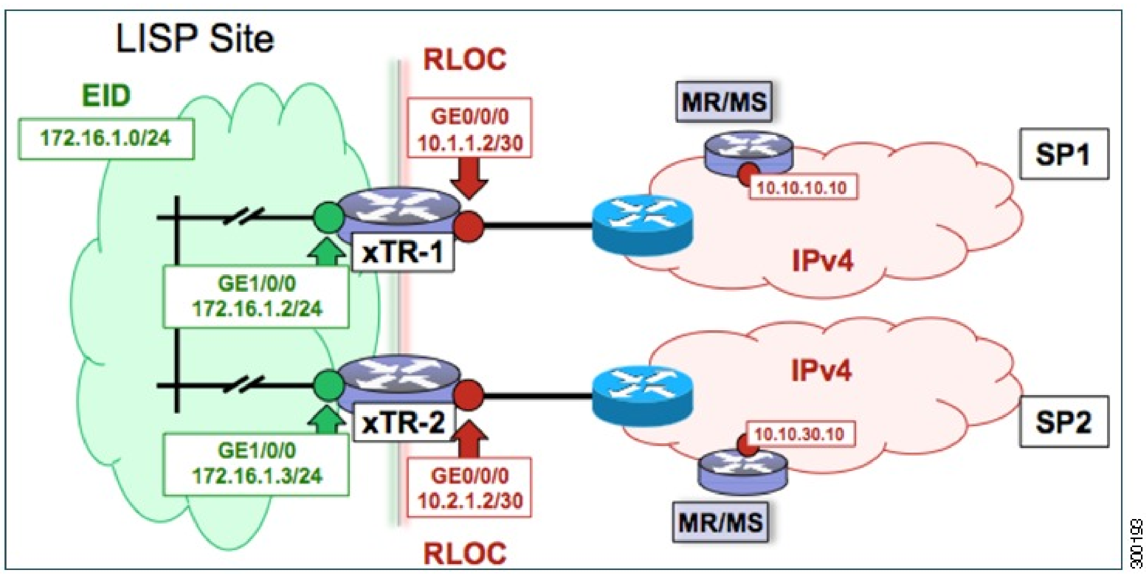
	Command or Action	Purpose
		Note Up to two map resolvers may be configured if multiple map resolvers are available. (See the *LISP Command Reference* for more details.)
Step 8	Repeat Step 7 to configure a second locator address for the map resolver. **Example:** `Router(config-router-lisp)# ipv4 itr map-resolver 10.10.30.10`	Configures a second locator address for the LISP map resolver to which this router will send Map-Request messages for IPv4 EID-to-RLOC mapping resolutions.
Step 9	**ipv4 etr map-server** *map-server-address* **key** *key-type authentication-key* **Example:** `Router(config-router-lisp)# ipv4 etr map-server 10.10.10.10 key 0 some-key`	Configures a locator address for the LISP map server and an authentication key that this router, acting as an IPv4 LISP ETR, will use to register with the LISP mapping system. • In this example, each xTR must register to both map servers. • The map server must be configured with EID prefixes matching those configured on this ETR and with an identical authentication key. **Note** The locator address of the map server may be an IPv4 or IPv6 address. In this example, because each xTR has only IPv4 RLOC connectivity, the map server is reachable via its IPv4 locator address. (See the *LISP Command Reference* for more details.) **Note** Up to two map servers may be configured if multiple map servers are available. (See the *LISP Command Reference* for more details.)
Step 10	Repeat Step 9 to configure a second locator address for the map server. **Example:** `Router(config-router-lisp)# ipv4 etr map-server 10.10.30.10 key 0 some-key`	Configures a second locator address for the LISP map server and the authentication key that this router will use to register with the LISP mapping system.
Step 11	**exit** **Example:** `Router(config-router-lisp)# exit`	Exits LISP configuration mode and returns to global configuration mode.

Step 12	**ip route** *ipv4-prefix next-hop*
	
	Example:
	
	`Router(config)# ip route 0.0.0.0 0.0.0.0 10.1.1.1`
		Configures a default route to the upstream next hop for all IPv4 destinations.
		• All IPv4 EID-sourced packets destined to both LISP and non-LISP sites are forwarded in one of two ways:
		  • LISP-encapsulated to a LISP site when traffic is LISP-to-LISP
		  • natively forwarded when traffic is LISP-to-non-LISP
		• Packets are deemed to be a candidate for LISP encapsulation when they are sourced from a LISP EID and the destination matches one of the following entries:

	Command or Action	Purpose
		• a current map-cache entry • a default route with a legitimate next-hop • no route at all In this configuration example, because the xTR has IPv4 RLOC connectivity, a default route to the upstream SP is used for all IPv4 packets to support LISP processing.
Step 13	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

Example:

Figure 5: Typical Multihomed LISP Site with Two xTRs and Two IPv4 RLOCs and an IPv4 EID



The examples below show the complete configuration for the LISP topology illustrated in the figure above and in this task:

Example configuration for xTR-1:

```
!
hostname xTR-1
!
no ip domain lookup
```

```
ip cef
!
interface Loopback0
 ip address 172.17.1.1 255.255.255.255
!
interface LISP0
!
interface GigabitEthernet0/0/0
 description Link to SP1 (RLOC)
 ip address 10.1.1.2 255.255.255.252
!
interface GigabitEthernet1/0/0
 description Link to Site (EID)
 ip address 172.16.1.2 255.255.255.0
!
router lisp
 database-mapping 172.16.1.0/24 10.1.1.2 priority 1 weight 50
 database-mapping 172.16.1.0/24 10.2.1.2 priority 1 weight 50
 ipv4 itr
 ipv4 etr
 ipv4 itr map-resolver 10.10.10.10
 ipv4 itr map-resolver 10.10.30.10
 ipv4 etr map-server 10.10.10.10 key 0 some-key
 ipv4 etr map-server 10.10.30.10 key 0 some-key
 exit
!
ip route 0.0.0.0 0.0.0.0 10.1.1.1
```

Example configuration for xTR-2:

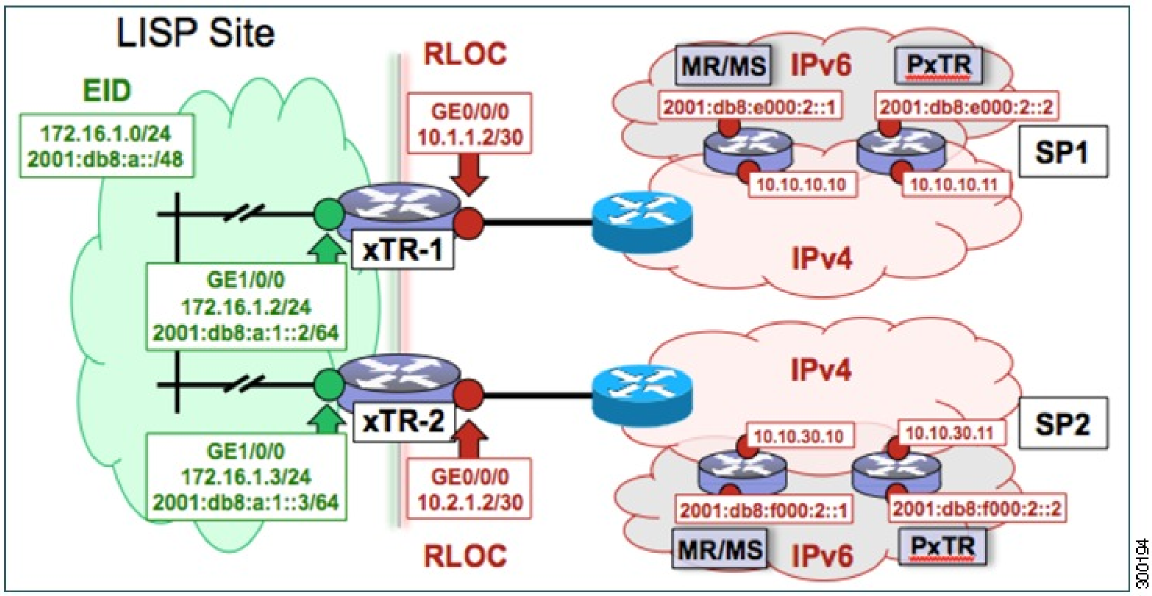
```
!
hostname xTR-2
!
no ip domain lookup
ip cef
!
interface Loopback0
 ip address 172.17.1.2 255.255.255.255
!
interface LISP0
!
interface GigabitEthernet0/0/0
 description Link to SP2 (RLOC)
 ip address 10.2.1.2 255.255.255.252
!
interface GigabitEthernet1/0/0
 description Link to Site (EID)
 ip address 172.16.1.3 255.255.255.0
!
router lisp
 database-mapping 172.16.1.0/24 10.1.1.2 priority 1 weight 50
 database-mapping 172.16.1.0/24 10.2.1.2 priority 1 weight 50
 ipv4 itr
 ipv4 etr
 ipv4 itr map-resolver 10.10.10.10
 ipv4 itr map-resolver 10.10.30.10
 ipv4 etr map-server 10.10.10.10 key 0 some-key
 ipv4 etr map-server 10.10.30.10 key 0 some-key
 exit
!
ip route 0.0.0.0 0.0.0.0 10.2.1.1
```

Configure a Multihomed LISP Site with Two xTRs and Two IPv4 RLOCs and Both an IPv4 and an IPv6 EID

Perform this task to configure a multihomed LISP site with two xTRs, two IPv4 RLOCs, and both an IPv4 and an IPv6 EID. In this task, a LISP site uses two edge routers. Each edge router is configured as an xTR (each performs as both an ITR and an ETR) and each also includes a single IPv4 connection to an upstream provider. (Two different providers are used in this example but the same upstream provider could be used for both connections.) Both of the RLOCs and one of the EIDs are IPv4. However, in this example, the LISP site includes an IPv6 EID, as well.

This LISP site requires the use of Proxy Ingress/Egress Tunnel Router (PxTR) LISP infrastructure for access to non-LISP IPv6 addresses. That is, the LISP site uses only its IPv4 RLOCs to reach IPv6 LISP and non-LISP addresses. Additionally, this LISP site registers to two map resolver/map server (MR/MS) devices in the network core. The topology used in this multihomed LISP configuration is shown in the figure below.

Figure 6: Multihomed LISP Site with Two xTRs, Two IPv4 RLOCs, and Both an IPv4 and an IPv6 EID



The components illustrated in the topology shown in the figure are described below:

- **LISP site:**
    - Two CPE routers make up the LISP site: xTR-1 and xTR-2.
    - Both CPE routers function as LISP xTRs (that is, an ITR and an ETR).
    - The LISP site is authoritative for both the IPv4 EID prefix of 172.16.1.0/24 and the IPv6 EID prefix 2001:db8:a::/48.
    - Each LISP xTR has a single RLOC connection to the core: the RLOC connection for xTR-1 to SP1 is 10.1.1.2/30; the RLOC connection for xTR-2 to SP2 is 10.2.1.2/30.
    - For this multihomed case, the LISP site policy specifies equal load-sharing between service provider (SP) links for ingress traffic engineering.

- **Mapping system:**
    - Two map resolver/map server (MR/MS) systems are assumed to be available for the LISP xTR to configure. The MR/MSs have IPv4 RLOCs 10.10.10.10 and 10.10.30.10.

- Mapping services are assumed to be provided as part of this LISP solution via a private mapping system or as a public LISP mapping system. From the perspective of the configuration of these LISP site xTRs, there is no difference.

  Note Map server and map resolver configurations are not shown here. See the "Configure a Private LISP Mapping System Using a Standalone Map Resolver/Map Server" section for information about map server and map resolver configuration.

- PxTR services are also assumed to be provided as part of this LISP solution via a private or public mapping system. From the perspective of the configuration of these LISP site xTRs, there is no difference.
- The PxTRs have IPv4 RLOCs of 10.10.10.11 and 10.10.30.11 and will be used (as PETRs) for LISP IPv6 EIDs to reach non-LISP IPv6 sites. Return traffic is attracted by the PITR function (with the assumption that the PITR advertises coarse aggregates for IPv6 LISP EIDs into the IPv6 core.)

Perform the steps in this task (once through for each xTR in the LISP site) to enable and configure LISP ITR and ETR (xTR) functionality when using a LISP map server and map resolver for mapping services. The example configurations at the end of this task show the full configuration for two xTRs (xTR1 and xTR2).

SUMMARY STEPS

1. **configure terminal**
2. **router lisp**
3. **database-mapping** *EID-prefix/prefix-length locator* **priority** *priority* **weight** *weight*
4. Repeat Step 3 to configure a second RLOC (10.2.1.2) for the same xTR and IPv4 EID prefix.
5. Repeat Step 3 and Step 4 to configure the same RLOC connections, again, for the same xTR but, when repeating these two steps, associate the IPv6 EID prefix, 2001:db8:a::/48, instead of the IPv4 EID prefix.
6. **ipv4 itr**
7. **ipv4 etr**
8. **ipv4 itr map-resolver** *map-resolver-address*
9. Repeat Step 8 to configure a second locator address of the map resolver.
10. **ipv4 etr map-server** *map-server-address* **key** *key-type authentication-key*
11. Repeat Step 10 to configure a second locator address for the map server.
12. **ipv6 itr**
13. **ipv6 etr**
14. **ipv6 itr map-resolver** *map-resolver-address*
15. Repeat Step 14 to configure a second locator address for the map resolver.
16. **ipv6 etr map-server** *map-server-address* **key** *key-type authentication-key*
17. Repeat Step 16 to configure a second locator address for the map server.
18. **ipv6 use-petr** *petr-address*
19. Repeat Step 18 to configure a second locator address for the PETR.
20. **exit**
21. **ip route** *ipv4-prefix next-hop*
22. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 2	**router lisp** **Example:** `Router(config)# router lisp`	Enters LISP configuration mode (software only).
Step 3	**database-mapping** *EID-prefix/prefix-length locator* **priority** *priority* **weight** *weight*	Configures an EID-to-RLOC mapping relationship and its associated traffic policy for this LISP site.

	Command or Action	Purpose
	Example: `Router(config-router-lisp)# database-mapping 172.16.1.0/24 10.1.1.2 priority 1 weight 50`	• In steps 3, 4, and 5 of this example, an IPv4 EID prefix, 172.16.1.0/24, and an IPv6 prefix, 2001:db8:a::/48, are being associated with a LISP site that contains two separate xTRs that each have a single IPv4 RLOC connection to the core. In this first step example, xTR-1 is configured with an IPv4 RLOC connection to SP1 at 10.1.1.2 but the *weight* argument of 50 signifies that a second **database-mapping** command is to be configured in the next step. **Note** Four **database-mapping** commands are required for each xTR to indicate to the mapping system that both the associated IPv4 and IPv6 EID prefixes are reachable at this LISP site via these two IPv4 RLOCs. In this example, one RLOC is local (connected) to one xTR and the other is local (connected) to the other xTR.
Step 4	Repeat Step 3 to configure a second RLOC (10.2.1.2) for the same xTR and IPv4 EID prefix. **Example:** `Router(config-router-lisp)# database-mapping 172.16.1.0/24 10.2.1.2 priority 1 weight 50`	Configures an EID-to-RLOC mapping relationship and its associated traffic policy for an xTR on this LISP site. • In this step example, the second RLOC connection for xTR-1 has an IPv4 RLOC connection to SP2 (10.2.1.2). **Note** When a LISP site contains multiple xTRs, all xTRs must be configured with identical **database-mapping** commands to provide the mapping system with consistent information about EID-to-RLOC mappings.
Step 5	Repeat Step 3 and Step 4 to configure the same RLOC connections, again, for the same xTR but, when repeating these two steps, associate the IPv6 EID prefix, 2001:db8:a::/48, instead of the IPv4 EID prefix.	—
Step 6	**ipv4 itr** **Example:** `Router(config-router-lisp)# ipv4 itr`	Enables LISP ITR functionality for the IPv4 address family.
Step 7	**ipv4 etr** **Example:** `Router(config-router-lisp)# ipv4 etr`	Enables LISP ETR functionality for the IPv4 address family.
Step 8	**ipv4 itr map-resolver** *map-resolver-address* **Example:** `Router(config-router-lisp)# ipv4 itr map-resolver 10.10.10.10`	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. • The locator address of the map resolver may be an IPv4 or IPv6 address. In this example, because each xTR has only IPv4 RLOC connectivity, the map resolver is reachable via its IPv4 locator address. (See the *LISP Command Reference* for more details.)

	Command or Action	Purpose
		Note Up to two map resolvers may be configured if multiple map resolvers are available. (See the *LISP Command Reference* for more details.)
Step 9	Repeat Step 8 to configure a second locator address of the map resolver. **Example:** `Router(config-router-lisp)# ipv4 itr map-resolver 10.10.30.10`	Configures a second locator address for the LISP map resolver to which this router will send Map-Request messages for IPv4 EID-to-RLOC mapping resolutions.
Step 10	**ipv4 etr map-server** *map-server-address* **key** *key-type authentication-key* **Example:** `Router(config-router-lisp)# ipv4 etr map-server 10.10.10.10 key 0 some-key`	Configures a locator address for the LISP map server and an authentication key that this router, acting as an IPv4 LISP ETR, will use to register with the LISP mapping system. • In this example, each xTR must register to both map servers. • The map server must be configured with EID prefixes matching those configured on this ETR and with an identical authentication key. **Note** The locator address of the map server may be an IPv4 or IPv6 address. In this example, because each xTR has only IPv4 RLOC connectivity, the map server is reachable via its IPv4 locator address. (See the *LISP Command Reference* for more details.) **Note** Up to two map servers may be configured if multiple map servers are available. (See the *LISP Command Reference* for more details.)
Step 11	Repeat Step 10 to configure a second locator address for the map server. **Example:** `Router(config-router-lisp)# ipv4 etr map-server 10.10.30.10 key 0 some-key`	Configures a second locator address for the LISP map server and the authentication key that this router will use to register with the LISP mapping system.
Step 12	**ipv6 itr** **Example:** `Router(config-router-lisp)# ipv6 itr`	Enables LISP ITR functionality for the IPv6 address family.
Step 13	**ipv6 etr** **Example:** `Router(config-router-lisp)# ipv6 etr`	Enables LISP ETR functionality for the IPv6 address family.

	Command or Action	Purpose
Step 14	**ipv6 itr map-resolver** *map-resolver-address* **Example:** `Router(config-router-lisp)# ipv6 itr map-resolver 10.10.10.10`	Configures a locator address for the LISP map resolver to which this router will send Map-Request messages for IPv6 EID-to-RLOC mapping resolutions. • The locator address of the map resolver may be an IPv4 or IPv6 address. In this example, because each xTR has only IPv4 RLOC connectivity, the map resolver is reachable via its IPv4 locator address. (See the *LISP Command Reference* for more details.) **Note** Up to two map resolvers may be configured if multiple map resolvers are available. (See the *LISP Command Reference* for more details.)
Step 15	Repeat Step 14 to configure a second locator address for the map resolver. **Example:** `Router(config-router-lisp)# ipv6 itr map-resolver 10.10.30.10`	Configures a second locator address for the LISP map resolver to which this router will send Map-Request messages for IPv4 EID-to-RLOC mapping resolutions.
Step 16	**ipv6 etr map-server** *map-server-address* **key** *key-type authentication-key* **Example:** `Router(config-router-lisp)# ipv6 etr map-server 10.10.10.10 key 0 some-key`	Configures a locator address for the LISP map server and an authentication key that this router, acting as an IPv6 LISP ETR, will use to register to the LISP mapping system. • In this example, each xTR must register to both map servers. • The map server must be configured with EID prefixes matching those configured on this ETR and with an identical authentication key. **Note** The locator address of the map server may be an IPv4 or IPv6 address. In this example, because each xTR has only IPv4 RLOC connectivity, the map server is reachable via its IPv4 locator address. (See the *LISP Command Reference* for more details.) **Note** Up to two map servers may be configured if multiple map servers are available. (See the *LISP Command Reference* for more details.)
Step 17	Repeat Step 16 to configure a second locator address for the map server. **Example:** `Router(config-router-lisp)# ipv6 itr map-server 10.10.30.10 key 0 some-key`	Configures a second locator address for the LISP map server and an authentication key that this router, acting as an IPv6 LISP ETR, will use to register with the LISP mapping system.
Step 18	**ipv6 use-petr** *petr-address* **Example:** `Router(config-router-lisp)# ipv6 use-petr 10.10.10.11`	Configures a locator address for the Proxy Egress Tunnel Router (PETR) to which each xTR will forward LISP-encapsulated IPv6 EIDs (using the xTR's IPv4 RLOC) to reach non-LISP IPv6 addresses.

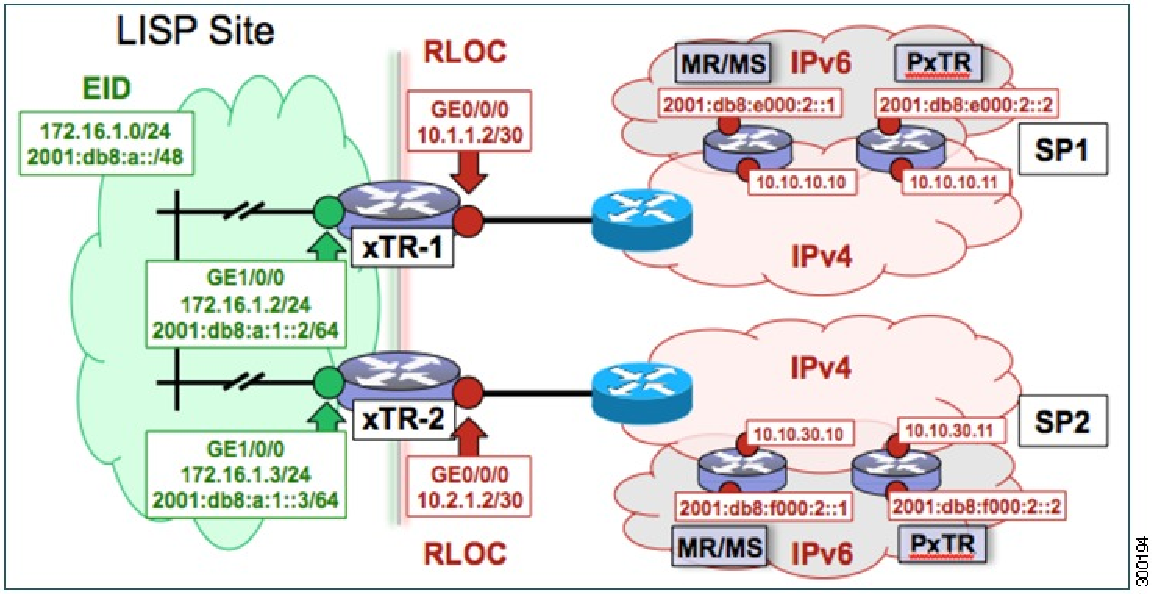
	Command or Action	Purpose
		Note The PETR is assumed to be dual-stacked and capable of natively reaching the non-LISP IPv6 address. In addition, the PITR is assumed to be dual-stacked and to be advertising coarse aggregates for IPv6 LISP EIDs into the IPv6 core to handle return traffic (non-LISP IPv6 to LISP IPv6 over an IPv4 infrastructure). **Note** The locator address of the PETR may be an IPv4 or IPv6 address. In this example, because each xTR has only IPv4 RLOC connectivity, the PETR is reachable via its IPv4 locator address. (See the *LISP Command Reference* for more details.) **Note** Up to eight PETRs may be configured if multiple PETRs are available. (See the *LISP Command Reference* for more details.)
Step 19	Repeat Step 18 to configure a second locator address for the PETR. **Example:** `Router(config-router-lisp)# ipv6 use-petr 10.10.30.11`	Configures a second locator address for the PETR to which each xTR will forward LISP-encapsulated IPv6 EIDs (using the xTR's IPv4 RLOC) to reach non-LISP IPv6 addresses.
Step 20	**exit** **Example:** `Router(config-router-lisp)# exit`	Exits LISP configuration mode and returns to global configuration mode.

Step 21	**ip route** *ipv4-prefix next-hop*
	
	Example:
	
	`Router(config)# ip route 0.0.0.0 0.0.0.0 10.1.1.1`
		Configures a default route to the upstream next hop for all IPv4 destinations.
		• All IPv4 EID-sourced packets destined to both LISP and non-LISP sites are forwarded in one of two ways:
		  • LISP-encapsulated to a LISP site when traffic is LISP-to-LISP
		  • natively forwarded when traffic is LISP-to-non-LISP
		• Packets are deemed to be a candidate for LISP encapsulation when they are sourced from a LISP EID and the destination matches one of the following entries:
		  • a current map-cache entry
		  • a default route with a legitimate next-hop
		  • no route at all
		
		In this configuration example, because the xTR has IPv4 RLOC connectivity, a default route to the upstream SP is used for all IPv4 packets to support LISP processing.

	Command or Action	Purpose
Step 22	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

Example:

Figure 7: Multihomed LISP Site with Two xTRs, Two IPv4 RLOCs, and Both an IPv4 and an IPv6 EID



The examples below show the complete configuration for the LISP topology illustrated in the figure above and in this task:

Example configuration for xTR-1:

```
!
hostname xTR-1
!
no ip domain lookup
ip cef
ipv6 unicast-routing
ipv6 cef
!
interface Loopback0
 ip address 172.17.1.1 255.255.255.255
!
interface LISP0
!
interface GigabitEthernet0/0/0
 description Link to SP1 (RLOC)
 ip address 10.1.1.2 255.255.255.252
```

```
!
interface GigabitEthernet1/0/0
 description Link to Site (EID)
 ip address 172.16.1.2 255.255.255.0
 ipv6 address 2001:db8:a:1::2/64
!
router lisp
 database-mapping 172.16.1.0/24 10.1.1.2 priority 1 weight 50
 database-mapping 172.16.1.0/24 10.2.1.2 priority 1 weight 50
 database-mapping 2001:db8:a::/48 10.1.1.2 priority 1 weight 50
 database-mapping 2001:db8:a::/48 10.2.1.2 priority 1 weight 50
 ipv4 itr
 ipv4 etr
 ipv4 itr map-resolver 10.10.10.10
 ipv4 itr map-resolver 10.10.30.10
 ipv4 etr map-server 10.10.10.10 key 0 some-key
 ipv4 etr map-server 10.10.30.10 key 0 some-key
 ipv6 itr
 ipv6 etr
 ipv6 itr map-resolver 10.10.10.10
 ipv6 itr map-resolver 10.10.30.10
 ipv6 etr map-server 10.10.10.10 key 0 some-key
 ipv6 etr map-server 10.10.30.10 key 0 some-key
 ipv6 use-petr 10.10.10.11
 ipv6 use-petr 10.10.30.11
 exit
!
ip route 0.0.0.0 0.0.0.0 10.1.1.1
!
ipv6 route ::/0
```

Example configuration for xTR-2:

```
!
no ip domain lookup
ip cef
ipv6 unicast-routing
ipv6 cef
!
interface Loopback0
 ip address 172.17.1.2 255.255.255.255
!
interface LISP0
!
interface GigabitEthernet0/0/0
 description Link to SP2 (RLOC)
 ip address 10.2.1.2 255.255.255.252
!
interface GigabitEthernet1/0/0
 description Link to Site (EID)
 ip address 172.16.1.3 255.255.255.0
 ipv6 address 2001:db8:a:1::3/64
!
router lisp
 database-mapping 172.16.1.0/24 10.1.1.2 priority 1 weight 50
 database-mapping 172.16.1.0/24 10.2.1.2 priority 1 weight 50
 database-mapping 2001:db8:a::/48 10.1.1.2 priority 1 weight 50
 database-mapping 2001:db8:a::/48 10.2.1.2 priority 1 weight 50
 ipv4 itr
 ipv4 etr
 ipv4 itr map-resolver 10.10.10.10
 ipv4 itr map-resolver 10.10.30.10
 ipv4 etr map-server 10.10.10.10 key 0 some-xtr-key
 ipv4 etr map-server 10.10.30.10 key 0 some-xtr-key
 ipv6 itr
 ipv6 etr
 ipv6 itr map-resolver 10.10.10.10
 ipv6 itr map-resolver 10.10.30.10
 ipv6 etr map-server 10.10.10.10 key 0 some-xtr-key
 ipv6 etr map-server 10.10.30.10 key 0 some-xtr-key
 ipv6 use-petr 10.10.10.11
 ipv6 use-petr 10.10.30.11
```

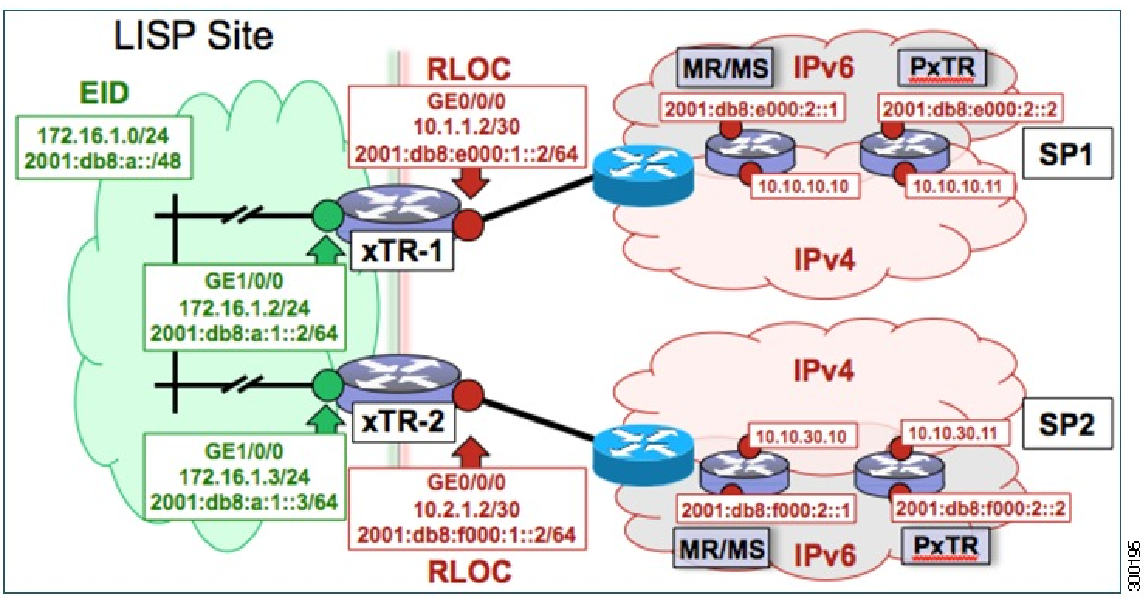
```
 exit
!
ip route 0.0.0.0 0.0.0.0 10.2.1.1
!
ipv6 route ::/0
```

Configure a Multihomed LISP Site with Two xTRs that Each have Both an IPv4 and an IPv6 RLOC and Both an IPv4 and an IPv6 EID

Perform this task to configure a multihomed LISP site with two xTRs, each with both an IPv4 and an IPv6 RLOC and both with an IPv4 and an IPv6 EID. In this task, a LISP site uses two edge routers. Each edge router is configured as an xTR (each performs as both an ITR and an ETR) and each also includes a single, dual stack (IPv4 and IPv6) connection to an upstream provider. (Two different providers are used in this example but the same upstream provider could be used for both connections.) Each xTR has an IPv4 RLOC and an IPv6 RLOC and both IPv4 and IPv6 EID prefixes are being used within the LISP site. However, because the site has both IPv4 and IPv6 RLOCs, it does not require a Proxy Ingress/Egress Tunnel Router (PxTR) LISP infrastructure for access to non-LISP IPv6 addresses. (The PxTR infrastructure can still be configured as a resiliency mechanism if desired.)

The LISP site registers to two map resolver/map server (MR/MS) devices in the network core using both IPv4 and IPv6 locators. The topology used in this multihomed LISP configuration is shown in the figure below.

Figure 8: Multihomed LISP Site with Two xTRs, Each with an IPv4 and an IPv6 RLOC and each with an IPv4 and an IPv6 EID



The components illustrated in the topology shown in the figure are described below:

- **LISP site:**
  - Two CPE routers make up the LISP site: xTR-1 and xTR-2.
  - Both CPE routers function as LISP xTRs (that is, an ITR and an ETR).

  - The LISP site is authoritative for both the IPv4 EID prefix of 172.16.1.0/24 and the IPv6 EID prefix 2001:db8:a::/48.
  - Each LISP xTR has a single IPv4 RLOC connection and a single IPv6 RLOC connection to the core: the RLOC connections for xTR-1 to SP1 include an IPv4 RLOC, 10.1.1.2/30, and an IPv6 RLOC, 2001:db8:e000:1::2/64. The xTR-2 connections to SP2 include IPv4 RLOC 10.2.1.2/30 and IPv6 RLOC 2001:db8:f000:1::2/64.
  - For this multihomed case, the LISP site policy specifies equal load-sharing between service provider (SP) links for ingress traffic engineering.

- **Mapping system:**
  - Two map resolver/map server systems are assumed to be available for the LISP xTR to configure. The MR/MSs have IPv4 RLOCs 10.10.10.10 and 10.10.30.10 and IPv6 RLOCs 2001:db8:e000:2::1 and 2001:db8:f000:2::1.
  - Mapping services are assumed to be provided as part of this LISP solution via a private mapping system or as a public LISP mapping system. From the perspective of the configuration of these LISP site xTRs, there is no difference.

Note Map resolver and map server configurations are not shown here. See the "Configure a Private LISP Mapping System Using a Standalone Map Resolver/Map Server" section for information about map resolver and map server configuration.

  - PxTR services are not required in this example since both xTRs have dual-stack connectivity to the core.

Perform the steps in this task (once through for each xTR in the LISP site) to enable and configure LISP ITR and ETR (xTR) functionality when using a LISP map resolver and map server for mapping services. The example configurations at the end of this task show the full configuration for two xTRs (xTR1 and xTR2).

SUMMARY STEPS

1. **configure terminal**
2. **router lisp**
3. **database-mapping** *EID-prefix/prefix-length locator* **priority** *priority* **weight** *weight*
4. Repeat Step 3 to configure a second IPv4 RLOC for the same xTR and IPv4 EID prefix.
5. Repeat Step 3 and Step 4 to configure the same RLOC connections, again, for the same xTR but, when repeating these two steps, associate the IPv6 EID prefix, 2001:db8:a::/48, instead of the IPv4 EID prefix.
6. Repeat Step 3, Step 4, and Step 5 to configure the second set of IPv4 and IPv6 RLOC connections on the same xTR for both the IPv4 and IPv6 EID prefixes.
7. **ipv4 itr**
8. **ipv4 etr**
9. **ipv4 itr map-resolver** *map-resolver-address*
10. Repeat Step 9 to configure a second locator address of the LISP map resolver.
11. Repeat Step 9 and Step 10 to configure the IPv6 locator addresses of the LISP two map resolvers.
12. **ipv4 etr map-server** *map-server-address* **key** *key-type authentication-key*
13. Repeat Step 12 to configure a second locator address of the map server.
14. Repeat Step 12 and Step 13 to configure the IPv6 locator addresses of the two map servers.
15. **ipv6 itr**
16. **ipv6 etr**
17. **ipv6 itr map-resolver** *map-resolver-address*
18. Repeat Step 17 to configure a second IPv6 locator address of the LISP map resolver.
19. Repeat Step 17 and Step18 to configure the IPv6 (instead of IPv4) locator addresses for the two map resolvers to which this router will send Map-Request messages for IPv6 EID-to-RLOC mapping resolutions.
20. **ipv6 etr map-server** *map-server-address* **key** *key-type authentication-key*
21. Repeat Step 20 to configure a second locator address of the LISP map server.
22. Repeat Steps 20 and 21 to configure the IPv6 locator addresses of the two map servers for which this router, acting as an IPv6 LISP ETR, will use to register to the LISP mapping system.
23. **exit**
24. **ip route** *ipv4-prefix next-hop*
25. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 2	**router lisp** **Example:** `Router(config)# router lisp`	Enters LISP configuration mode (software only).
Step 3	**database-mapping** *EID-prefix/prefix-length locator* **priority** *priority* **weight** *weight* **Example:** `Router(config-router-lisp)# database-mapping 172.16.1.0/24 10.1.1.2 priority 1 weight 50`	Configures an EID-to-RLOC mapping relationship and its associated traffic policy for this LISP site. • In this example, a single IPv4 EID prefix, 172.16.1.0/24, and a single IPv6 prefix, 2001:db8:a::/48, are being associated with a LISP site that contains two separate xTRs that each have a single IPv4 RLOC connection and a single IPv6 connection to the core. In this first database-mapping step example, xTR-1 is configured with an IPv4 RLOC connection to SP1 (10.1.1.2) and an IPv6 RLOC connection to SP1 (2001:db8:e000:1::2/64.) while xTR-2 has an IPv4 RLOC connection of10.2.1.2 to SP2 and an IPv6 RLOC connection of 2001:db8:f000:1::2/64 to SP2. The *weight* argument of 50 signifies that a second **database-mapping** command is to be configured in the next step. **Note** Eight **database-mapping** commands are required for each xTR to indicate to the mapping system that both the IPv4 and IPv6 EID prefixes are reachable at this LISP site via both the two IPv4 RLOCs and the two IPv6 RLOCs. In this example, one IPv4 RLOC and one IPv6 RLOC are local (connected) to one xTR and the others are local (connected) to the other xTR.
Step 4	Repeat Step 3 to configure a second IPv4 RLOC for the same xTR and IPv4 EID prefix. **Example:** `Router(config-router-lisp)# database-mapping 172.16.1.0/24 10.2.1.2 priority 1 weight 50`	Configures an EID-to-RLOC mapping relationship and its associated traffic policy for an xTR on this LISP site. • In this step example, the second RLOC connection for xTR-1 has an IPv4 RLOC connection to SP2 (10.2.1.2). **Note** When a LISP site contains multiple xTRs, all xTRs must be configured with identical **database-mapping** commands to provide the mapping system with consistent information about EID-to-RLOC mappings.
Step 5	Repeat Step 3 and Step 4 to configure the same RLOC connections, again, for the same xTR but, when repeating these two steps, associate the IPv6 EID prefix, 2001:db8:a::/48, instead of the IPv4 EID prefix. **Example:** `Router(config-router-lisp)# database-mapping 2001:db8:a::/48 10.1.1.2 priority 1 weight 50`	—

	Command or Action	Purpose
	Example: `Router(config-router-lisp)# database-mapping 2001:db8:a::/48 10.2.1.2 priority 1 weight 50`	
Step 6	Repeat Step 3, Step 4, and Step 5 to configure the second set of IPv4 and IPv6 RLOC connections on the same xTR for both the IPv4 and IPv6 EID prefixes.	—
Step 7	**ipv4 itr** **Example:** `Router(config-router-lisp)# ipv4 itr`	Enables LISP ITR functionality for the IPv4 address family.
Step 8	**ipv4 etr** **Example:** `Router(config-router-lisp)# ipv4 etr`	Enables LISP ETR functionality for the IPv4 address family.
Step 9	**ipv4 itr map-resolver** *map-resolver-address* **Example:** `Router(config-router-lisp)# ipv4 itr map-resolver 10.10.10.10`	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. • The locator address of the map resolver may be an IPv4 or IPv6 address. In this example, because each xTR has both IPv4 and IPv6 RLOC connectivity, the map resolver is reachable via both IPv4 and IPv6 locator addresses. (See the *LISP Command Reference* for more details.) **Note** Up to two map resolvers may be configured if multiple map resolvers are available. (See the *LISP Command Reference* for more details.)
Step 10	Repeat Step 9 to configure a second locator address of the LISP map resolver. **Example:** `Router(config-router-lisp)# ipv4 itr map-resolver 10.10.30.10`	Configures a second locator address for the LISP map resolver to which this router will send Map-Request messages for IPv4 EID-to-RLOC mapping resolutions.
Step 11	Repeat Step 9 and Step 10 to configure the IPv6 locator addresses of the LISP two map resolvers.	—

	Command or Action	Purpose
Step 12	**ipv4 etr map-server** *map-server-address* **key** *key-type authentication-key* **Example:** `Router(config-router-lisp)# ipv4 etr map-server 10.10.10.10 key 0 some-key`	Configures a locator address for the LISP map server and an authentication key that this router, acting as an IPv4 LISP ETR, will use to register with the LISP mapping system. • In this example, a second xTR can be registered to the same two map servers using the same authentication key. • The map server must be configured with EID prefixes matching those configured on this ETR and with an identical authentication key. **Note** The locator address of the map server may be an IPv4 or IPv6 address. In this example, because each xTR has both IPv4 and IPv6 RLOC connectivity, the map server is reachable via both IPv4 and IPv6 locator addresses. (See the *LISP Command Reference* for more details.) **Note** Up to two map servers may be configured if multiple map servers are available. (See the *LISP Command Reference* for more details.)
Step 13	Repeat Step 12 to configure a second locator address of the map server. **Example:** `Router(config-router-lisp)# ipv4 etr map-server 10.10.30.10 key 0 some-key`	Configures a second IPv4 locator address of the LISP map server and the authentication key that this router, acting as an IPv4 LISP ETR, will use to register with the LISP mapping system.

Step 14	Repeat Step 12 and Step 13 to configure the IPv6 locator addresses of the two map servers.
	
	Example:
	`ipv4 etr map-server 2001:db8:e000:2::1 key 0 some-xtr-key`
	
	Example:
	`ipv4 etr map-server 2001:db8:f000:2::1 key 0 some-xtr-key`
		—

Step 15	**ipv6 itr** **Example:** `Router(config-router-lisp)# ipv6 itr`	Enables LISP ITR functionality for the IPv6 address family.
Step 16	**ipv6 etr** **Example:** `Router(config-router-lisp)# ipv6 etr`	Enables LISP ETR functionality for the IPv6 address family.

	Command or Action	Purpose
Step 17	**ipv6 itr map-resolver** *map-resolver-address* **Example:** `Router(config-router-lisp)# ipv6 itr map-resolver 10.10.10.10`	Configures a locator address for the LISP map resolver to which this router will send Map-Request messages for IPv6 EID-to-RLOC mapping resolutions. • The locator address of the map resolver may be an IPv4 or IPv6 address. In this example, because each xTR has both IPv4 and IPv6 RLOC connectivity, the map resolver is reachable via both IPv4 and IPv6 locator addresses. (See the *LISP Command Reference* for more details.) **Note** Up to two map resolvers may be configured if multiple map resolvers are available. (See the *LISP Command Reference* for more details.)
Step 18	Repeat Step 17 to configure a second IPv6 locator address of the LISP map resolver. **Example:** `Router(config-router-lisp)# ipv6 itr map-resolver 10.10.30.10`	Configures a second locator address of the map resolver to which this router will send Map-Request messages for IPv6 EID-to-RLOC mapping resolutions.

Step 19	Repeat Step 17 and Step18 to configure the IPv6 (instead of IPv4) locator addresses for the two map resolvers to which this router will send Map-Request messages for IPv6 EID-to-RLOC mapping resolutions.
	
	Example:
	`ipv6 itr map-resolver 2001:db8:e000:2::1`
	
	Example:
	`ipv6 itr map-resolver 2001:db8:f000:2::1`
		—

Step 20	**ipv6 etr map-server** *map-server-address* **key** *key-type authentication-key* **Example:** `Router(config-router-lisp)# ipv6 etr map-server 10.10.10.10 key 0 some-key`	Configures a locator address for the LISP map server and an authentication key that this router, acting as an IPv6 LISP ETR, will use to register to the LISP mapping system. • In this example, a second xTR can be registered to the same two map servers using the same authentication key. • The map server must be configured with EID prefixes matching those configured on this ETR and with an identical authentication key. **Note** The locator address of the map server may be an IPv4 or IPv6 address. In this example, because each xTR has both IPv4 and IPv6 RLOC connectivity, the map server is reachable via both IPv4 and IPv6 locator addresses. (See the *LISP Command Reference* for more details.)

	Command or Action	Purpose
		Note Up to two map servers may be configured if multiple map servers are available. (See the *LISP Command Reference* for more details.)
Step 21	Repeat Step 20 to configure a second locator address of the LISP map server. **Example:** `Router(config-router-lisp)# ipv6 etr map-server 10.10.30.10 key 0 some-key`	Configures a second locator address for the LISP map server and an authentication key that this router, acting as an IPv6 LISP ETR, will use to register with the LISP mapping system.

Step 22	Repeat Steps 20 and 21 to configure the IPv6 locator addresses of the two map servers for which this router, acting as an IPv6 LISP ETR, will use to register to the LISP mapping system.
	Example:
	`ipv6 etr map-server 2001:db8:e000:2::1 key 0 some-xtr-key`
	Example:
	`ipv6 etr map-server 2001:db8:f000:2::1 key 0 some-xtr-key`
		—

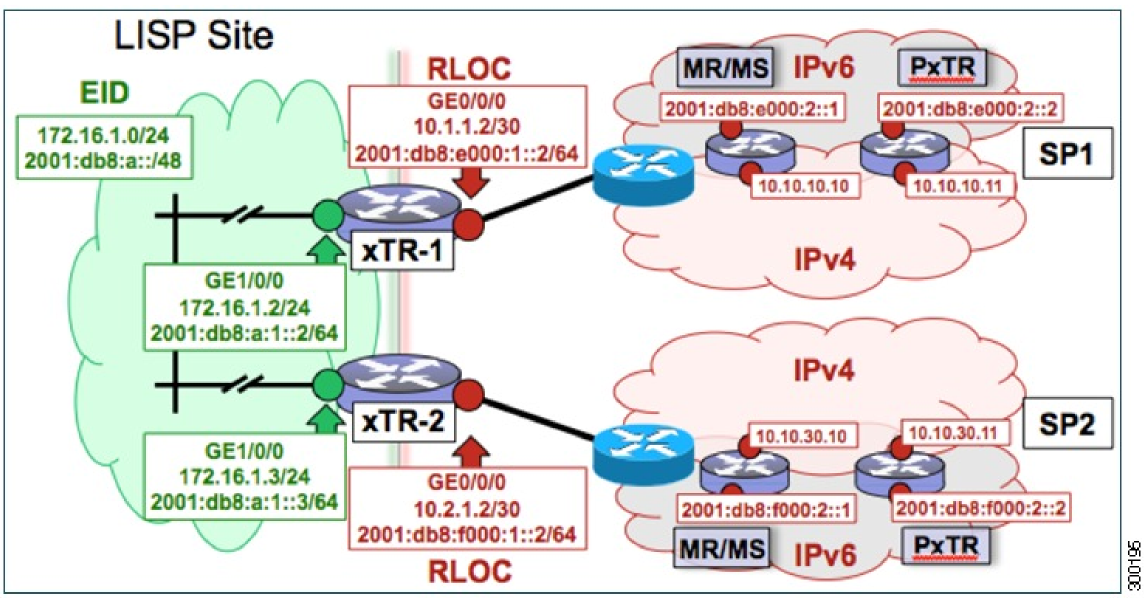
Step 23	**exit** **Example:** `Router(config-router-lisp)# exit`	Exits LISP configuration mode and returns to global configuration mode.

Step 24	**ip route** *ipv4-prefix next-hop*
	Example:
	`Router(config)# ip route 0.0.0.0 0.0.0.0 10.1.1.1`
		Configures a default route to the upstream next hop for all IPv4 destinations.
		• All IPv4 EID-sourced packets destined to both LISP and non-LISP sites are forwarded in one of two ways:
		  • LISP-encapsulated to a LISP site when traffic is LISP-to-LISP
		  • natively forwarded when traffic is LISP-to-non-LISP
		• Packets are deemed to be a candidate for LISP encapsulation when they are sourced from a LISP EID and the destination matches one of the following entries:
		  • a current map-cache entry
		  • a default route with a legitimate next-hop
		  • no route at all

	Command or Action	Purpose
		In this configuration example, because the xTR has IPv4 RLOC connectivity, a default route to the upstream SP is used for all IPv4 packets to support LISP processing.
Step 25	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

Example:

Figure 9: Multihomed LISP Site with Two xTRs, Each with an IPv4 and an IPv6 RLOC and each with an IPv4 and an IPv6 EID



The examples below show the complete configuration for the LISP topology illustrated in the figure above and in this task:

Example configuration for xTR-1:

```
!
hostname xTR-1
!
no ip domain lookup
ip cef
ipv6 unicast-routing
ipv6 cef
!
interface Loopback0
 ip address 172.17.1.1 255.255.255.255
```

```
!
interface LISP0
!
interface GigabitEthernet0/0/0
 description Link to SP1 (RLOC)
 ip address 10.1.1.2 255.255.255.252
 ipv6 address 2001:db8:e000:1::2/64
!
interface GigabitEthernet1/0/0
 description Link to Site (EID)
 ip address 172.16.1.2 255.255.255.0
 ipv6 address 2001:db8:a:1::2/64
!
router lisp
 database-mapping 172.16.1.0/24 10.1.1.2 priority 1 weight 50
 database-mapping 172.16.1.0/24 10.2.1.2 priority 1 weight 50
 database-mapping 2001:db8:a::/48 10.1.1.2 priority 1 weight 50
 database-mapping 2001:db8:a::/48 10.2.1.2 priority 1 weight 50
 database-mapping 172.16.1.0/24 2001:db8:e000:1::2 priority 1 weight 50
 database-mapping 172.16.1.0/24 2001:db8:f000:1::2 priority 1 weight 50
 database-mapping 2001:db8:a::/48 2001:db8:e000:1::2 priority 1 weight 50
 database-mapping 2001:db8:a::/48 2001:db8:f000:1::2 priority 1 weight 50
 ipv4 itr
 ipv4 etr
 ipv4 itr map-resolver 10.10.10.10
 ipv4 itr map-resolver 10.10.30.10
 ipv4 itr map-resolver 2001:db8:e000:2::1
 ipv4 itr map-resolver 2001:db8:f000:2::1
 ipv4 etr map-server 10.10.10.10 key 0 some-xtr-key
 ipv4 etr map-server 10.10.30.10 key 0 some-xtr-key
 ipv4 etr map-server 2001:db8:e000:2::1 key 0 some-xtr-key
 ipv4 etr map-server 2001:db8:f000:2::1 key 0 some-xtr-key
 ipv6 itr
 ipv6 etr
 ipv6 itr map-resolver 10.10.10.10
 ipv6 itr map-resolver 10.10.30.10
 ipv6 itr map-resolver 2001:db8:e000:2::1
 ipv6 itr map-resolver 2001:db8:f000:2::1
 ipv6 etr map-server 10.10.10.10 key 0 some-xtr-key
 ipv6 etr map-server 10.10.30.10 key 0 some-xtr-key
 ipv6 etr map-server 2001:db8:e000:2::1 key 0 some-xtr-key
 ipv6 etr map-server 2001:db8:f000:2::1 key 0 some-xtr-key
 exit
!
ip route 0.0.0.0 0.0.0.0 10.1.1.1
!
ipv6 route ::/0 2001:db8:e000:1::1
!
```

Example configuration for xTR-2:

```
!
hostname xTR-2
!
no ip domain lookup
ip cef
ipv6 unicast-routing
ipv6 cef
!
interface Loopback0
 ip address 172.17.1.2 255.255.255.255
!
interface LISP0
!
interface GigabitEthernet0/0/0
 description Link to SP2 (RLOC)
 ip address 10.2.1.2 255.255.255.252
 ipv6 address 2001:db8:f000:1::2/64
!
interface GigabitEthernet1/0/0
 description Link to Site (EID)
 ip address 172.16.1.3 255.255.255.0
 ipv6 address 2001:db8:a:1::3/64
```

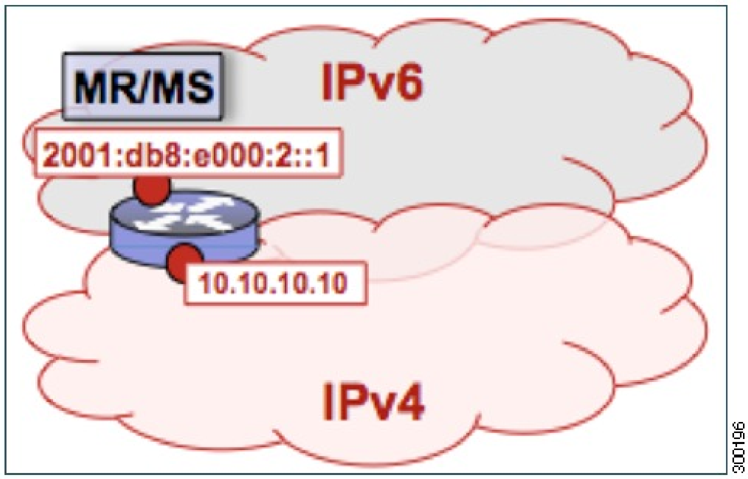
```
!
router lisp
 database-mapping 172.16.1.0/24 10.1.1.2 priority 1 weight 50
 database-mapping 172.16.1.0/24 10.2.1.2 priority 1 weight 50
 database-mapping 2001:db8:a::/48 10.1.1.2 priority 1 weight 50
 database-mapping 2001:db8:a::/48 10.2.1.2 priority 1 weight 50
 database-mapping 172.16.1.0/24 2001:db8:e000:1::2 priority 1 weight 50
 database-mapping 172.16.1.0/24 2001:db8:f000:1::2 priority 1 weight 50
 database-mapping 2001:db8:a::/48 2001:db8:e000:1::2 priority 1 weight 50
 database-mapping 2001:db8:a::/48 2001:db8:f000:1::2 priority 1 weight 50
 ipv4 itr
 ipv4 etr
 ipv4 itr map-resolver 10.10.10.10
 ipv4 itr map-resolver 10.10.30.10
 ipv4 itr map-resolver 2001:db8:e000:2::1
 ipv4 itr map-resolver 2001:db8:f000:2::1
 ipv4 etr map-server 10.10.10.10 key 0 some-xtr-key
 ipv4 etr map-server 10.10.30.10 key 0 some-xtr-key
 ipv4 etr map-server 2001:db8:e000:2::1 key 0 some-xtr-key
 ipv4 etr map-server 2001:db8:f000:2::1 key 0 some-xtr-key
 ipv6 itr
 ipv6 etr
 ipv6 itr map-resolver 10.10.10.10
 ipv6 itr map-resolver 10.10.30.10
 ipv6 itr map-resolver 2001:db8:e000:2::1
 ipv6 itr map-resolver 2001:db8:f000:2::1
 ipv6 etr map-server 10.10.10.10 key 0 some-xtr-key
 ipv6 etr map-server 10.10.30.10 key 0 some-xtr-key
 ipv6 etr map-server 2001:db8:e000:2::1 key 0 some-xtr-key
 ipv6 etr map-server 2001:db8:f000:2::1 key 0 some-xtr-key
 exit
!
ip route 0.0.0.0 0.0.0.0 10.2.1.1
!
ipv6 route ::/0 2001:db8:f000:1::1
!
```

Configure a Private LISP Mapping System Using a Standalone Map Resolver/Map Server

Perform this task to configure and enable standalone LISP map resolver/map server (MR/MS) functionality for both IPv4 and IPv6 address families. In this task, a Cisco device is configured as a standalone MR/MS for a private LISP mapping system. Because the MR/MS is configured as a standalone device, it has no need for LISP alternative logical topology (ALT) connectivity. All relevant LISP sites must be configured to register with this map server so that this map server has full knowledge of all registered EID prefixes within the (assumed) private LISP system. However, because this device is functioning as a map resolver/map server, the data structure associated with an ALT virtual routing and forwarding (VRF) table must still be configured to hold LISP EIDs for registered sites.

The map resolver/map server is configured with both IPv4 and IPv6 RLOC addresses. The topology used in this most basic LISP MR/MS configuration is shown in the figure below.

Figure 10: Standalone LISP Map Resolver/Map Server with both IPv4 and IPv6 RLOCs



The components illustrated in the topology shown in the figure are described below, although the map resolver is configured separately:

Mapping System

- The LISP device is configured to function as a standalone map resolver/map server (MR/MS).
- The xTRs in the LISP site are assumed to be registered to this map server. That is, the xTR registers the IPv4 EID prefix of 172.16.1.0/24 and, when IPv6 EIDs are used, the xTR also registers the IPv6 EID of prefix 2001:db8:a::/48.
- The MR/MS has an IPv4 locator of 10.10.10.10/24 and an IPv6 locator of 2001:db8:e000:2::1/64.

SUMMARY STEPS

1. **configure terminal**
2. **vrf definition** *vrf-name*
3. **address-family ipv4** [**unicast**]
4. **exit-address-family**
5. **address-family ipv6**
6. **exit-address-family**
7. **exit**
8. **router lisp**
9. **ipv4 alt-vrf** *vrf-name*
10. **ipv4 map-server**
11. **ipv4 map-resolver**
12. **ipv6 alt-vrf** *vrf-name*
13. **ipv6 map-server**
14. **ipv6 map-resolver**
15. **site** *site-name*
16. **eid-prefix** *EID-prefix*
17. **authentication-key** [*key-type*] *authentication-key*
18. **exit**
19. Repeat Steps 15 through 18 to configure additional LISP sites.
20. **exit**
21. **ip route** *ipv4-prefix next-hop*
22. **ipv6 route** *ipv6-prefix next-hop*
23. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 2	**vrf definition** *vrf-name* **Example:** `Router(config)# vrf definition lisp`	Creates a virtual routing and forwarding (VRF) table and enters VRF configuration mode. • Use the *vrf-name* argument to specify a name to be assigned to the VRF table. In this example, a VRF table named lisp is created to hold EID prefixes.

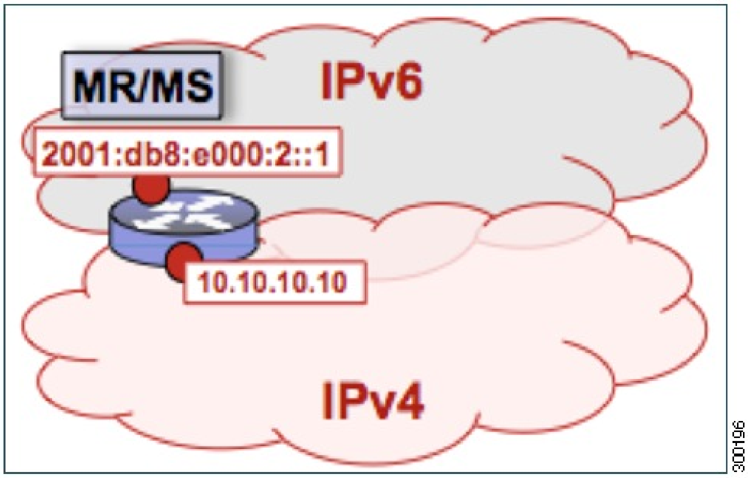
	Command or Action	Purpose
Step 3	**address-family ipv4** [**unicast**] **Example:** `Router(config-vrf)# address-family ipv4`	Enters VRF IPv4 address family configuration mode to specify an IPv4 address family for a VRF table. • In this example, the VRF table named lisp handles IPv4 EID prefixes.
Step 4	**exit-address-family** **Example:** `Router(config-vrf-af)# exit-address-family`	Exits VRF IPv4 address family configuration mode and returns to VRF configuration mode.
Step 5	**address-family ipv6** **Example:** `Router(config-vrf)# address-family ipv6`	Enters VRF IPv6 address family configuration mode to specify an IPv6 address family for a VRF table. • In this example, the VRF table named lisp handles IPv6 EID prefixes.
Step 6	**exit-address-family** **Example:** `Router(config-vrf-af)# exit-address-family`	Exits VRF IPv6 address family configuration mode and returns to VRF configuration mode.
Step 7	**exit** **Example:** `Router(config-vrf)# exit`	Exits VRF configuration mode and enters global configuration mode.
Step 8	**router lisp** **Example:** `Router(config)# router lisp`	Enters LISP configuration mode (software only).
Step 9	**ipv4 alt-vrf** *vrf-name* **Example:** `Router(config-router-lisp)# ipv4 alt-vrf lisp`	Associates a VRF table with the LISP ALT for IPv4 EIDs. • In this example, the VRF table named lisp (created in Step 2) is associated with the LISP ALT.
Step 10	**ipv4 map-server** **Example:** `Router(config-router-lisp)# ipv4 map-server`	Enables LISP map server functionality for EIDs in the IPv4 address family.

	Command or Action	Purpose
Step 11	**ipv4 map-resolver** **Example:** `Router(config-router-lisp)# ipv4 map-resolver`	Enables LISP map resolver functionality for EIDs in the IPv4 address family.
Step 12	**ipv6 alt-vrf** *vrf-name* **Example:** `Router(config-router-lisp)# ipv6 alt-vrf lisp`	Associates a VRF table with the LISP ALT for IPv6 EIDs. • In this example, the VRF table named lisp (created in Step 2) is associated with the LISP ALT.
Step 13	**ipv6 map-server** **Example:** `Router(config-router-lisp)# ipv6 map-server`	Enables LISP map server functionality for EIDs in the IPv6 address family.
Step 14	**ipv6 map-resolver** **Example:** `Router(config-router-lisp)# ipv6 map-resolver`	Enables LISP map resolver functionality for EIDs in the IPv6 address family.
Step 15	**site** *site-name* **Example:** `Router(config-router-lisp)# site Site-1`	Specifies a LISP site named Site-1 and enters LISP site configuration mode. **Note** 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	**eid-prefix** *EID-prefix* **Example:** `Router(config-router-lisp-site)# eid-prefix 172.16.1.0/24`	Configures an IPv4 or IPv6 EID prefix associated with this LISP site. • Repeat this step as necessary to configure additional EID prefixes under this LISP sites. • In this step example, only an IPv4 EID prefix is configured but to complete the configuration, an IPv6 EID prefix must also be configured. **Note** The LISP ETR must be configured with matching EID prefixes and an identical authentication key. **Note** Additional **eid-prefix** command configuration options are available. (See the *LISP Command Reference* for more details.)
Step 17	**authentication-key** [*key-type*] *authentication-key*	Configures the authentication key associated with this site.

	Command or Action	Purpose
	Example: `Router(config-router-lisp-site)# authentication-key 0 some-key`	**Note** The LISP ETR must be configured with matching EID prefixes and an identical authentication key. **Note** The **authentication-key** can be configured with Type 6 encryption. (See the *LISP Command Reference* for more details.)
Step 18	**exit** **Example:** `Router(config-router-lisp-site)# exit`	Exits LISP site configuration mode and returns to LISP configuration mode.
Step 19	Repeat Steps 15 through 18 to configure additional LISP sites.	—
Step 20	**exit** **Example:** `Router(config-router-lisp)# exit`	Exits LISP configuration mode and returns to global configuration mode.
Step 21	**ip route** *ipv4-prefix next-hop* **Example:** `Router(config)# ip route 0.0.0.0 0.0.0.0 10.1.1.1`	Configures an IPv4 static route. • In this example, a default route to the upstream next hop for all IPv4 destinations is created.
Step 22	**ipv6 route** *ipv6-prefix next-hop* **Example:** `Router(config)# ipv6 route ::/0 2001:db8:e000:1::1`	Configures an IPv6 static route. • In this example, a default route to the upstream next hop for all IPv6 destinations is created.
Step 23	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Example:

Figure 11: Standalone LISP Map Resolver/Map Server with both IPv4 and IPv6 RLOCs



The example below shows the complete configuration for the LISP topology illustrated in the figure above and in this task. However, this example is for a full configuration of a standalone LISP MR/MS and includes some basic IPv4 and IPv6 configuration not covered in this task:

```
!
hostname MR-MS
!
vrf definition lisp
!
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 exit-address-family
!
no ip domain lookup
ip cef
ipv6 unicast-routing
ipv6 cef
!
interface Loopback0
 ip address 172.17.2.1 255.255.255.255
!
interface LISP0
!
interface GigabitEthernet0/0/0
 description Link to SP1 (RLOC)
 ip address 10.10.10.10 255.255.255.0
 ipv6 address 2001:db8:e000:2::1/64
!
router lisp
 site Site-1
  authentication-key some-key
  eid-prefix 172.16.1.0/24
  eid-prefix 2001:db8:a::/48
  exit
 !
 site Site-2
  authentication-key another-key
  eid-prefix 172.16.2.0/24
  eid-prefix 2001:db8:b::/48
  exit
```

```
 !
!---more LISP site configs---
 !
 ipv4 map-server
 ipv4 map-resolver
 ipv4 alt-vrf lisp
 ipv6 map-server
 ipv6 map-resolver
 ipv6 alt-vrf lisp
 exit
!
ip route 0.0.0.0 0.0.0.0 10.10.10.1
!
ipv6 route ::/0 2001:db8:e000:2::fof
```

Configure a Public Mapping System Using Separate ALT-Connected Map Resolver and Map Server Devices

The following tasks show how to configure a map resolver (MR) and a map server (MS) on separate devices, each using LISP alternative logical topology (ALT) connectivity. The MR and MS share their EID prefix information via the LISP ALT connectivity, which is typical of a public LISP deployment model where higher performance and scalability (for tasks such as the handling of Map-Request messages) is required. The LISP ALT is implemented as an overlay virtualized network using GRE tunnels and BGP, which allows for separation of EID prefixes from the underlying core network.

Configuring an ALT-Connected LISP Map Resolver

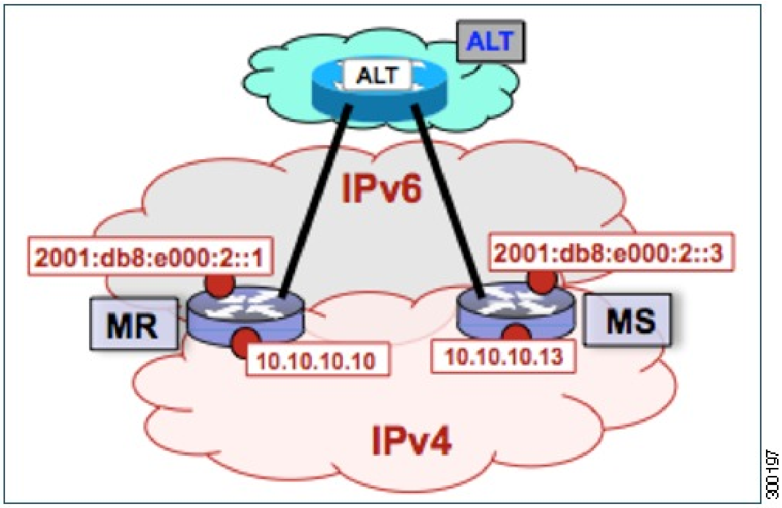
Before You Begin

Perform this task to configure LISP alternative logical topology (ALT) map resolver functionality for both IPv4 and IPv6 address family mapping services.

Note You must also configure an ALT-connected LISP map server (see the Configuring an ALT-Connected LISP Map Server task).

In the figure below, the map resolver (MR) and map server (MS) are configured on separate devices and share their EID prefix information via connectivity.

Figure 12: ALT-Connected LISP Map Resolver and Map Server, each having both an IPv4 and an IPv6 RLOC



The map resolver illustrated in the topology shown in the figure is described below; the map server and LISP ALT are configured in separate tasks:

Mapping System

- Two LISP devices are configured, one as an MS and the other as an MR.
- The MS has an IPv4 locator of 10.10.10.13/24 and an IPv6 locator of 2001:db8:e000:2::3/64.
- The MR has an IPv4 locator of 10.10.10.10/24 and an IPv6 locator of 2001:db8:e000:2::1/64.
- Assume that the xTRs in the LISP site register to this map server. That is, the xTR registers the IPv4 EID-prefix of 172.16.1.0/24 and, when IPv6 EIDs are used, the xTR registers the IPv6 EID-prefix of 2001:db8:a::/48.

Note The configuration of the xTR must be changed to use the MS RLOC for its map server configuration and the MR RLOC for its map resolver configuration. For example:

- **ipv4 itr map-resolver 10.10.10.10**
- **ipv4 etr map-server 10.10.10.13 key 0 some-key**

Other Infrastructure

- The MR has IPv4 and IPv6 tunnel endpoints in the VRF table (named lisp) of 192.168.1.1/30 and 2001:db8:ffff::1/64, respectively, and the MS has IPv4 and IPv6 tunnel endpoints of 192.168.1.2/30 and 2001:db8:ffff::2/64, respectively, in the same VRF table. This tunnel is used for the ALT.

SUMMARY STEPS

1. **configure terminal**
2. **vrf definition** *vrf-name*
3. **rd** *route-distinguisher*
4. **address-family ipv4** [**unicast**]
5. **exit-address-family**
6. **address-family ipv6**
7. **exit-address-family**
8. **exit**
9. **interface** *type number*
10. **vrf forwarding** *vrf-name*
11. **ip address** *ip-address mask*
12. **ipv6 address** *ipv6-address/mask*
13. **tunnel source** *interface-type interface-number*
14. **tunnel destination** *ipv4-address*
15. **exit**
16. **router lisp**
17. **ipv4 map-resolver**
18. **ipv4 alt-vrf** *vrf-name*
19. **ipv6 map-resolver**
20. **ipv6 alt-vrf** *vrf-name*
21. **exit**
22. **router bgp** *autonomous-system-number*
23. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
24. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
25. **neighbor** *ip-address* **activate**
26. **exit**
27. **address-family ipv6 vrf** *vrf-name*
28. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
29. **neighbor** *ip-address* **activate**
30. **exit**
31. **exit**
32. **ip route** *ipv4-prefix next-hop*
33. **ipv6 route** *ipv6-prefix next-hop*
34. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 2	**vrf definition** *vrf-name* **Example:** `Router(config)# vrf definition lisp`	Creates a virtual routing and forwarding (VRF) table and enters VRF configuration mode. • Use the *vrf-name* argument to specify a name to be assigned to the VRF. In this example, a VRF named lisp is created to hold EID prefixes.
Step 3	**rd** *route-distinguisher* **Example:** `Router(config-vrf)# rd 1:1`	Creates routing and forwarding tables for a VRF.
Step 4	**address-family ipv4** [**unicast**] **Example:** `Router(config-vrf)# address-family ipv4`	Enters VRF IPv4 address family configuration mode to specify an IPv4 address family for a VRF table. • In this example, the VRF table named lisp handles IPv4 EID prefixes.
Step 5	**exit-address-family** **Example:** `Router(config-vrf-af)# exit-address-family`	Exits VRF IPv4 address family configuration mode and returns to VRF configuration mode.
Step 6	**address-family ipv6** **Example:** `Router(config-vrf)# address-family ipv6`	Enters VRF IPv6 address family configuration mode to specify an IPv6 address family for a VRF table. • In this example, the VRF table named lisp handles IPv6 EID prefixes.
Step 7	**exit-address-family** **Example:** `Router(config-vrf-af)# exit-address-family`	Exits VRF IPv6 address family configuration mode and returns to VRF configuration mode.
Step 8	**exit** **Example:** `Router(config-vrf)# exit`	Exits VRF configuration mode and enters global configuration mode.

	Command or Action	Purpose
Step 9	**interface** *type number* **Example:** `Router(config)# interface tunnel 192`	Specifies the interface type of tunnel and the interface number and enters interface configuration mode.
Step 10	**vrf forwarding** *vrf-name* **Example:** `Router(config-if)# vrf forwarding lisp`	Associates a VRF instance configured in Step 2 with the tunnel interface configured in Step 9. • When the interface is bound to a VRF, previously configured IP addresses are removed, and the interface is disabled.
Step 11	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 192.168.1.1 255.255.255.252`	Configures an IPv4 address for the tunnel interface.
Step 12	**ipv6 address** *ipv6-address/mask* **Example:** `Router(config-if)# ipv6 address 2001:db8:ffff::1/64`	Configures an IPv6 address for the tunnel interface.
Step 13	**tunnel source** *interface-type interface-number* **Example:** `Router(config-if)# tunnel source GigabitEthernet 0/0/0`	Configures the tunnel source.
Step 14	**tunnel destination** *ipv4-address* **Example:** `Router(config-if)# tunnel destination 10.10.10.13`	Configures the tunnel destination IPv4 address for the tunnel interface.
Step 15	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 16	**router lisp** **Example:** `Router(config)# router lisp`	Enters LISP configuration mode (software only).

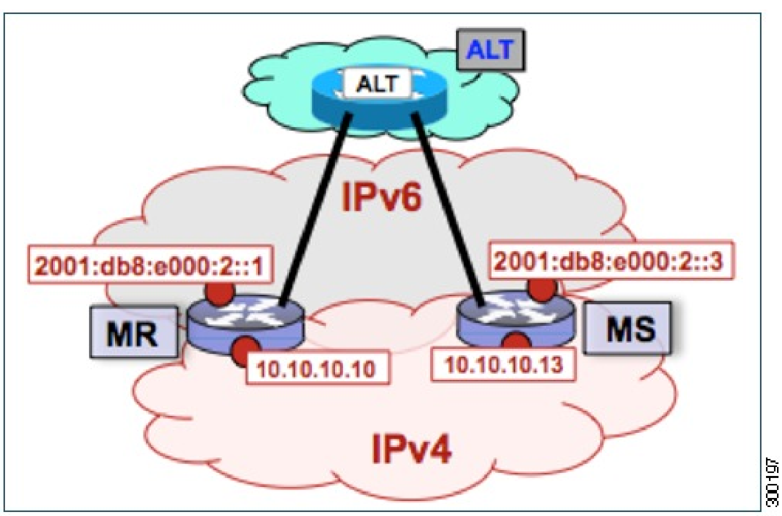
	Command or Action	Purpose
Step 17	**ipv4 map-resolver** **Example:** `Router(config-router-lisp)# ipv4 map-resolver`	Enables LISP map resolver functionality for EIDs in the IPv4 address family.
Step 18	**ipv4 alt-vrf** *vrf-name* **Example:** `Router(config-router-lisp)# ipv4 alt-vrf lisp`	Associates a VRF table with the LISP ALT for IPv4 EIDs. • In this example, the VRF table named lisp (created in Step 2) is associated with the LISP ALT.
Step 19	**ipv6 map-resolver** **Example:** `Router(config-router-lisp)# ipv6 map-resolver`	Enables LISP map resolver functionality for EIDs in the IPv6 address family.
Step 20	**ipv6 alt-vrf** *vrf-name* **Example:** `Router(config-router-lisp)# ipv6 alt-vrf lisp`	Associates a VRF table with the LISP ALT for IPv6 EIDs. • In this example, the VRF table named lisp (created in Step 2) is associated with the LISP ALT.
Step 21	**exit** **Example:** `Router(config-router-lisp)# exit`	Exits LISP configuration mode and returns to global configuration mode.
Step 22	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 65010`	Enters router configuration mode for the specified routing process.
Step 23	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] **Example:** `Router(config-router)# address-family ipv4 vrf lisp`	Specifies the IPv4 address family and enters IPv4 address family configuration mode. • The **vrf** keyword and *vrf-name* argument specify the name of the VRF instance to associate with subsequent commands. • In this example, the VRF table named lisp (created in Step 2) is associated with the BGP IPv4 VRF that carries EID-prefixes in the LISP ALT.

	Command or Action	Purpose
Step 24	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Router(config-router-af)# neighbor 192.168.1.2 remote-as 65011`	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.
Step 25	**neighbor** *ip-address* **activate** **Example:** `Router(config-router-af)# neighbor 192.168.1.2 activate`	Enables the neighbor to exchange prefixes for the IPv4 unicast address family.
Step 26	**exit** **Example:** `Router(config-router-af)# exit`	Exits IPv4 address family configuration mode and returns to router configuration mode.
Step 27	**address-family ipv6 vrf** *vrf-name* **Example:** `Router(config-router)# address-family ipv6 vrf lisp`	Specifies the IPv6 address family and enters IPv6 address family configuration mode. • The **vrf** keyword and *vrf-name* argument specify the name of the VRF instance to associate with subsequent commands. • In this example, the VRF table named lisp (created in Step 2) is associated with the BGP IPv6 VRF that carries EID-prefixes in the LISP ALT.
Step 28	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Router(config-router-af)# neighbor 2001:db8:ffff::2 remote-as 65011`	Adds the IPv6 address of the neighbor in the specified autonomous system to the IPv6 multiprotocol BGP neighbor table of the local router.
Step 29	**neighbor** *ip-address* **activate** **Example:** `Router(config-router-af)# neighbor 2001:db8:ffff::2 activate`	Enables the neighbor to exchange prefixes for the IPv6 unicast address family.
Step 30	**exit** **Example:** `Router(config-router-af)# exit`	Exits address family configuration mode and returns to router configuration mode.

	Command or Action	Purpose
Step 31	**exit** **Example:** `Router(config-router)# exit`	Exits router configuration mode and returns to global configuration mode.
Step 32	**ip route** *ipv4-prefix next-hop* **Example:** `Router(config)# ip route 0.0.0.0 0.0.0.0 10.10.10.1`	Configures an IPv4 static route. • In this example, a default route to the upstream next hop for all IPv4 destinations is created.
Step 33	**ipv6 route** *ipv6-prefix next-hop* **Example:** `Router(config)# ipv6 route ::/0 2001:db8:e000:2::f0f`	Configures an IPv6 static route. • In this example, a default route to the upstream next hop for all IPv6 destinations is created.
Step 34	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Examples

Figure 13: ALT-Connected LISP Map Resolver and Map Server, each having both an IPv4 and an IPv6 RLOC



The example below shows the full configuration for a LISP map resolver including some basic IP and IPv6 configuration not included in the task table for this task:

```
!
vrf definition lisp
 rd 1:1
 !
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 exit-address-family
!
no ip domain lookup
ip cef
ipv6 unicast-routing
ipv6 cef
!
interface Loopback0
 no ip address
!
interface Tunnel192
 vrf forwarding lisp
 ip address 192.168.1.1 255.255.255.252
 ipv6 address 2001:db8:ffff::1/64
 tunnel source GigabitEthernet 0/0/0
 tunnel destination 10.10.10.13
!
interface GigabitEthernet 0/0/0
 description Link to SP1 (RLOC)
 ip address 10.10.10.10 255.255.255.0
 ipv6 address 2001:db8:e000:2::1/64
!
router lisp
 ipv4 map-resolver
 ipv4 alt-vrf lisp
 ipv6 map-resolver
 ipv6 alt-vrf lisp
 exit
!
router bgp 65010
 bgp asnotation dot
 bgp log-neighbor-changes
 !
 address-family ipv4 vrf lisp
  neighbor 192.168.1.2 remote-as 65011
  neighbor 192.168.1.2 activate
 exit-address-family
 !
 address-family ipv6 vrf lisp
  neighbor 2001:db8:ffff::2 remote-as 65011
  neighbor 2001:db8:ffff::2 activate
 exit-address-family
!
ip route 0.0.0.0 0.0.0.0 10.10.10.1
!
ipv6 route ::/0 2001:db8:e000:2::f0f
!
```

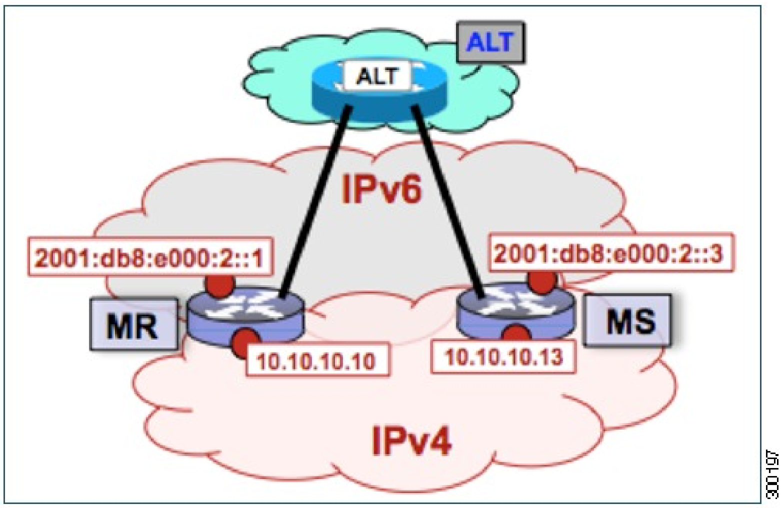
Configuring an ALT-Connected LISP Map Server

Perform this task to configure LISP alternative logical topology (ALT) map server functionality for both IPv4 and IPv6 address family mapping services.

Note You must also configure an ALT-connected LISP map resolver (see the Configuring an ALT-Connected LISP Map Resolver task).

In the figure below, the map resolver (MR) and map server (MS) are configured on separate devices and share their EID prefix information via connectivity.

Figure 14: ALT-Connected LISP Map Resolver and Map Server, each having both an IPv4 and an IPv6 RLOC



The map server illustrated in the topology shown in the figure is described below; the map resolver and LISP ALT are configured in separate tasks:

Mapping System

- Two LISP devices are configured, one as an MS and the other as an MR.
- The MS has an IPv4 locator of 10.10.10.13/24 and an IPv6 locator of 2001:db8:e000:2::3/64.
- The MR has an IPv4 locator of 10.10.10.10/24 and an IPv6 locator of 2001:db8:e000:2::1/64.
- Assume that the xTRs in the LISP site register to this map server. That is, the xTR registers the IPv4 EID-prefix of 172.16.1.0/24 and, when IPv6 EIDs are used, the xTR registers the IPv6 EID-prefix of 2001:db8:a::/48.

Note The configuration of the xTR must be changed to use the MS RLOC for its map server configuration and the MR RLOC for its map resolver configuration. For example:

- **ipv4 itr map-resolver 10.10.10.10**
- **ipv4 etr map-server 10.10.10.13 key 0 some-key**

Other Infrastructure

- The MR has IPv4 and IPv6 tunnel endpoints in the VRF table (named lisp) of 192.168.1.1/30 and 2001:db8:ffff::1/64, respectively, and the MS has IPv4 and IPv6 tunnel endpoints of 192.168.1.2/30 and 2001:db8:ffff::2/64, respectively, in the same VRF table. This tunnel is used for the ALT.

SUMMARY STEPS

1. **configure terminal**
2. **vrf definition** *vrf-name*
3. **rd** *route-distinguisher*
4. **address-family ipv4** [**unicast**]
5. **exit-address-family**
6. **address-family ipv6**
7. **exit-address-family**
8. **exit**
9. **interface** *type number*
10. **vrf forwarding** *vrf-name*
11. **ip address** *ip-address mask*
12. **ipv6 address** *ipv6-address/mask*
13. **tunnel source** *interface-type interface-number*
14. **tunnel destination** *ipv4-address*
15. **exit**
16. **router lisp**
17. **ipv4 map-server**
18. **ipv4 alt-vrf** *vrf-name*
19. **ipv6 map-server**
20. **ipv6 alt-vrf** *vrf-name*
21. **site** *site-name*
22. **eid-prefix** *EID-prefix*
23. **authentication-key** *key-type authentication-key*
24. **exit**
25. Repeat Steps 21 through 24 to configure additional LISP sites.
26. **exit**
27. **router bgp** *autonomous-system-number*
28. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
29. **redistribute lisp**
30. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
31. **neighbor** *ip-address* **activate**
32. **exit**
33. **address-family ipv6 vrf** *vrf-name*
34. **redistribute lisp**
35. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
36. **neighbor** *ip-address* **activate**
37. **exit**
38. **exit**
39. **ip route** *ipv4-prefix next-hop*

40. **ipv6 route** *ipv6-prefix next-hop*
41. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 2	**vrf definition** *vrf-name* **Example:** `Router(config)# vrf definition lisp`	Creates a virtual routing and forwarding (VRF) table and enters VRF configuration mode. • Use the *vrf-name* argument to specify a name to be assigned to the VRF. In this example, a VRF named lisp is created to hold EID prefixes.
Step 3	**rd** *route-distinguisher* **Example:** `Router(config-vrf)# rd 1:1`	Creates routing and forwarding tables for a VRF.
Step 4	**address-family ipv4** [**unicast**] **Example:** `Router(config-vrf)# address-family ipv4`	Enters VRF IPv4 address family configuration mode to specify an IPv4 address family for a VRF table. • In this example, the VRF table named lisp handles IPv4 EID prefixes.
Step 5	**exit-address-family** **Example:** `Router(config-vrf-af)# exit-address-family`	Exits VRF IPv4 address family configuration mode and returns to VRF configuration mode.
Step 6	**address-family ipv6** **Example:** `Router(config-vrf)# address-family ipv6`	Enters VRF IPv6 address family configuration mode to specify an IPv6 address family for a VRF table. • In this example, the VRF table named lisp handles IPv6 EID prefixes.
Step 7	**exit-address-family** **Example:** `Router(config-vrf-af)# exit-address-family`	Exits VRF IPv6 address family configuration mode and returns to VRF configuration mode.

	Command or Action	Purpose
Step 8	**exit** **Example:** `Router(config-vrf)# exit`	Exits VRF configuration mode and enters global configuration mode.
Step 9	**interface** *type number* **Example:** `Router(config)# interface tunnel 191`	Specifies the interface type of tunnel and the interface number and enters interface configuration mode.
Step 10	**vrf forwarding** *vrf-name* **Example:** `Router(config-if)# vrf forwarding lisp`	Associates a VRF instance configured in Step 2 with the tunnel interface configured in Step 9. • When the interface is bound to a VRF, previously configured IP addresses are removed, and the interface is disabled.
Step 11	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 192.168.1.6 255.255.255.252`	Configures an IPv4 address for the tunnel interface.
Step 12	**ipv6 address** *ipv6-address/mask* **Example:** `Router(config-if)# ipv6 address 2001:DB8:ffff::6/64`	Configures an IPv6 address for the tunnel interface.
Step 13	**tunnel source** *interface-type interface-number* **Example:** `Router(config-if)# tunnel source GigabitEthernet 0/0/0`	Configures the tunnel source.
Step 14	**tunnel destination** *ipv4-address* **Example:** `Router(config-if)# tunnel destination 10.10.10.13`	Configures the tunnel destination IPv4 address for the tunnel interface.
Step 15	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.

	Command or Action	Purpose
Step 16	**router lisp** **Example:** `Router(config)# router lisp`	Enters LISP configuration mode (software only).
Step 17	**ipv4 map-server** **Example:** `Router(config-router-lisp)# ipv4 map-server`	Enables LISP map server functionality for EIDs in the IPv4 address family.
Step 18	**ipv4 alt-vrf** *vrf-name* **Example:** `Router(config-router-lisp)# ipv4 alt-vrf lisp`	Associates a VRF table with the LISP ALT for IPv4 EIDs. • In this example, the VRF table named lisp (created in Step 2) is associated with the LISP ALT.
Step 19	**ipv6 map-server** **Example:** `Router(config-router-lisp)# ipv6 map-server`	Enables LISP map server functionality for EIDs in the IPv6 address family.
Step 20	**ipv6 alt-vrf** *vrf-name* **Example:** `Router(config-router-lisp)# ipv6 alt-vrf lisp`	Associates a VRF table with the LISP ALT for IPv6 EIDs. • In this example, the VRF table named lisp (created in Step 2) is associated with the LISP ALT.
Step 21	**site** *site-name* **Example:** `Router(config-router-lisp)# site Site-1`	Specifies a LISP site and enters LISP site configuration mode. **Note** 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 22	**eid-prefix** *EID-prefix* **Example:** `Router(config-router-lisp-site)# eid-prefix 172.16.1.0/24`	Configures an IPv4 or IPv6 EID prefix associated with this LISP site. • Repeat this step as necessary to configure additional EID prefixes under this LISP sites. • In this step example, only an IPv4 EID prefix is configured but to complete the configuration, an IPv6 EID prefix must also be configured. **Note** The LISP ETR must be configured with matching EID prefixes and an identical authentication key.

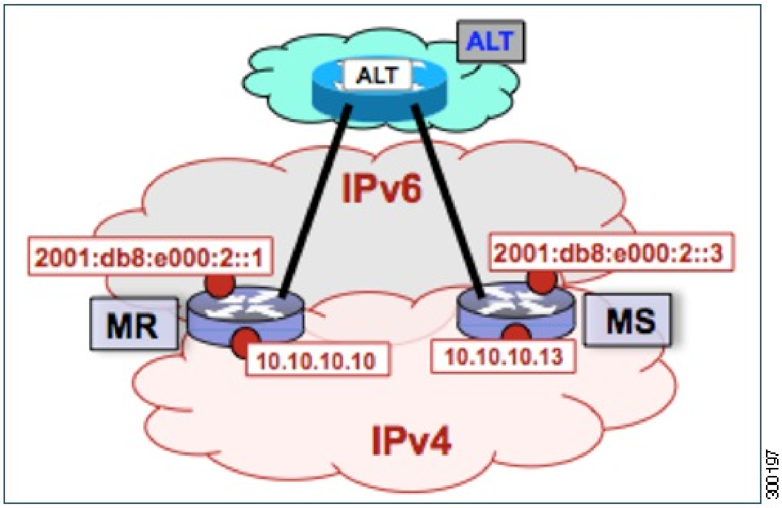
	Command or Action	Purpose
		Note Additional **eid-prefix** command configuration options are available. (See the *LISP Command Reference* for more details.)
Step 23	**authentication-key** *key-type authentication-key* **Example:** `Router(config-router-lisp-site)# authentication-key 0 some-key`	Configures the authentication key associated with this site. **Note** The LISP ETR must be configured with matching EID prefixes and an identical authentication key. **Note** The **authentication-key** can be configured with Type 6 encryption. (See the *LISP Command Reference* for more details.)
Step 24	**exit** **Example:** `Router(config-router-lisp-site)# exit`	Exits LISP site configuration mode and returns to LISP configuration mode.
Step 25	Repeat Steps 21 through 24 to configure additional LISP sites.	—
Step 26	**exit** **Example:** `Router(config-router-lisp)# exit`	Exits LISP configuration mode and returns to global configuration mode.
Step 27	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 65011`	Enters router configuration mode for the specified routing process.
Step 28	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] **Example:** `Router(config-router)# address-family ipv4 vrf lisp`	Specifies the IPv4 address family and enters IPv4 address family configuration mode. • The **vrf** keyword and *vrf-name* argument specify the name of the VRF instance to associate with subsequent commands. • In this example, the VRF table named lisp (created in Step 2) is associated with the BGP IPv4 VRF that carries EID prefixes in the LISP ALT.
Step 29	**redistribute lisp** **Example:** `Router(config-router-af)# redistribute lisp`	Redistributes EID prefixes known to LISP into BGP.

	Command or Action	Purpose
Step 30	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Router(config-router-af)# neighbor 192.168.1.1 remote-as 65010`	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.
Step 31	**neighbor** *ip-address* **activate** **Example:** `Router(config-router-af)# neighbor 192.168.1.1 activate`	Enables the neighbor to exchange prefixes for the IPv4 unicast address family.
Step 32	**exit** **Example:** `Router(config-router-af)# exit`	Exits address family configuration mode and returns to router configuration mode.
Step 33	**address-family ipv6 vrf** *vrf-name* **Example:** `Router(config-router)# address-family ipv6 vrf lisp`	Specifies the IPv6 address family and enters IPv6 address family configuration mode. • The **vrf** keyword and *vrf-name* argument specify the name of the VRF instance to associate with subsequent commands. • In this example, the VRF table named lisp (created in Step 2) is associated with the BGP IPv6 VRF that carries EID prefixes in the LISP ALT.
Step 34	**redistribute lisp** **Example:** `Router(config-router-af)# redistribute lisp`	Redistributes EID prefixes known to LISP into BGP.
Step 35	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Router(config-router-af)# neighbor 2001:db8:ffff::1 remote-as 65010`	Adds the IPv6 address of the neighbor in the specified autonomous system to the IPv6 multiprotocol BGP neighbor table of the local router.
Step 36	**neighbor** *ip-address* **activate** **Example:** `Router(config-router-af)# neighbor 2001:db8:ffff::1 activate`	Enables the neighbor to exchange prefixes for the IPv6 unicast address family.

	Command or Action	Purpose
Step 37	**exit** **Example:** `Router(config-router-af)# exit`	Exits address family configuration mode and returns to router configuration mode.
Step 38	**exit** **Example:** `Router(config-router)# exit`	Exits router configuration mode and returns to global configuration mode.
Step 39	**ip route** *ipv4-prefix next-hop* **Example:** `Router(config)# ip route 0.0.0.0 0.0.0.0 10.10.10.1`	Configures an IPv4 static route. • In this example, a default route to the upstream next hop for all IPv4 destinations is created.
Step 40	**ipv6 route** *ipv6-prefix next-hop* **Example:** `Router(config)# ipv6 route ::/0 2001:db8:e000:2::f0f`	Configures an IPv6 static route. • In this example, a default route to the upstream next hop for all IPv6 destinations is created.
Step 41	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Example:

Figure 15: ALT-Connected LISP Map Resolver and Map Server, each having both an IPv4 and an IPv6 RLOC



The example below shows the full configuration for a LISP map server including some basic IP and IPv6 configuration not included in the task table for this task:

```
!
hostname MS
!
vrf definition lisp
 rd 1:1
 !
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 exit-address-family
!
no ip domain lookup
ip cef
ipv6 unicast-routing
ipv6 cef
!
interface Loopback0
 no ip address
!
interface Tunnel192
 vrf forwarding lisp
 ip address 192.168.1.2 255.255.255.252
 ipv6 address 2001:db8:ffff::2/64
 tunnel source GigabitEthernet 0/0/0
 tunnel destination 10.10.10.10
!
interface GigabitEthernet 0/0/0
 description Link to SP1 (RLOC)
 ip address 10.10.10.13 255.255.255.0
 ipv6 address 2001:db8:e000:2::3/64
!
router lisp
 site Site-1
  authentication-key 0 some-xtr-key
  eid-prefix 172.16.1.0/24
  eid-prefix 2001:db8:a::/48
  exit
```

```
 !
 site Site-2
  authentication-key 0 another-xtr-key
  eid-prefix 172.16.2.0/24
  eid-prefix 2001:db8:b::/48
  exit
 !
 !---configure more LISP sites as required---
 !
 ipv4 map-server
 ipv4 alt-vrf lisp
 ipv6 map-server
 ipv6 alt-vrf lisp
 exit
!
router bgp 65011
 bgp asnotation dot
 bgp log-neighbor-changes
 !
 address-family ipv4 vrf lisp
  redistribute lisp
  neighbor 192.168.1.1 remote-as 65010
  neighbor 192.168.1.1 activate
 exit-address-family
 !
 address-family ipv6 vrf lisp
  redistribute lisp
  neighbor 2001:db8:ffff::1 remote-as 65010
  neighbor 2001:db8:ffff::1 activate
 exit-address-family
!
ip route 0.0.0.0 0.0.0.0 10.10.10.1
!
ipv6 route ::/0 2001:db8:e000:2::f0f
```

Configure a PETR and a PITR

The following tasks show how to design and deploy a Proxy Egress Tunnel Router (PETR) and a Proxy Ingress Tunnel Router (PITR). The example scenario shows deployment of a PETR and PITR as separate devices but it is also possible to deploy a single device that acts simultaneously as a PETR and a PITR, which is called a PxTR.

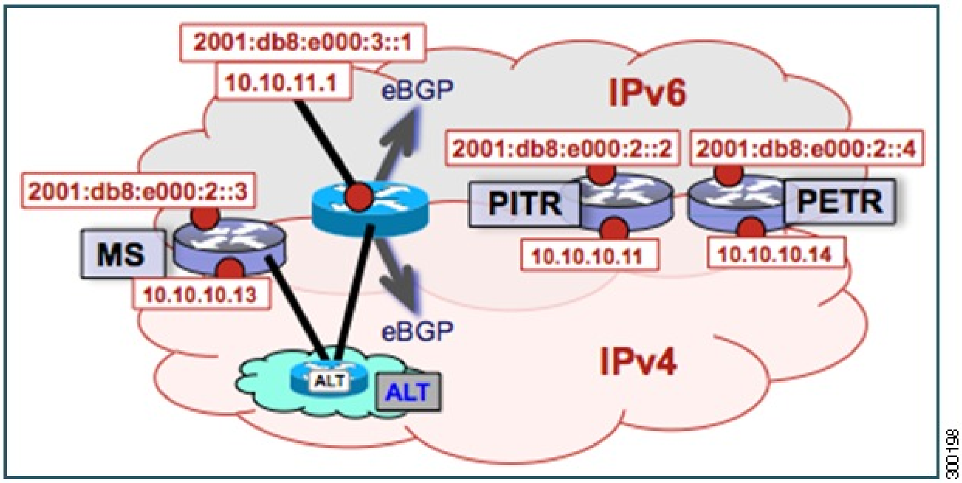
Deploying a Proxy Egress Tunnel Router with both an IPv4 and an IPv6 RLOC

Perform this task to deploy a Proxy Egress Tunnel Router (PETR) for both IPv4 and IPv6 address families. You can also perform this task to configure PETR functionality on a single device that acts simultaneously as a PETR and as a Proxy Ingress Tunnel Router (PITR), referred to as a PxTR.

A PETR simply takes in LISP encapsulated packets and decapsulates them and forwards them. For example, a PETR can be used to provide IPv6 LISP EIDs access to non-LISP EIDs when the LISP site only has IPv4 RLOC connectivity. A PETR, therefore, is used for LISP-to-non-LISP access in situations where cross-address family connectivity is an issue. (A PETR can still be used for matching EID and RLOC address families if desired.) Note that a PITR is required to provide return-traffic flow. A PETR is simple to deploy because it need only provide dual-stack connectivity to the core.

The topology used in this PETR example is shown in the figure. The PETR and PITR in this example are deployed as separate devices and each have both an IPv4 and an IPv6 locator.

Figure 16: Proxy Egress Tunnel Router with both an IPv4 and an IPv6 RLOC



The components illustrated in the topology shown in the figure are described below:

PETR

- When deployed as a standalone LISP device, the PETR has dual-stack connectivity to the core network.
- The PETR IPv4 locator is 10.10.10.14/24 and the IPv6 locator is 2001:db8:e000:2::4/64.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router lisp**
4. **ipv4 proxy-etr**
5. **ipv6 proxy-etr**
6. **exit**
7. **ip route** *ipv4-prefix next-hop*
8. **ipv6 route** *ipv6-prefix next-hop*
9. **exit**

DETAILED STEPS

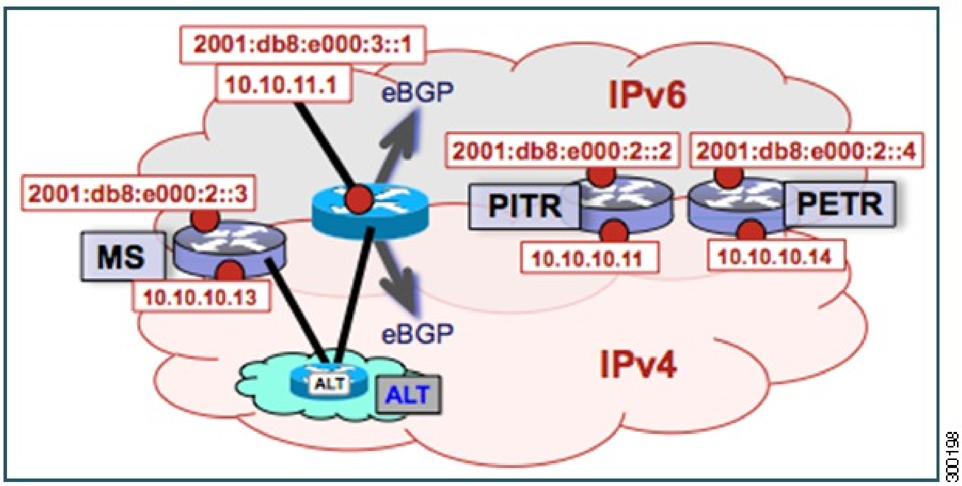
	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router lisp** **Example:** `Router(config)# router lisp`	Enters LISP configuration mode (software only).
Step 4	**ipv4 proxy-etr** **Example:** `Router(config-router-lisp)# ipv4 proxy-etr`	Enables PETR functionality for IPv4 EIDs.
Step 5	**ipv6 proxy-etr** **Example:** `Router(config-router-lisp)# ipv6 proxy-etr`	Enables PETR functionality for IPv6 EIDs.
Step 6	**exit** **Example:** `Router(config-router-lisp)# exit`	Exits LISP configuration mode and enters global configuration mode.
Step 7	**ip route** *ipv4-prefix next-hop* **Example:** `Router(config)# ip route 0.0.0.0 0.0.0.0 10.10.10.1`	Configures an IPv4 static route. • In this example, a default route to the upstream next hop for all IPv4 destinations is created.
Step 8	**ipv6 route** *ipv6-prefix next-hop* **Example:** `Router(config)# ipv6 route ::/0 2001:db8:e000:2::f0f`	Configures an IPv6 static route. • In this example, a default route to the upstream next hop for all IPv6 destinations is created.

	Command or Action	Purpose
Step 9	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Example:

Figure 17: Proxy Egress Tunnel Router with both an IPv4 and an IPv6 RLOC



The example below shows the full configuration for a PETR including some basic IP and IPv6 configuration not included in the task table for this task:

```
!
hostname PETR
!
no ip domain lookup
ip cef
ipv6 unicast-routing
ipv6 cef
!
interface Loopback0
 no ip address
!
interface GigabitEthernet 0/0/0
 description Link to Core (RLOC)
 ip address 10.10.10.14 255.255.255.0
 ipv6 address 2001:db8:e000:2::4/64
!
router lisp
 ipv4 proxy-etr
 ipv6 proxy-etr
 exit
!
ip route 0.0.0.0 0.0.0.0 10.10.10.1
```

```
!
ipv6 route ::/0 2001:db8:e000:2::f0f
```

Deploying a Proxy Ingress Tunnel Router with both an IPv4 and an IPv6 RLOC

Perform this task to deploy a Proxy Ingress Tunnel Router (PITR) for both IPv4 and IPv6 address families. You can also perform this task to configure PITR functionality on a single device that acts simultaneously as a PITR and as a Proxy Egress Tunnel Router (PETR), referred to as a PxTR.

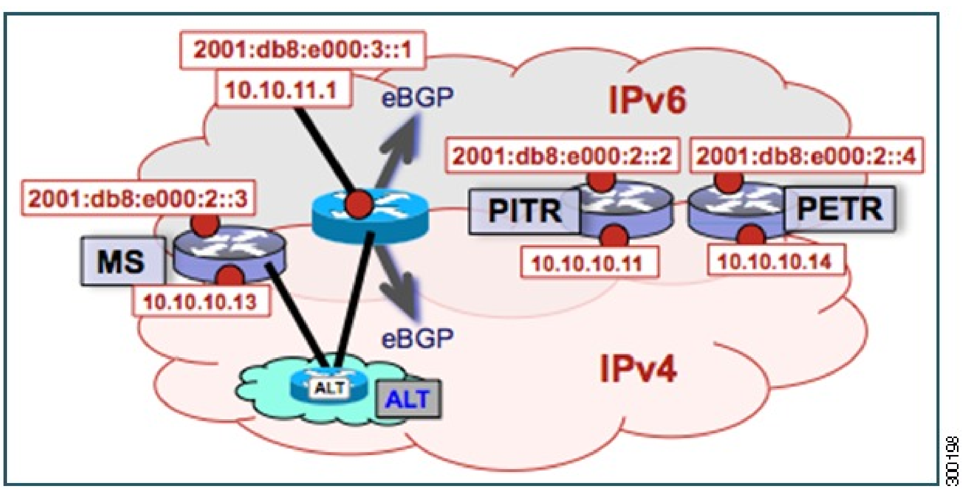
A PITR attracts non-LISP packets by advertising a coarse-aggregate prefix for LISP EIDs into the core (such as the Internet or a Multiprotocol Label Switching (MPLS) core) and then performs LISP encapsulation services (like an ITR) to provide access to LISP EIDs. Thus, a PITR provides non-LISP-to-LISP interworking. A PITR is also used to provide address family “hop-over□? for non-LISP-to-LISP traffic. For example, a dual-stacked PxTR can be used to provide a return-traffic path from non-LISP IPv6 sites to IPv6 LISP sites that contain only IPv4 RLOCs.

To resolve EID-to-RLOC mappings for creating non-LISP-to-LISP flows, configure PITR to query the LISP mapping system. In this task, the PITR is configured to send Map-Rrequest messages via the LISP alternate logical topology (ALT) to resolve EID-to-RLOC mappings.

Note To attract non-LISP traffic destined to LISP sites, the PITR must advertise coarse-aggregate EID prefixes into the underlying network infrastructure. In an Internet-as-the-core example, attracting non-LISP traffice destined to LISP sites is typically managed via external BGP (eBGP) and by advertising the coarse-aggregate that includes all appropriate EID prefixes into the Internet. The example configuration in the figure utilizes this approach. Because this is a standard BGP configuration, summary and detailed command guidance is not provided in the task table for this task, although the complete configuration example that follows the task table does include an accurate example of this eBGP peering. Any other approach that advertises coarse-aggregates that include all appropriate EID prefixes into the core are also acceptable.

The topology used in this example is shown in the figure. The PITR is deployed as a separate device, with both an IPv4 and an IPv6 locator. A map resolver and core-peering router are also shown in the figure for reference because they are required components for completing the PITR configuration shown in the figure.

Figure 18: Proxy Ingress Tunnel Router with both an IPv4 and an IPv6 RLOC



The components illustrated in the topology shown in the figure are described below:

PITR

- When deployed as a standalone LISP device, the PITR has dual-stack connectivity to the core network.
- The PITR IPv4 locator is 10.10.10.11/24 and the IPv6 locator is 2001:db8:e000:2::2/64.
- The use of LISP EID prefixes throughout this task (172.16.1.0/24 and 2001:db8:a::/48 configuration) is assumed and are part of LISP EID blocks that can be summarized in coarse-aggregates and advertised by the PITR into the core network. The advertisement of the IPv4 coarse-aggregate of 172.16.0.0/16 and the IPv6 coarse-aggregate of 2001:db8::/33 by the PITR into the IPv4 and IPv6 core networks is also assumed.
- The PITR eBGP peers with the core router with locators 10.10.11.1 and 2001:db8:e000:3::1 in order to advertise the coarse-aggregate IPv4 EID prefix of 172.16.0.0/16 and the IPv6 EID prefix of 2001:db8::/33 into the IPv4 and IPv6 cores, respectively.
- The PITR is configured to use the LISP ALT (GRE+BGP) via the map server with locators 10.10.10.13 and 2001:db8:e000:2::3. The relevant configuration is shown for the PITR.

Other Infrastructure

- The MS has IPv4 and IPv6 tunnel endpoints in the VRF table (named lisp) of 192.168.5/30 and 2001:db8:ffff::5/64, respectively. The configuration of the map server is not in the task table.
- The core router has an IPv4 address of 10.10.11.1 and an IPv6 address of 2001:db8:e000:3::1. These addresses will be used for eBGP peering. The core router configuration is assumed to be familiar as a typical ISP peering router and is therefore not included in the task table.

SUMMARY STEPS

1. **configure terminal**
2. **vrf definition** *vrf-name*
3. **rd** *route-distinguisher*
4. **address-family ipv4** [**unicast**]
5. **exit-address-family**
6. **address-family ipv6**
7. **exit-address-family**
8. **exit**
9. **interface** *type number*
10. **vrf forwarding** *vrf-name*
11. **ip address** *ip-address mask*
12. **ipv6 address** *ipv6-address/mask*
13. **tunnel source** *interface-type interface-number*
14. **tunnel destination** *ipv4-address*
15. **exit**
16. **router lisp**
17. **ipv4 alt-vrf** *vrf-name*
18. **ipv4 proxy-itr** *ipv4-locator* [*ipv6-locator*]
19. **ipv4 map-cache-limit** *map-cache-limit*
20. **ipv6 alt-vrf** *vrf-name*
21. **ipv6 proxy-itr** *ipv6-locator* [*ipv4-locator*]
22. **ipv6 map-cache-limit** *map-cache-limit*
23. **exit**
24. **router bgp** *autonomous-system-number*
25. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*]
26. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
27. **neighbor** *ip-address* **activate**
28. **exit**
29. **address-family ipv6** [**unicast** | **multicast** | **vrf** *vrf-name*]
30. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
31. **neighbor** *ip-address* **activate**
32. **exit**
33. **exit**
34. **ip route** *ipv4-prefix next-hop*
35. **ip route** *ipv4-prefix next-hop*
36. **ipv6 route** *ipv6-prefix next-hop*
37. **ipv6 route** *ipv6-prefix next-hop*
38. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 2	**vrf definition** *vrf-name* **Example:** `Router(config)# vrf definition lisp`	Configures a virtual routing and forwarding (VRF) table and enters VRF configuration mode. • Use the *vrf-name* argument to specify a name to be assigned to the VRF. In this example, a VRF named lisp is created to hold EID prefixes.
Step 3	**rd** *route-distinguisher* **Example:** `Router(config-vrf)# rd 1:1`	Creates routing and forwarding tables for a VRF.
Step 4	**address-family ipv4** [**unicast**] **Example:** `Router(config-vrf)# address-family ipv4`	Enters VRF IPv4 address family configuration mode to specify an IPv4 address family for a VRF table. • In this example, the VRF named lisp handles IPv4 EID prefixes.
Step 5	**exit-address-family** **Example:** `Router(config-vrf-af)# exit-address-family`	Exits VRF address family configuration mode and returns to VRF configuration mode.
Step 6	**address-family ipv6** **Example:** `Router(config-vrf)# address-family ipv6`	Enters VRF IPv6 address family configuration mode to specify an IPv6 address family for a VRF table. • In this example, the VRF table named lisp handles IPv6 EID prefixes.
Step 7	**exit-address-family** **Example:** `Router(config-vrf-af)# exit-address-family`	Exits VRF address family configuration mode and returns to VRF configuration mode.
Step 8	**exit** **Example:** `Router(config-vrf)# exit`	Exits VRF configuration mode and enters global configuration mode.

	Command or Action	Purpose
Step 9	**interface** *type number* **Example:** `Router(config)# interface tunnel 191`	Specifies the interface type of tunnel and the interface number and enters interface configuration mode.
Step 10	**vrf forwarding** *vrf-name* **Example:** `Router(config-if)# vrf forwarding lisp`	Associates a VRF instance configured in Step 2 with the tunnel interface configured in Step 9. • When the interface is bound to a VRF, previously configured IP addresses are removed, and the interface is disabled.
Step 11	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 192.168.1.6 255.255.255.252`	Configures an IPv4 address for the tunnel interface.
Step 12	**ipv6 address** *ipv6-address/mask* **Example:** `Router(config-if)# ipv6 address 2001:DB8:ffff::6/64`	Configures an IPv6 address for the tunnel interface.
Step 13	**tunnel source** *interface-type interface-number* **Example:** `Router(config-if)# tunnel source GigabitEthernet 0/0/0`	Configures the tunnel source.
Step 14	**tunnel destination** *ipv4-address* **Example:** `Router(config-if)# tunnel destination 10.10.10.13`	Configures the tunnel destination IPv4 address for the tunnel interface.
Step 15	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.
Step 16	**router lisp** **Example:** `Router(config)# router lisp`	Enters LISP configuration mode (software only).
Step 17	**ipv4 alt-vrf** *vrf-name*	Associates a VRF table with the LISP ALT for IPv4 EIDs.

	Command or Action	Purpose
	Example: `Router(config-router-lisp)# ipv4 alt-vrf lisp`	• In this example, the VRF table named lisp (created in Step 2) is associated with the LISP ALT.
Step 18	**ipv4 proxy-itr** *ipv4-locator* [*ipv6-locator*] **Example:** `Router(config-router-lisp)# ipv4 proxy-itr 10.10.10.11 2001:db8:e000:2::2`	Enables Proxy Ingress Tunnel Router (PITR) functionality for IPv4 EIDs, and specifies the IPv4 and (optionally) the IPv6 RLOCs (local to the PITR) to use when LISP-encapsulating packets to LISP sites.
Step 19	**ipv4 map-cache-limit** *map-cache-limit* **Example:** `Router(config-router-lisp)# ipv4 map-cache-limit 100000`	Specifies the maximum number of IPv4 map-cache entries to be maintained by the PITR. • When the map-cache reaches this limit, existing entries are removed according to the rules described in the command reference guide. (See the *LISP Command Reference* for more details.) • The default map-cache-limit is 10000. In this example, since the device is being configured as a PITR, a larger map-cache limit is configured.
Step 20	**ipv6 alt-vrf** *vrf-name* **Example:** `Router(config-router-lisp)# ipv6 alt-vrf lisp`	Associates a VRF table with the LISP ALT for IPv6 EIDs. • In this example, the VRF table named lisp (created in Step 2) is associated with the LISP ALT.
Step 21	**ipv6 proxy-itr** *ipv6-locator* [*ipv4-locator*] **Example:** `Router(config-router-lisp)# ipv6 proxy-itr 2001:db8:e000:2::2 10.10.10.11`	Enables Proxy Ingress Tunnel Router (PITR) functionality for IPv6 EIDs, and specifies the IPv6 and (optionally) the IPv4 RLOCs (local to the PITR) to use when LISP-encapsulating packets to LISP sites.
Step 22	**ipv6 map-cache-limit** *map-cache-limit* **Example:** `Router(config-router-lisp)# ipv6 map-cache-limit 100000`	Specifies the maximum number of IPv6 map-cache entries to be maintained by the PITR. • When the map-cache reaches this limit, existing entries are removed according to the rules described in the command reference guide. (See the *LISP Command Reference* for more details.) The default map-cache-limit is 10000. In this example, since the device is being configured as a PITR, a larger map-cache limit is configured.

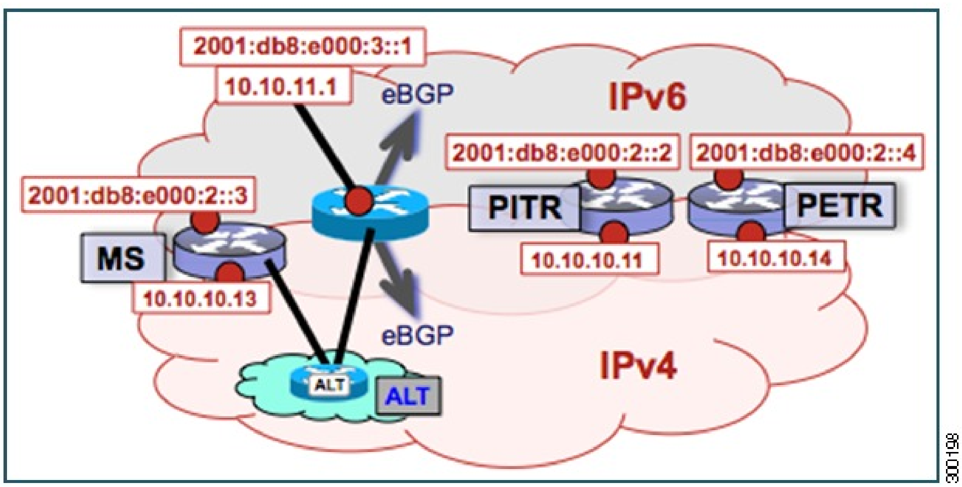
	Command or Action	Purpose
Step 23	**exit** **Example:** `Router(config-router-lisp)# exit`	Exits LISP configuration mode and returns to global configuration mode.
Step 24	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 65015`	Enters router configuration mode for the specified routing process.
Step 25	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] **Example:** `Router(config-router)# address-family ipv4 vrf lisp`	Specifies the IPv4 address family and enters IPv4 address family configuration mode. • The **vrf** keyword and *vrf-name* argument specify the name of the VRF instance to associate with subsequent commands. • In this example, the VRF table named lisp (created in Step 2) is associated with the BGP IPv4 VRF that carries EID prefixes in the LISP ALT.
Step 26	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Router(config-router-af)# neighbor 192.168.1.5 remote-as 65011`	Adds the IP address of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local router.
Step 27	**neighbor** *ip-address* **activate** **Example:** `Router(config-router-af)# neighbor 192.168.1.5 activate`	Enables the neighbor to exchange prefixes for the IPv4 unicast address family.
Step 28	**exit** **Example:** `Router(config-router-af)# exit`	Exits address family configuration mode.
Step 29	**address-family ipv6** [**unicast** | **multicast** | **vrf** *vrf-name*] **Example:** `Router(config-router-af)# address-family ipv6 vrf lisp`	Specifies the IPv6 address family and enters IPv6 address family configuration mode. • The **vrf** keyword and *vrf-name* argument specify the name of the VRF instance to associate with subsequent commands. • In this example, the VRF table named lisp (created in Step 2) is associated with the BGP IPv6 VRF that carries EID prefixes in the LISP ALT.

	Command or Action	Purpose
Step 30	**neighbor** *ip-address* **remote-as** *autonomous-system-number* **Example:** `Router(config-router-af)# neighbor 2001:db8:ffff::5 remote-as 65011`	Adds the IPv6 address of the neighbor in the specified autonomous system to the IPv6 multiprotocol BGP neighbor table of the local router.
Step 31	**neighbor** *ip-address* **activate** **Example:** `Router(config-router-af)# neighbor 2001:db8:ffff::5 activate`	Enables the neighbor to exchange prefixes for the IPv6 unicast address family.
Step 32	**exit** **Example:** `Router(config-router-af)# exit`	Exits address family configuration mode.
Step 33	**exit** **Example:** `Router(config-router)# exit`	Exits router configuration mode.
Step 34	**ip route** *ipv4-prefix next-hop* **Example:** `Router(config)# ip route 0.0.0.0 0.0.0.0 10.10.10.1`	Configures an IPv4 static route. • In this example, a default route to the upstream next hop for all IPv4 destinations is created.
Step 35	**ip route** *ipv4-prefix next-hop* **Example:** `Router(config)# ip route 172.16.0.0 255.255.0.0 Null0 tag 123`	Configures an IPv4 static route. • In this example, a static route is configured to Null0 for the coarse-aggregate IPv4 EID prefix 172.16.0.0/16. This static route is required to ensure proper operation of LISP in querying the mapping system for LISP EIDs. The tag 123 is added to this null route as a reference point for the route map used to permit the advertisement of this coarse aggregate to the upstream ISP BGP peer.
Step 36	**ipv6 route** *ipv6-prefix next-hop* **Example:** `Router(config)# ipv6 route ::/0 2001:db8:e000:2::f0f`	Configures an IPv6 static route. • In this example, a default route to the upstream next hop for all IPv6 destinations is created.
Step 37	**ipv6 route** *ipv6-prefix next-hop*	Configures an IPv6 static route.

	Command or Action	Purpose
	Example: `Router(config)# ipv6 route 2001:db8::/33 Null0 tag 123`	• In this example, a static route is configured to Null0 for the coarse-aggregate IPv6 EID prefix 2001:db8::/33. This is required to ensure proper operation of LISP in querying the mapping system for LISP EIDs. The tag 123 is added to this null route as a handy reference point for the route-map used to permit the advertisement of this coarse-aggregate to the upstream ISP BGP peer.
Step 38	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

Example:

Figure 19: Proxy Ingress Tunnel Router with both an IPv4 and an IPv6 RLOC



The example below shows the full configuration for a PITR includes some basic IP, BGP, and route map configuration not included in the task table for this task:

```
!
hostname PITR
!
no ip domain lookup
ip cef
ipv6 unicast-routing
ipv6 cef
!
interface Loopback0
 no ip address
!
interface Tunnel191
 vrf forwarding lisp
```

```
 ip address 192.168.1.6 255.255.255.252
 ipv6 address 2001:db8:ffff::6/64
 tunnel source GigabitEthernet 0/0/0
 tunnel destination 10.10.10.13
!
interface GigabitEthernet 0/0/0
 description Link to Core (RLOC)
 ip address 10.10.10.11 255.255.255.0
 ipv6 address 2001:db8:e000:2::2/64
!
router lisp
 ipv4 alt-vrf lisp
 ipv4 map-cache-limit 100000
 ipv4 proxy-itr 10.10.10.11 2001:db8:e000:2::2
 ipv6 alt-vrf lisp
 ipv6 map-cache-limit 100000
 ipv6 proxy-itr 2001:db8:e000:2::2 10.10.10.11
 exit
!
router bgp 65015
 bgp asnotation dot
 bgp log-neighbor-changes
 neighbor 10.10.11.1 remote-as 65111
 neighbor 2001:db8:e000:3::1 remote-as 65111
 !
 address-family ipv4
  no synchronization
  redistribute static route-map populate-default
  neighbor 10.10.11.1 activate
  neighbor 10.10.11.1 send-community both
  neighbor 10.10.11.1 route-map dfz-out out
 exit-address-family
 !
 address-family ipv6
  redistribute static route-map populate-default
  neighbor 2001:db8:e000:3::1  activate
  neighbor 2001:db8:e000:3::1  send-community both
  neighbor 2001:db8:e000:3::1  route-map dfz-out out
 exit-address-family
 !
 address-family ipv4 vrf lisp
  no synchronization
  neighbor 192.168.1.5 remote-as 65011
  neighbor 192.168.1.5 activate
 exit-address-family
 !
 address-family ipv6 vrf lisp
  no synchronization
  neighbor 2001:db8:ffff::5 remote-as 65011
  neighbor 2001:db8:ffff::5 activate
 exit-address-family
!
ip bgp-community new-format
ip community-list standard dfz-upstream permit 65100:123
!
ip route 0.0.0.0 0.0.0.0 10.10.10.1
ip route 172.16.0.0 255.255.0.0 Null0 tag 123
!
ipv6 route 2001:db8::/33 Null0 tag 123
ipv6 route ::/0 2001:db8:e000:2::f0f
!
route-map populate-default permit 10
 match tag 123
 set origin igp
 set community 65100:123
!
route-map dfz-out permit 10
 match community dfz-upstream
!
```

Verify and Troubleshoot Locator ID Separation Protocol

Once LISP is configured, you can verify and troubleshoot LISP configuration and operations by following the optional steps in this task. Note that certain verification and troubleshooting steps are specific to certain LISP devices and only apply if configured in your LISP site.

SUMMARY STEPS

1. **enable**
2. **show running-config | section router lisp**
3. **show** [**ip** | **ipv6**] **lisp**
4. **show** [**ip** | **ipv6**] **lisp map-cache**
5. **show** [**ip** | **ipv6**] **lisp database**
6. **show lisp site** [**name** *site-name*]
7. **lig** {[**self** {**ipv4** | **ipv6**}] | {*hostname* | *destination-EID*}}
8. **ping** {*hostname* | *destination-EID*}
9. **clear** [**ip** | **ipv6**] **lisp map-cache**

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode. Enter your password if prompted.

Example:

```
Router> enable
```

Step 2 **show running-config | section router lisp**

The **show running-config | section router lisp** command is useful for quickly verifying the LISP configuration on the device. This command applies to any Cisco IOS LISP device.

The following is sample output from the **show running-config | section router lisp** command when a mulithomed LISP site is configured with IPv4 and IPv6 EID prefixes:

Example:

```
Router# show running-config | section router lisp

router lisp
 database-mapping 172.16.1.0/24 10.1.1.2 priority 1 weight 50
 database-mapping 172.16.1.0/24 10.2.1.2 priority 1 weight 50
 database-mapping 2001:DB8:A::/48 10.1.1.2 priority 1 weight 50
 database-mapping 2001:DB8:A::/48 10.2.1.2 priority 1 weight 50
 ipv4 itr map-resolver 10.10.10.10
 ipv4 itr map-resolver 10.10.30.10
 ipv4 itr
 ipv4 etr map-server 10.10.10.10 key some-key
 ipv4 etr map-server 10.10.30.10 key some-key
 ipv4 etr
 ipv6 use-petr 10.10.10.11
 ipv6 use-petr 10.10.30.11
 ipv6 itr map-resolver 10.10.10.10
```

```
 ipv6 itr map-resolver 10.10.30.10
 ipv6 itr
 ipv6 etr map-server 10.10.10.10 key some-key
 ipv6 etr map-server 10.10.30.10 key some-key
 ipv6 etr
 exit
```

Step 3 **show** [**ip** | **ipv6**] **lisp**

The **show ip lisp** and **show ipv6 lisp** commands are useful for quickly verifying the operational status of LISP as configured on the device, as applicable to the IPv4 and IPv6 address families, respectively. This command applies to any Cisco IOS LISP device.

Example:

The following example shows LISP operational status and IPv4 address family information:

```
Router# show ip lisp

Ingress Tunnel Router (ITR):      enabled
  Egress Tunnel Router (ETR):       enabled
  Proxy-ITR Router (PITR):          disabled
  Proxy-ETR Router (PETR):          disabled
  Map Server (MS):                  disabled
  Map Resolver (MR):                disabled
  Map-Request source:               172.16.1.1
  ITR Map-Resolver(s):              10.10.10.10, 10.10.30.10
  ETR Map-Server(s):                10.10.10.10 (00:00:56), 10.10.30.10 (00:00:12)
  ETR accept mapping data:          disabled, verify disabled
  ETR map-cache TTL:                1d00h
  Locator Status Algorithms:
    RLOC-probe algorithm:           disabled
  Static mappings configured:       0
  Map-cache size/limit:             2/1000
  Map-cache activity check period:  60 secs
  Map-database size:                1
```

Example:

The following example shows LISP operational status and IPv6 address family information:

```
Router# show ip lisp

Ingress Tunnel Router (ITR):      enabled
  Egress Tunnel Router (ETR):       enabled
  Proxy-ITR Router (PITR):          disabled
  Proxy-ETR Router (PETR):          disabled
  Map Server (MS):                  disabled
  Map Resolver (MR):                disabled
  Map-Request source:               2001:DB8:A::1
ITR Map-Resolver(s):                10.10.10.10, 10.10.30.10
  ETR Map-Server(s):                10.10.10.10 (00:00:23), 10.10.30.10 (00:00:40)
  ETR accept mapping data:          disabled, verify disabled
  ETR map-cache TTL:                1d00h
  Locator Status Algorithms:
    RLOC-probe algorithm:           disabled
  Static mappings configured:       0
  Map-cache size/limit:             1/1000
  Map-cache activity check period:  60 secs
  Map-database size:                1
```

Step 4 **show** [**ip** | **ipv6**] **lisp map-cache**

The **show ip lisp map-cache** and **show ipv6 lisp map-cache** commands are useful for quickly verifying the operational status of the map-cache on a device configured as an ITR or PITR, as applicable to the IPv4 and IPv6 address families, respectively. Based on a configuration when a mulithomed LISP site is configured with IPv4 and IPv6 EID prefixes, this example output assumes that a map-cache entry has been received for another site with the IPv4 EID prefix of 172.16.2.0/24 and the IPv6 EID prefix of 2001:db8:b::/48.

Example:

The following example shows IPv4 mapping cache information:

```
Router# show ip lisp map-cache

LISP IPv4 Mapping Cache, 2 entries

0.0.0.0/0, uptime: 02:48:19, expires: never, via static send map-request
  Negative cache entry, action: send-map-request
172.16.2.0/24, uptime: 01:45:24, expires: 22:14:28, via map-reply, complete
  Locator   Uptime    State      Pri/Wgt
  10.0.0.6  01:45:24  up           1/1
```

Example:

The following example shows IPv6 mapping cache information:

```
Router# show ipv6 lisp map-cache

LISP IPv6 Mapping Cache, 2 entries

::/0, uptime: 02:49:39, expires: never, via static send map-request
  Negative cache entry, action: send-map-request
2001:DB8:B::/48, uptime: 00:00:07, expires: 23:59:46, via map-reply, complete
  Locator   Uptime    State      Pri/Wgt
  10.0.0.6  00:00:07  up           1/1
```

Step 5 **show** [**ip** | **ipv6**] **lisp database**

The **show ip lisp database** and **show ipv6 lisp database** commands are useful for quickly verifying the the operational status of the database mapping on a device configured as an ETR, as applicable to the IPv4 and IPv6 address families, respectively. The following example output is based on a configuration when a mulithomed LISP site is configured with IPv4 and IPv6 EID prefixes.

Example:

The following example shows IPv4 mapping database information:

```
Router# show ip lisp database

LISP ETR IPv4 Mapping Database, LSBs: 0x3, 1 entries

172.16.1.0/24
  Locator   Pri/Wgt  Source     State
  10.1.1.2    1/50   cfg-addr   site-self, reachable
  10.2.1.2    1/50   cfg-addr   site-other, report-reachable
```

Example:

The following example shows IPv6 mapping database information:

```
Router# show ipv6 lisp database

LISP ETR IPv6 Mapping Database, LSBs: 0x1, 1 entries

2001:DB8:A::/48
  Locator   Pri/Wgt  Source     State
```

```
  10.1.1.2    1/50   cfg-addr   site-self, reachable
  10.2.1.2    1/50   cfg-addr   site-other, report-reachable
```

Step 6 **show lisp site** [**name** *site-name*]

The **show lisp site** command is useful for quickly verifying the operational status of LISP sites, as configured on a map server. This command applies only to a device configured as a map server.

The following examples are based on configurations where a mulithomed LISP site is configured with both IPv4 and IPv6 EID prefixes:

Example:

```
Router# show lisp site

LISP Site Registration Information

Site Name      Last      Up   Who Last             EID Prefix
               Register       Registered
Site-1         00:00:15  yes  10.1.1.2             172.16.1.0/24
               00:00:11  yes  10.1.1.2             2001:DB8:A::/48
Site-2         00:00:27  yes  10.0.0.6             172.16.2.0/24
               00:00:37  yes  10.0.0.6             2001:DB8:B::/48
```

Example:

```
Router# show lisp site name Site-1

Site name: Site-1
Allowed configured locators: any
Allowed EID-prefixes:
  EID-prefix: 172.16.1.0/24
    First registered:     00:04:51
    Routing table tag:    0
    Origin:               Configuration
    Merge active:         No
    Proxy reply:          No
    TTL:                  1d00h
    Registration errors:
      Authentication failures:   0
      Allowed locators mismatch: 0
    ETR 10.1.1.2, last registered 00:00:01, no proxy-reply, map-notify
                 TTL 1d00h, no merge
      Locator  Local  State      Pri/Wgt
      10.1.1.2  yes   up          1/50
    ETR 10.2.1.2, last registered 00:00:03, no proxy-reply, map-notify
                 TTL 1d00h, merge
      Locator  Local  State      Pri/Wgt
      10.1.1.2 yes    up          1/50
      10.2.1.2 yes    up          1/50
  EID-prefix: 2001:DB8:A::/48
    First registered:     00:04:51
    Routing table tag:    0
    Origin:               Configuration
    Merge active:         No
    Proxy reply:          No
    TTL:                  1d00h
    Registration errors:
      Authentication failures:   0
      Allowed locators mismatch: 0
    ETR 10.1.1.2, last registered 00:00:01, no proxy-reply, map-notify
                 TTL 1d00h, no merge
      Locator  Local  State      Pri/Wgt
      10.1.1.2  yes   up          1/50
    ETR 10.2.1.2, last registered 00:00:03, no proxy-reply, map-notify
                 TTL 1d00h, merge
      Locator  Local  State      Pri/Wgt
      10.1.1.2 yes    up          1/50
```

```
      10.2.1.2 yes    up          1/50
```

Step 7 **lig** {[**self** {**ipv4** | **ipv6**}] | {*hostname* | *destination-EID*}}

The LISP Internet Groper (lig) command is useful for testing the LISP control plane. The **lig** command can be used to query for the indicated destination hostname or EID, or the router's local EID prefix. This command provides a simple means of testing whether a destination EID exists in the LISP mapping database system, or whether your site is registered with the mapping database system. This command is applicable for both the IPv4 and IPv6 address families and applies to any Cisco IOS LISP device that maintains a map-cache (i.e. configured as an ITR or PITR).

The following examples are based on configurations where a mulithomed LISP site is configured with both IPv4 and IPv6 EID prefixes:

Example:

```
Router# lig self ipv4

Mapping information for EID 172.16.1.0 from 10.1.1.2 with RTT 12 msecs
172.16.1.0/24, uptime: 00:00:00, expires: 23:59:52, via map-reply, self
  Locator   Uptime    State      Pri/Wgt
  10.1.1.2  00:00:00  up, self     1/50
  10.2.1.2  00:00:00  up           1/50
```

Example:

```
Router# lig self ipv6

Mapping information for EID 2001:DB8:A:: from 10.0.0.2 with RTT 12 msecs
2001:DB8:A::/48, uptime: 00:00:00, expires: 23:59:52, via map-reply, self
  Locator   Uptime    State      Pri/Wgt
  10.1.1.2  00:00:00  up, self     1/50
  10.2.1.2  00:00:00  up           1/50
```

Example:

```
Router# lig 172.16.2.1

Mapping information for EID 2001:DB8:A:: from 10.0.0.2 with RTT 12 msecs
2001:DB8:A::/48, uptime: 00:00:00, expires: 23:59:52, via map-reply, self
  Locator   Uptime    State      Pri/Wgt
  10.1.1.2  00:00:00  up, self     1/50
  10.2.1.2  00:00:00  up           1/50
```

Example:

```
Router# lig 2001:db8:b::1

Mapping information for EID 172.16.2.1 from 10.0.0.6 with RTT 4 msecs
2001:DB8:B::/48, uptime: 01:52:45, expires: 23:59:52, via map-reply, complete
  Locator   Uptime    State      Pri/Wgt
  10.0.0.6  01:52:45  up           1/1
```

Step 8 **ping** {*hostname* | *destination-EID*}

The **ping** command is useful for testing basic network connectivity and reachability and liveness of a destination EID or RLOC address. It is important to be aware that because LISP uses encapsulation, you should always specify a source address when using **ping**. Never allow the **ping** application to assign its own default source address because there are four possible ways to use **ping** and unless the source address is explicitly named, the wrong address may be used by the application and return erroneous results that complicate operational verification or troubleshooting.

The four possible uses of **ping** are:

- RLOC-to-RLOC—Sends out “echo□? packets natively (no LISP encapsulation) and receives the “echo-reply□? back natively. This use of **ping** can test the underlying network connectivity between locators of various devices, such as between an xTR and a map server or map resolver.
- EID-to-EID—Sends out “echo□? packets with LISP encapsulation and receives the “echo-reply□? back as LISP encapsulated. This use of **ping** can be used to test the LISP data plane (encapsulation) between LISP sites.
- EID-to-RLOC—Sends out “echo□? packets natively (no LISP encapsulation) and receives the "echo-reply" back as LISP encapsulated through a PITR mechanism. This use of **ping** can be used to test the PITR infrastructure.
- RLOC-to-EID - Sends out “echo□? packets with LISP encapsulation and receives the “echo-reply□? back natively (no LISP encapsulation. This use of **ping** can be used to test PETR capabilities.

The **ping** command is applicable to the IPv4 and IPv6 address families, respectively, and can be used on any LISP device but is limited by the LISP device and site configuration. (For example, the ability to do LISP encapsulation requires the device to be configured as either an ITR or PITR.)

The following examples are based on configurations where a mulithomed LISP site is configured with both IPv4 and IPv6 EID prefixes:

Example:

```
Router# ping 172.16.2.1 source 172.16.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.2.1, timeout is 2 seconds:
Packet sent with a source address of 172.16.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/3/8 ms
```

Example:

```
Router# ping 2001:db8:b::1 source 2001:db8:a::1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:B::1, timeout is 2 seconds:
Packet sent with a source address of 2001:DB8:A::1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/3/8 ms
```

Step 9 **clear** [**ip** | **ipv6**] **lisp map-cache**

The **clear ip lisp map-cache** and **clear ipv6 lisp map-cache** commands remove all IPv4 or IPv6 dynamic LISP map-cache entries stored by the router. This command applies to a LISP device that maintains a map-cache (like one configured as an ITR or PITR) and can be useful if trying to quickly verify the operational status of the LISP control plane. Based on a configuration when a mulithomed LISP site is configured with both IPv4 and IPv6 EID prefixes, the following example output assumes that a map-cache entry has been received for another site with the IPv4 EID prefix of 172.16.2.0/24 or an IPv6 EID prefix of 2001:db8:b::/48.

Example:

The following example shows IPv4 mapping cache information, how to clear the mapping cache, and the **show** information after the cache is cleared.

```
Router# show ip lisp map-cache
```

```
LISP IPv4 Mapping Cache, 2 entries

0.0.0.0/0, uptime: 02:48:19, expires: never, via static send map-request
  Negative cache entry, action: send-map-request
172.16.2.0/24, uptime: 01:45:24, expires: 22:14:28, via map-reply, complete
  Locator   Uptime    State      Pri/Wgt
  10.0.0.6  01:45:24  up           1/1

Router# clear ip lisp map-cache

Router# show ip lisp map-cache

LISP IPv4 Mapping Cache, 1 entries

0.0.0.0/0, uptime: 00:00:02, expires: never, via static send map-request
  Negative cache entry, action: send-map-request
```

Example:

The following example shows IPv6 mapping cache information, how to clear the mapping cache, and the **show** information after the cache is cleared.

```
Router# show ipv6 lisp map-cache

LISP IPv6 Mapping Cache, 2 entries

::/0, uptime: 02:49:39, expires: never, via static send map-request
  Negative cache entry, action: send-map-request
2001:DB8:B::/48, uptime: 00:00:07, expires: 23:59:46, via map-reply, complete
  Locator   Uptime    State      Pri/Wgt
  10.0.0.6  00:00:07  up           1/1

Router# clear ip lisp map-cache

Router# show ip lisp map-cache

LISP IPv6 Mapping Cache, 1 entries

::/0, uptime: 00:00:02, expires: never, via static send map-request
  Negative cache entry, action: send-map-request
```

Additional References

The following sections provide references related to the Locator ID Separation Protocol.

Related Documents

Document Title	Location
Cisco IOS LISP Lab Test Configuration Application Note	http://lisp4.cisco.com/lisp_tech.html
Cisco IOS IP Routing: LISP Command Reference	http://www.cisco.com/en/US/docs/ios-xml/ios/iproute_lisp/command/ip-lisp-cr-book.html

Standards

Standard	Title
IANA Address Family Numbers	http://www.iana.org/assignments/address-family-numbers/address-family-numbers.xml

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
draft-ietf-lisp-07	Locator/ID Separation Protocol (LISP) http://tools.ietf.org/html/draft-ietf-lisp-07
draft-ietf-lisp-alt-04	LISP Alternative Topology (LISP+ALT) http://tools.ietf.org/html/draft-ietf-lisp-alt-04
draft-ietf-lisp-interworking-01	Interworking LISP with IPv4 and IPv6 http://tools.ietf.org/html/draft-ietf-lisp-interworking-01
draft-ietf-lisp-lig-00	LISP Internet Groper (LIG) http://tools.ietf.org/html/draft-ietf-lisp-lig-00
draft-ietf-lisp-ms-05	LISP Map Server http://tools.ietf.org/html/draft-ietf-lisp-ms-05

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for LISP

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

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

Table 2: Feature Information for Locator/ID Separation Protocol

Feature Name	Release	Feature Configuration Information
Configure LISP	Cisco IOS Release 15.1(4)M 15.1(1)SY1	Introduces LISP functionality to support ITR, ETR, PITR, PETR, MS, MR, and LISP ALT devices for IPv4 and IPv6 address families on Cisco IOS Release 15.1M&T and later releases.

CHAPTER 3

LISP Shared Model Virtualization

This guide describes how to configure Locator ID Separation Protocol (LISP) shared model virtualization using Software on all LISP-related devices, including the Egress Tunnel Router, Ingress Tunnel Router (ITR), Proxy ETR (PETR), Proxy ITR (PITR), Map Resolver (MR), and Map Server (MS).

LISP implements a new routing architecture that utilizes a "level of indirection" to separate an IP address into two namespaces: Endpoint Identifiers (EIDs), which are assigned to end-hosts, and Routing Locators (RLOCs), which are assigned to devices (primarily routers) that make up the global routing system. Splitting EID and RLOC functions yields several advantages including: improved routing system scalability, multihoming with ingress traffic engineering; efficient IPv6 Transition support; high-scale virtualization/multitenancy support; data center/VM-mobility support, including session persistence across mobility events; and seamless mobile node support.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

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

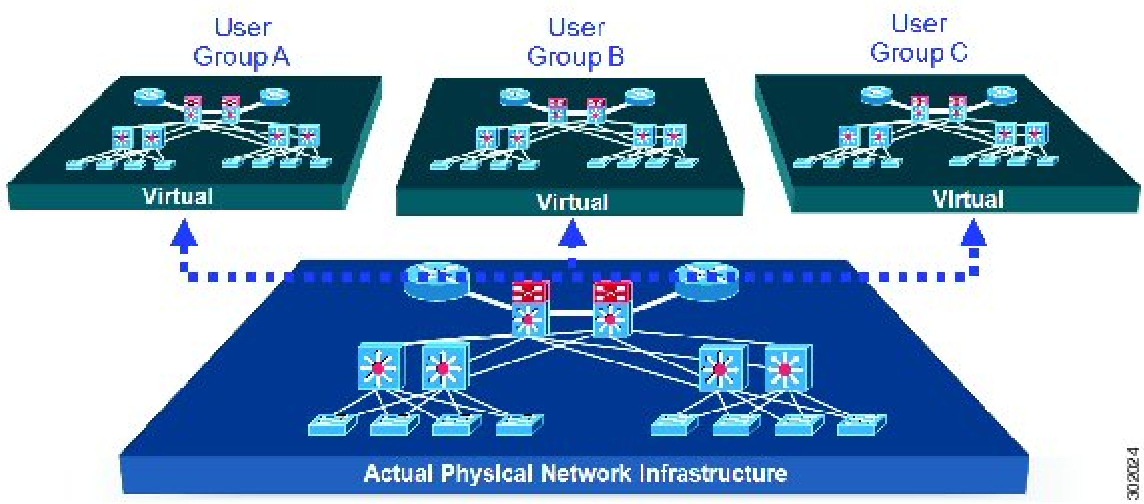
Information About LISP Shared Model Virtualization

Overview of LISP Virtualization

Deploying physical network infrastructure requires both capital investments for hardware, as well as manpower investments for installation and operational management support. When distinct user groups within an organization desire to control their own networks, it rarely makes economic sense for these user groups to deploy and manage separate physical networks. Physical plants are rarely utilized to their fullest, resulting in stranded capacity (bandwidth, processor, memory, etc.). In addition, the power, rack space, and cooling needs to physical plants do not satisfy modern "green" requirements. Network virtualization offers the opportunity to satisfy organizational needs, while efficiently utilizing physical assets.

The purpose of network virtualization, as shown in the figure below, is to create multiple, logically separated topologies across one common physical infrastructure.

Figure 20: LISP Deployment Environment



When considering the deployment of a virtualized network environment, take into account both the device and the path level.

Device Level Virtualization

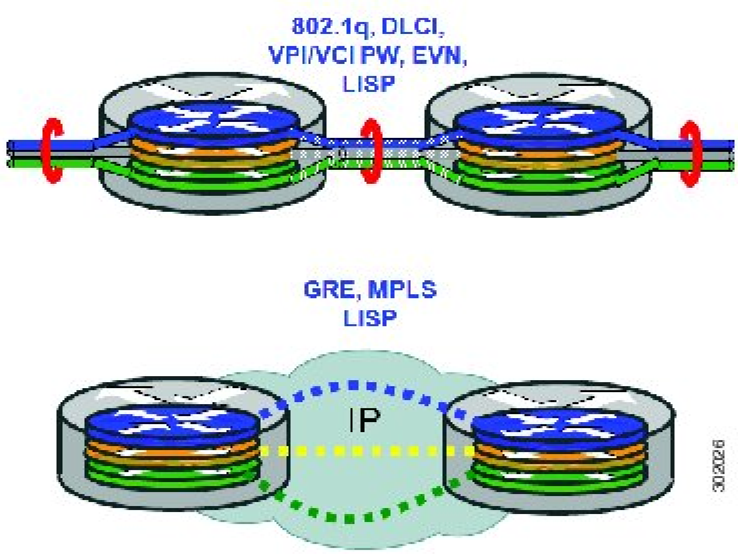
Virtualization at the device level entails the use of the virtual routing and forwarding (VRF) to create multiple instances of Layer 3 routing tables, as illustrated in the figure below. VRFs provide segmentation across IP addresses, allowing for overlapped address space and traffic separation. Separate routing, QoS, security, and management policies can be applied to each VRF instance. An IGP or EGP routing process is typically enabled within a VFR, just as it would be in the global (default) routing table. As described in detail below, LISP binds VRFs to instance IDs for similar purposes.

Figure 21: Device Level Virtualization

Path Level Virtualization

VRF table separation is maintained across network paths using any number of traditional mechanisms, as illustrated in the figure below. Single-hop path segmentation (hop-by-hop) is typically accomplished by techniques such as 802.1q VLANs, VPI/VCI PW, or EVN. LISP can also be used. Traditional multi-hop mechanisms include MPLS and GRE tunnels. As described in detail below, LISP binds VRFs to instance IDs (IIDs), and then these IIDs are included in the LISP header to provide data plane (traffic flow) separation for single or multihop needs.

Figure 22: Path Level Virtualization



LISP Virtualization at the Device Level

Recalling that LISP implements Locator ID separation and, in so doing, creates two namespaces (EIDs and RLOCs), it is easy to see that LISP virtualization can consider both EID and RLOC namespaces for virtualization. That is, either or both can be virtualized.

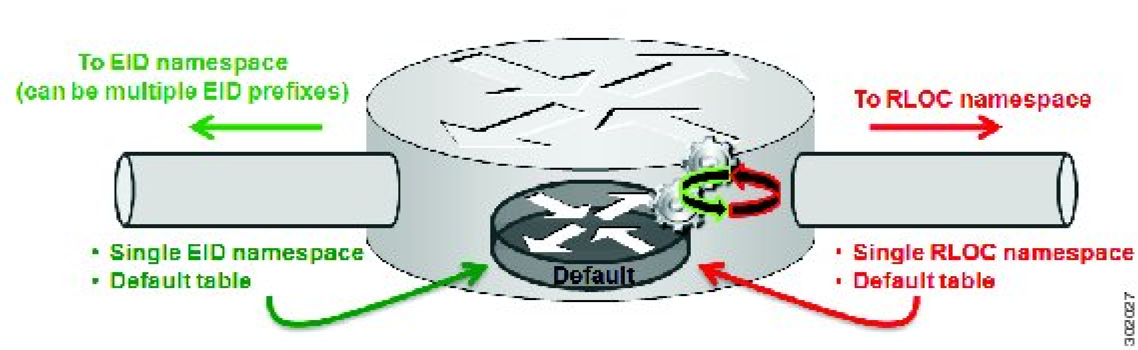
- EID virtualization—Enabled by binding a LISP instance ID to an EID VRF. Instance IDs are numerical tags defined in the LISP canonical address format (LCAF) draft, and are used to maintain address space segmentation in both the control plane and data plane.
- RLOC virtualization—Tying locator addresses and associated mapping services to the specific VRF within which they are reachable enables RLOC virtualization.

Because LISP considers virtualization of both EID and RLOC namespaces, two models of operation are defined: shared model and parallel model. For completeness, the discussions below begin first with a review of the default (non-virtualized) model of LISP, and then cover the details of shared and parallel models.

Default (Non-Virtualized) LISP Model

By default, LISP is not virtualized in either EID space or RLOC space. That is, unless otherwise configured, both EID and RLOC addresses are resolved in the default (global) routing table. This concept is illustrated in the figure below.

Figure 23: Default (Non-Virtualized) LISP Model (Resolves Both EID and RLOC Addresses in the Default (Global) Routing Table.

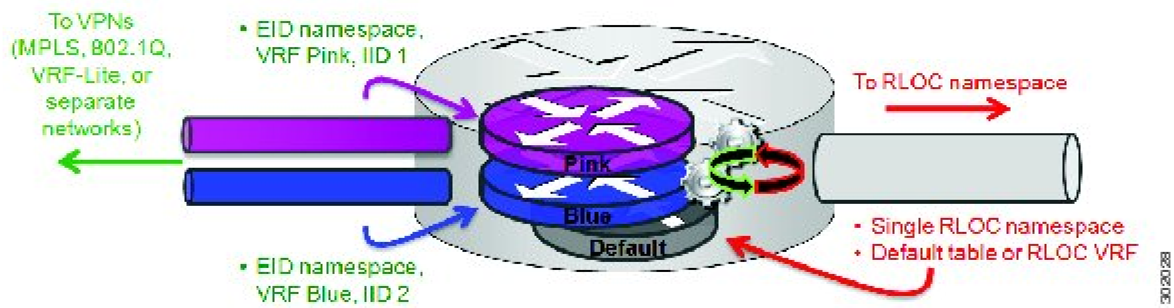


As shown in the figure above, both EID and RLOC addresses are resolved in the default table. The mapping system must also be reachable via the default table. This default model can be thought of as a single instantiation of the parallel model of LISP virtualization where EID and RLOC addresses are within the same namespace such as is the case in this default table.

LISP Shared Model Virtualization

LISP shared model virtualized EID space is created by binding VRFs associated with an EID space to Instance IDs. A common, shared locator space is used by all virtualized EIDs. This concept is illustrated in the figure below.

Figure 24: LISP shared model virtualization resolves EIDs within VRFs tied to Instance IDs. RLOC addresses are resolved in a common (shared) address space. The default (global) routing table is shown as the shared space.

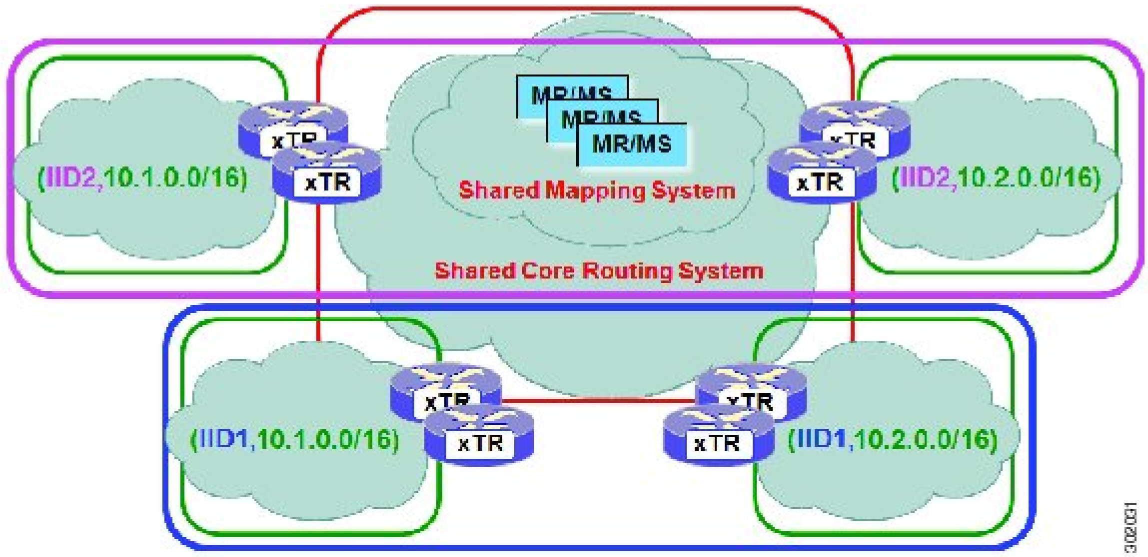


As shown in the figure above, EID space is virtualized through its association with VRFs, and these VRFs are tied to LISP Instance IDs to segment the control plane and data plane in LISP. A common, shared locator space, the default (global) table as shown in the figure above, is used to resolve RLOC addresses for all virtualized EIDs. The mapping system must also be reachable via the common locator space.

LISP Shared Model Virtualization Architecture

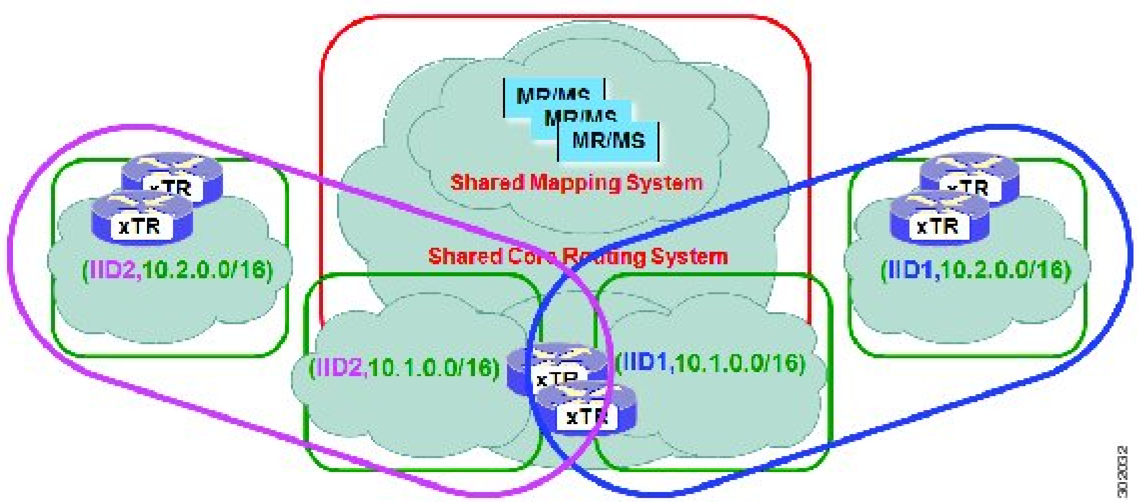
Architecturally, LISP shared model virtualization can be deployed in single or multitenancy configurations. In the shared model single tenancy case, xTRs are dedicated to a customer but share infrastructure with other customers. Each customer and all sites associated with it use the same instance ID and are part of a VPN using their own EID namespace as shown in the figure below.

Figure 25: In a LISP shared model single tenancy use case, customers use their own xTRs and a shared common core network and mapping system. LISP instance IDs segment the LISP data plane and control plane.



In the shared model multitenancy case, a set of xTRs is shared (virtualized) among multiple customers. These customers also share a common infrastructure with other single and multitenant customers. Each customer and all sites associated with it use the same instance ID and are part of a VPN using their own EID namespace as shown in the figure below.

Figure 26: In a LISP shared model multitenancy use case, customer's use shared xTRs and a shared common core network and mapping system. LISP instance IDs segment the LISP data plane and control plane.



LISP Shared Model Virtualization Implementation Considerations and Caveats

When LISP Shared Model is implemented, several important considerations and caveats are important. Instance IDs must be unique to an EID VRF. Review the example below:

```
xTR-1(config)# vrf definition alpha
xTR-1(config-vrf)# address-family ipv4
xTR-1(config-vrf-af)# exit
xTR-1(config)# vrf definition beta
xTR-1(config-vrf)# address-family ipv4
xTR-1(config-vrf-af)# exit
xTR-1(config-vrf)# exit
xTR-1(config)# router lisp
xTR-1(config-router-lisp)# eid-table vrf alpha instance-id 101
xTR-1(config-router-lisp-eid-table)# exit
xTR-1(config-router-lisp)# eid-table vrf beta instance-id 101
Instance ID 101 is bound to the vrf alpha EID table.
```

In the above example, two EID VRFs are created: alpha and beta. Under the **router lisp** command, an EID table VRF named alpha is specified and associated with the instance ID 101. Next, an EID table VRF named beta is specified and also associated with the instance ID 101. As indicated by the router, this is not permissible since instance ID 101 is already associated with the EID VRF named alpha. That is, you cannot connect the same instance-id to more than one EID VRF.

How to Configure LISP Shared Model Virtualization

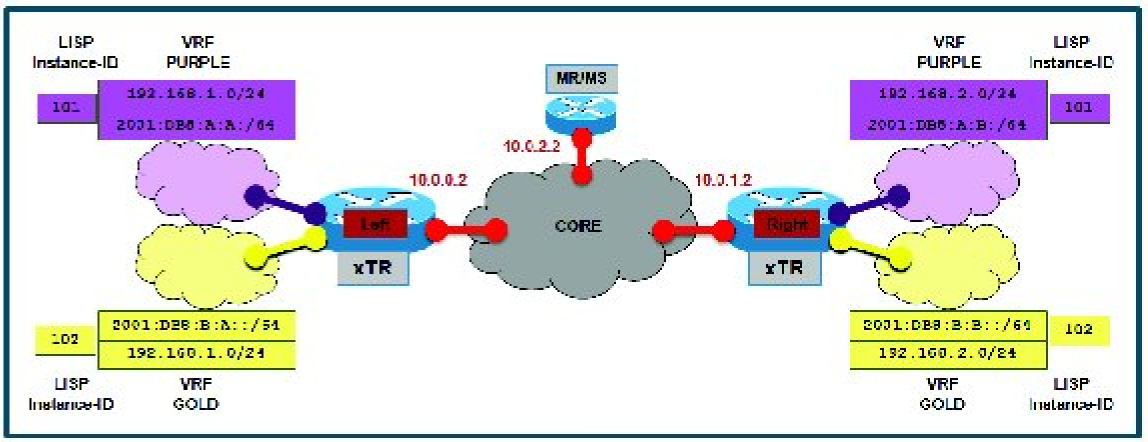
Configure Simple LISP Shared Model Virtualization

Perform this task to enable and configure LISP ITR/ETR (xTR) functionality with LISP map server and map resolver to implement LISP shared model virtualization. This LISP shared model reference configuration is for a very simple two-site LISP topology, including xTRs and an MS/MR.

The configuration implemented in this task and illustrated in the figure below shows a basic LISP shared model virtualization solution. In this example, two LISP sites are deployed, each containing two VRFs:

PURPLE and GOLD. LISP is used to provide virtualized connectivity between these two sites across a common IPv4 core, while maintaining address separation between the two VRFs.

Figure 27: Simple LISP Site with virtualized IPv4 and IPv6 EIDs and a shared IPv4 core



Each LISP Site uses a single edge router configured as both an ITR and ETR (xTR), with a single connection to its upstream provider. The RLOC is IPv4, and IPv4 and IPv6 EID prefixes are configured. Each LISP site registers to a map server/map resolver (MS/MR) device located in the network core within the shared RLOC address space. The topology used in this most basic LISP configuration is shown in the figure above.

The components illustrated in the topology shown in the figure above are described below:

- **LISP site:**
  - The CPE functions as a LISP ITR and ETR (xTR).
  - Both LISP xTRs have two VRFs: GOLD and PURPLE, with each VRF containing both IPv4 and IPv6 EID-prefixes, as shown in the figure above. Note the overlapping prefixes, used for illustration purposes. A LISP instance-id is used to maintain separation between two VRFs. Note that in this example, the share key is configured "per-site" and not "per-VRF." (Case 2 illustrates a configuration where the shared key is per-VPN.)
  - Each LISP xTR has a single RLOC connection to a shared IPv4 core network.

- **Mapping system:**
  - One map server/map resolver system is shown in the figure above and assumed available for the LISP xTR to register to. The MS/MR has an IPv4 RLOC address of 10.0.2.2, within the shared IPv4 core.
  - The map server site configurations are virtualized using LISP instance-ids to maintain separation between the two VRFs.

Perform the steps in this task (once through for each xTR in the LISP site) to enable and configure LISP ITR and ETR (xTR) functionality when using a LISP map-server and map-resolver for mapping services. The example configurations at the end of this task show the full configuration for two xTRs (xTR1 and xTR2).

Before You Begin

The configuration below assumes that the referenced VRFs were created using the **vrf definition** command.

SUMMARY STEPS

1. **configure terminal**
2. **router lisp**
3. **eid-table vrf***vrf-name* **instance-id** *instance-id*
4. Do one of the following:
    - **database-mapping** *EID-prefix/prefix-length locator* **priority** *priority* **weight** *weight*
    - **database-mapping** *EID-prefix/prefix-length locator* **priority** *priority* **weight** *weight*
5. Repeat Step 4 until all EID-to-RLOC mappings for the LISP site are configured.
6. **exit**
7. **ipv4 itr**
8. **ipv4 etr**
9. **ipv4 itr map-resolver** *map-resolver-address*
10. **ipv4 etr map-server** *map-server-address* **key** *key-type authentication-key*
11. **ipv6 itr**
12. **ipv6 etr**
13. **ipv6 itr map-resolver** *map-resolver-address*
14. **ipv6 etr map-server** *map-server-address* **key** *key-type authentication-key*
15. **exit**
16. **ip route** *ipv4-prefix next-hop*
17. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 2	**router lisp** **Example:** `Router(config)# router lisp`	Enters LISP configuration mode (software only).

	Command or Action	Purpose
Step 3	**eid-table vrf***vrf-name* **instance-id** *instance-id* **Example:** `Router(config-router-lisp)# eid-table vrf GOLD instance-id 102`	Configures an association between a VRF table and a LISP instance ID, and enters eid-table configuration submode. • In this example, the VRF table GOLD and instance-id 102 are associated together.

Step 4	Do one of the following:
	
	• **database-mapping** *EID-prefix/prefix-length locator* **priority** *priority* **weight** *weight*
	
	• **database-mapping** *EID-prefix/prefix-length locator* **priority** *priority* **weight** *weight*
	
	Example:
	
	`Router(config-router-lisp-eid-table)# database-mapping 192.168.1.0/24 10.0.0.2 priority 1 weight 100`
	
	Example:
	
	`Router(config-router-lisp-eid-table)# database-mapping 192.168.1.0/24 ipv4-interface Ethernet0/0 priority 1 weight 100`
		Configures an EID-to-RLOC mapping relationship and its associated traffic policy for this LISP site.
		
		• In the first example, a single IPv4 EID prefix, 192.168.1.0/24, is being associated with the single IPv4 RLOC 10.0.0.2.
		
		• In the second example, the alternative configuration shows the use of the dynamic interface form of the **database-mapping** command. This form is useful when the RLOC address is obtained dynamically, such as via DHCP.

Step 5	Repeat Step 4 until all EID-to-RLOC mappings for the LISP site are configured. **Example:** `Router(config-router-lisp-eid-table)# database-mapping 2001:db8:b:a::/64 10.0.0.2 priority 1 weight 100`	Configures an EID-to-RLOC mapping relationship and its associated traffic policy for this LISP site.
Step 6	**exit** **Example:** `Router(config-router-lisp-eid-table)# exit`	Exits eid-table configuration submode and returns to LISP configuration mode.
Step 7	**ipv4 itr** **Example:** `Router(config-router-lisp)# ipv4 itr`	Enables LISP ITR functionality for the IPv4 address family.

	Command or Action	Purpose
Step 8	**ipv4 etr** **Example:** `Router(config-router-lisp)# ipv4 etr`	Enables LISP ETR functionality for the IPv4 address family.
Step 9	**ipv4 itr map-resolver** *map-resolver-address* **Example:** `Router(config-router-lisp)# ipv4 itr map-resolver 10.0.2.2`	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. • The locator address of the map resolver may be an IPv4 or IPv6 address. In this example, because each xTR has only IPv4 RLOC connectivity, the map resolver is reachable using its IPv4 locator address. (See the *LISP Command Reference Guide* for more details.) **Note** Up to two map resolvers may be configured if multiple map resolvers are available. (See the *LISP Command Reference Guide* for more details.)
Step 10	**ipv4 etr map-server** *map-server-address* **key** *key-type authentication-key* **Example:** `Router(config-router-lisp)# ipv4 etr map-server 10.0.2.2 key 0 Left-key`	Configures a locator address for the LISP map server and an authentication key for which this router, acting as an IPv4 LISP ETR, will use to register with the LISP mapping system. • The map server must be configured with EID prefixes and instance IDs matching those configured on this ETR and with an identical authentication key. **Note** The locator address of the map server may be an IPv4 or IPv6 address. In this example, because each xTR has only IPv4 RLOC connectivity, the map-server is reachable using its IPv4 locator addresses. (See the *LISP Command Reference Guide* for more details.)
Step 11	**ipv6 itr** **Example:** `Router(config-router-lisp)# ipv6 itr`	Enables LISP ITR functionality for the IPv6 address family.
Step 12	**ipv6 etr** **Example:** `Router(config-router-lisp)# ipv6 etr`	Enables LISP ETR functionality for the IPv6 address family.
Step 13	**ipv6 itr map-resolver** *map-resolver-address* **Example:** `Router(config-router-lisp)# ipv6 itr map-resolver 10.0.2.2`	Configures a locator address for the LISP map resolver to which this router will send map request messages for IPv6 EID-to-RLOC mapping resolutions. • The locator address of the map resolver may be an IPv4 or IPv6 address. In this example, because each xTR has only IPv4 RLOC connectivity, the map-resolver is reachable using its IPv4 locator

	Command or Action	Purpose
		addresses. (See the *LISP Command Reference Guide* for more details.) **Note** Up to two map resolvers may be configured if multiple map resolvers are available. (See the *LISP Command Reference Guide* for more details.)
Step 14	**ipv6 etr map-server** *map-server-address* **key** *key-type authentication-key* **Example:** `Router(config-router-lisp)# ipv6 etr map-server 10.0.2.2 key 0 Left-key`	Configures a locator address for the LISP map-server and an authentication key that this router, acting as an IPv6 LISP ETR, will use to register to the LISP mapping system. • The map-server must be configured with EID prefixes and instance IDs matching those configured on this ETR and with an identical authentication key. **Note** The locator address of the map-server may be an IPv4 or IPv6 address. In this example, because each xTR has only IPv4 RLOC connectivity, the map-server is reachable using its IPv4 locator addresses. (See the *LISP Command Reference Guide* for more details.)
Step 15	**exit** **Example:** `Router(config-router-lisp)# exit`	Exits LISP configuration mode and returns to global configuration mode.

Step 16	**ip route** *ipv4-prefix next-hop*
	
	Example:
	
	`Router(config)# ip route 0.0.0.0 0.0.0.0 10.0.0.1`
		Configures a default route to the upstream next hop for all IPv4 destinations.
		
		• All IPv4 EID-sourced packets destined to both LISP and non-LISP sites are forwarded in one of two ways:
		  • LISP-encapsulated to a LISP site when traffic is LISP-to-LISP
		  • natively forwarded when traffic is LISP-to-non-LISP
		
		• Packets are deemed to be a candidate for LISP encapsulation when they are sourced from a LISP EID and the destination matches one of the following entries:
		  • a current map-cache entry
		  • a default route with a legitimate next-hop
		  • no route at all
		
		In this configuration example, because the xTR has IPv4 RLOC connectivity, a default route to the upstream SP is used for all IPv4 packets to support LISP processing.

	Command or Action	Purpose
Step 17	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

Example:

The examples below show the complete configuration for the LISP topology illustrated in the figure shown above the task steps and follows the examples in the steps in this task. On the xTRs, the VRFs and EID prefixes are assumed to be attached to VLANs configured on the devices.

Example configuration for the Left xTR:

```
hostname Left-xTR
!
ipv6 unicast-routing
!
vrf definition PURPLE
 address-family ipv4
 exit
 address-family ipv6
 exit
!
vrf definition GOLD
 address-family ipv4
 exit
 address-family ipv6
 exit
!
interface Ethernet0/0
 ip address 10.0.0.2 255.255.255.0
!
interface Ethernet1/0.1
 encapsulation dot1q 101
 vrf forwarding PURPLE
 ip address 192.168.1.1 255.255.255.0
 ipv6 address 2001:DB8:A:A::1/64
!
interface Ethernet1/0.2
 encapsulation dot1q 102
 vrf forwarding GOLD
 ip address 192.168.1.1 255.255.255.0
 ipv6 address 2001:DB8:B:A::1/64
!
router lisp
 eid-table vrf PURPLE instance-id 101
  database-mapping 192.168.1.0/24 10.0.0.2 priority 1 weight 1
  database-mapping 2001:DB8:A:A::/64 10.0.0.2 priority 1 weight 1
 eid-table vrf GOLD instance-id 102
  database-mapping 192.168.1.0/24 10.0.0.2 priority 1 weight 1
  database-mapping 2001:DB8:B:A::/64 10.0.0.2 priority 1 weight 1
 exit
 !
 ipv4 itr map-resolver 10.0.2.2
 ipv4 itr
 ipv4 etr map-server 10.0.2.2 key Left-key
 ipv4 etr
 ipv6 itr map-resolver 10.0.2.2
 ipv6 itr
 ipv6 etr map-server 10.0.2.2 key Left-key
```

```
 ipv6 etr
 exit
!
ip route 0.0.0.0 0.0.0.0 10.0.0.1
!
```

Example configuration for Right xTR:

```
hostname Right-xTR
!
ipv6 unicast-routing
!
vrf definition PURPLE
 address-family ipv4
 exit
 address-family ipv6
 exit
!
vrf definition GOLD
 address-family ipv4
 exit
 address-family ipv6
 exit
!
interface Ethernet0/0
 ip address 10.0.1.2 255.255.255.0
!
interface Ethernet1/0.1
 encapsulation dot1q 101
 vrf forwarding PURPLE
 ip address 192.168.2.1 255.255.255.0
 ipv6 address 2001:DB8:A:B::1/64
!
interface Ethernet1/0.2
 encapsulation dot1q 102
 vrf forwarding GOLD
 ip address 192.168.2.1 255.255.255.0
 ipv6 address 2001:DB8:B:B::1/64
!
router lisp
 eid-table vrf PURPLE instance-id 101
  database-mapping 192.168.2.0/24 10.0.1.2 priority 1 weight 1
  database-mapping 2001:DB8:A:B::/64 10.0.1.2 priority 1 weight 1
 eid-table vrf GOLD instance-id 102
  database-mapping 192.168.2.0/24 10.0.1.2 priority 1 weight 1
  database-mapping 2001:DB8:B:B::/64 10.0.1.2 priority 1 weight 1
 exit
 !
 ipv4 itr map-resolver 10.0.2.2
 ipv4 itr
 ipv4 etr map-server 10.0.2.2 key Right-key
 ipv4 etr
 ipv6 itr map-resolver 10.0.2.2
 ipv6 itr
 ipv6 etr map-server 10.0.2.2 key Right-key
 ipv6 etr
 exit
!
ip route 0.0.0.0 0.0.0.0 10.0.1.1
!
```

Configuring a Private LISP Mapping System for LISP Shared Model Virtualization

Perform this task to configure and enable standalone LISP map server/map resolver functionality for LISP shared model virtualization. In this task, a Cisco router is configured as a standalone map server/map resolver (MR/MS) for a private LISP mapping system. Because the MR/MS is configured as a stand-alone device, it has no need for LISP Alternate Logical Topology (ALT) connectivity. All relevant LISP sites must be configured to register with this map server so that this map server has full knowledge of all registered EID Prefixes within the (assumed) private LISP system.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router lisp**
4. **site** *site-name*
5. **authentication-key** [*key-type*] *authentication-key*
6. **eid-prefix instance-id** *instance-id EID-prefix*
7. **eid-prefix instance-id** *instance-id EID-prefix*
8. **exit**
9. **ipv4 map-resolver**
10. **ipv4 map-server**
11. **ipv6 map-resolver**
12. **ipv6 map-server**
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router lisp** **Example:** `Router(config)# router lisp`	Enters LISP configuration mode (IOS only).
Step 4	**site** *site-name* **Example:** `Router(config-router-lisp)# site Left`	Specifies a LISP site named Left and enters LISP site configuration mode. **Note** 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 EID-prefix or prefixes with an authentication key and other site-related mechanisms.

	Command or Action	Purpose
Step 5	**authentication-key** [*key-type*] *authentication-key* **Example:** `Router(config-router-lisp-site)# authentication-key 0 Left-key`	Configures the password used to create the SHA-2 HMAC hash for authenticating the map register messages sent by an ETR when registering to the map server. **Note** The LISP ETR must be configured with an identical authentication key as well as matching EID prefixes and instance IDs.
Step 6	**eid-prefix instance-id** *instance-id EID-prefix* **Example:** `Router(config-router-lisp-site)# eid-prefix instance-id 102 192.168.1.0/24`	Configures an EID prefix and instance ID that are allowed in a map register message sent by an ETR when registering to this map server. Repeat this step as necessary to configure additional EID prefixes under this LISP site. • In this example, the IPv4 EID prefix 192.168.1.0/24 and instance ID 102 are associated together. To complete this task, an IPv6 EID prefix is required.
Step 7	**eid-prefix instance-id** *instance-id EID-prefix* **Example:** `Router(config-router-lisp-site)# eid-prefix instance-id 102 2001:db8:a:b::/64`	Configures an EID prefix and instance ID that are allowed in a map register message sent by an ETR when registering to this map server. • In this example, the IPv6 EID prefix 2001:db8:a:b::/64 and instance ID 102 are associated together.
Step 8	**exit** **Example:** `Router(config-router-lisp-site)# exit`	Exits LISP site configuration mode and returns to LISP configuration mode.
Step 9	**ipv4 map-resolver** **Example:** `Router(config-router-lisp)# ipv4 map-resolver`	Enables LISP map resolver functionality for EIDs in the IPv4 address family.
Step 10	**ipv4 map-server** **Example:** `Router(config-router-lisp)# ipv4 map-server`	Enables LISP map server functionality for EIDs in the IPv4 address family.
Step 11	**ipv6 map-resolver** **Example:** `Router(config-router-lisp)# ipv6 map-resolver`	Enables LISP map resolver functionality for EIDs in the IPv6 address family.

	Command or Action	Purpose
Step 12	**ipv6 map-server** **Example:** `Router(config-router-lisp)# ipv6 map-server`	Enables LISP map server functionality for EIDs in the IPv6 address family.
Step 13	**end** **Example:** `Router(config-router-lisp)# end`	Exits LISP configuration mode and returns to privileged EXEC mode.

Example:

Example configuration for the map server/map resolver.

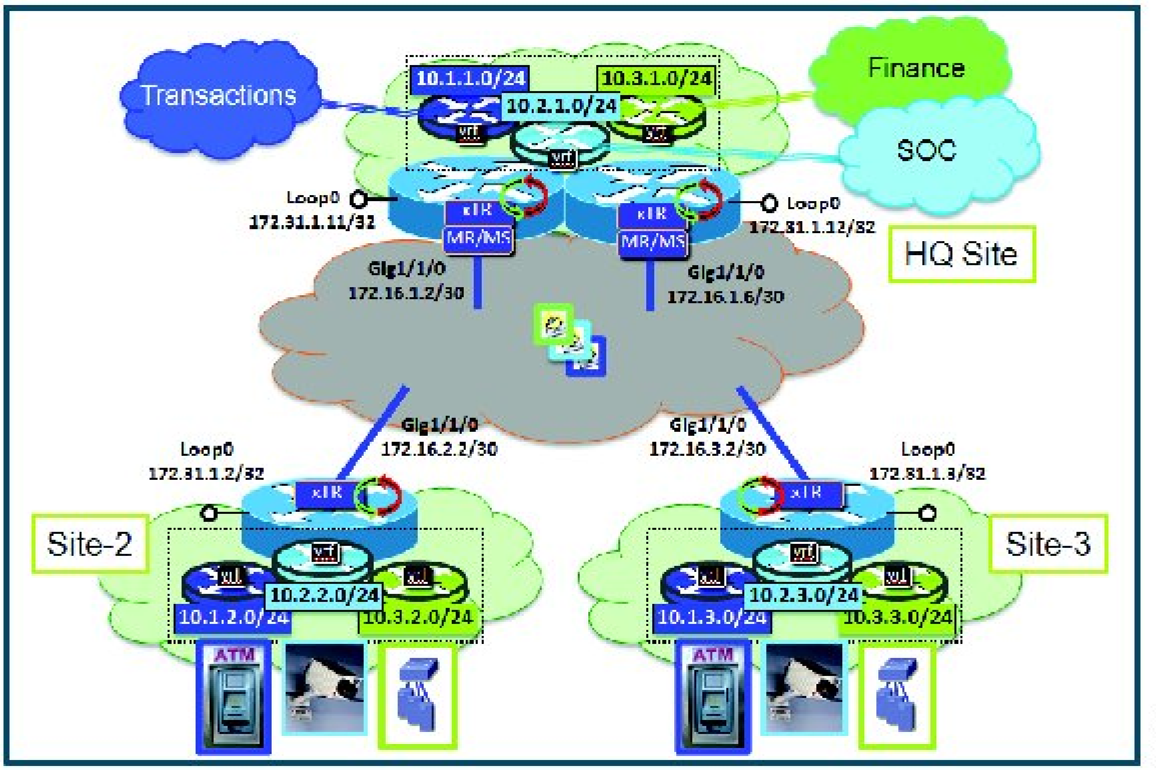
```
hostname MSMR
!
interface Ethernet0/0
 ip address 10.0.2.2 255.255.255.0
!
 router lisp
  !
  site Left
   authentication-key Left-key
   eid-prefix instance-id 101 192.168.1.0/24
   eid-prefix instance-id 101 2001:DB8:A:A::/64
   eid-prefix instance-id 102 192.168.1.0/24
   eid-prefix instance-id 102 2001:DB8:B:A::/64
   exit
  !
  site Right
   authentication-key Right-key
   eid-prefix instance-id 101 192.168.2.0/24
   eid-prefix instance-id 101 2001:DB8:A:B::/64
   eid-prefix instance-id 102 192.168.2.0/24
   eid-prefix instance-id 102 2001:DB8:B:B::/64
   exit
  !
  ipv4 map-server
  ipv4 map-resolver
  ipv6 map-server
  ipv6 map-resolver
  exit
 !
 ip route 0.0.0.0 0.0.0.0 10.0.2.1
```

Configure Large-Scale LISP Shared Model Virtualization

Perform this task to enable and configure LISP ITR/ETR (xTR) functionality with LISP map server and map resolver to implement LISP shared model virtualization. This LISP shared model reference configuration is for a large-scale, multiple-site LISP topology, including xTRs and multiple MS/MRs.

The configuration demonstrated in this task shows a more complex, larger scale LISP virtualization solution. In this task, an enterprise is deploying LISP Shared Model where EID space is virtualized over a shared, common core network. A subset of their entire network is illustrated in Figure 12. In this figure, three sites are shown: a multihomed "Headquarters" (HQ) site, and two remote office sites. The HQ site routers are deployed as xTRs and also as map resolver/map servers. The remote site routers only act as xTRs, and use the MS/MRs at the HQ site for LISP control plane support.

Figure 28: Large Scale LISP Site with Virtualized IPv4 EIDs and a Shared IPv4 Core



The components illustrated in the topology shown in the figure above are described below:

- **LISP site:**
  - Each CPE router functions as a LISP ITR and ETR (xTR), as well as a Map-Server/Map-Resolver (MS/MR).
  - Both LISP xTRs have three VRFs: TRANS (for transactions), SOC (for security operations), and FIN (for financials). Each VRF contains only IPv4 EID-prefixes. Note that no overlapping prefixes are used, but segmentation between each VRF by LISP instance-ids makes this possible. Also note that in this example, the separate authentication key is configured “per-vrf□? and not “per-site.□? This affects both the xTR and MS configurations.
  - The HQ LISP Site is multi-homed to the shared IPv4 core, but each xTR at the HQ site has a single RLOC.
  - Each CPE also functions as an MS/MR to which the HQ and Remote LISP sites can register.
  - The map server site configurations are virtualized using LISP instance IDs to maintain separation between the three VRFs.

- **LISP remote sites:**
  - Each remote site CPE router functions as a LISP ITR and ETR (xTR).
  - Each LISP xTRs has the same three VRFs as the HQ Site: TRANS, SOC, and FIN. Each VRF contains only IPv4 EID-prefixes.
  - Each remote site LISP xTR has a single RLOC connection to a shared IPv4 core network.

Before You Begin

The configuration below assumes that the referenced VRFs were created using the **vrf definition** command.

SUMMARY STEPS

1. **configure terminal**
2. **router lisp**
3. **site** *site-name*
4. **authentication-key** [*key-type*] *authentication-key*
5. **eid-prefix instance-id** *instance-id EID-prefix/prefix-length* **accept-more-specifics**
6. **exit**
7. Repeat steps 3 through 6 for each LISP site to be configured.
8. **ipv4 map-resolver**
9. **ipv4 map-server**
10. **eid-table vrf***vrf-name* **instance-id** *instance-id*
11. **database-mapping** *EID-prefix/prefix-length locator* **priority** *priority* **weight** *weight*
12. Repeat Step 11 until all EID-to-RLOC mappings within this eid-table vrf and instance ID for the LISP site are configured.
13. **ipv4 etr map-server** *map-server-address* **key** *key-type authentication-key*
14. Repeat Step 13 to configure another locator address for the same LISP map server
15. **exit**
16. **ipv4 itr map-resolver** *map-resolver-address*
17. Repeat Step 16 to configure another locator address for the LISP map resolver
18. **ipv4 itr**
19. **ipv4 etr**
20. **exit**
21. **ip route** *ipv4-prefix next-hop*
22. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 2	**router lisp** **Example:** `Router(config)# router lisp`	Enters LISP configuration mode (software only).
Step 3	**site** *site-name* **Example:** `Router(config-router-lisp)# site TRANS`	Specifies a LISP site named TRANS and enters LISP site configuration mode. **Note** 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 EID-prefix or prefixes with an authentication key and other site-related mechanisms.
Step 4	**authentication-key** [*key-type*] *authentication-key* **Example:** `Router(config-router-lisp-site)# authentication-key 0 TRANS-key`	Configures the password used to create the SHA-2 HMAC hash for authenticating the map register messages sent by an ETR when registering to the map server. **Note** The LISP ETR must be configured with an identical authentication key as well as matching EID prefixes and instance IDs.
Step 5	**eid-prefix instance-id** *instance-id EID-prefix/prefix-length* **accept-more-specifics** **Example:** `Router(config-router-lisp-site)# eid-prefix instance-id 1 10.1.0.0/16 accept-more-specifics`	Configures an EID prefix and instance ID that are allowed in a map register message sent by an ETR when registering to this map server. Repeat this step as necessary to configure additional EID prefixes under this LISP site. • In the example, EID-prefix 10.1.0.0/16 and instance-id 1 are associated together. The EID-prefix 10.1.0.0/16 is assumed to be an aggregate covering all TRANS EID-prefixes at all LISP Sites. The keyword **accept-more-specifics** is needed in this case to allow each site to register its more-specific EID-prefix contained within that aggregate. If aggregation is not possible, simply enter all EID-prefixes integrated within instance-id 1.
Step 6	**exit** **Example:** `Router(config-router-lisp-site)# exit`	Exits LISP site configuration mode and returns to LISP configuration mode.

	Command or Action	Purpose
Step 7	Repeat steps 3 through 6 for each LISP site to be configured.	In this example, steps 3 through 6 would be repeated for the site SOC and FIN as illustrated in the complete configuration example at the end of this task.
Step 8	**ipv4 map-resolver** **Example:** `Router(config-router-lisp)# ipv4 map-resolver`	Enables LISP map resolver functionality for EIDs in the IPv4 address family.
Step 9	**ipv4 map-server** **Example:** `Router(config-router-lisp)# ipv4 map-server`	Enables LISP map server functionality for EIDs in the IPv4 address family.
Step 10	**eid-table vrf***vrf-name* **instance-id** *instance-id* **Example:** `Router(config-router-lisp)# eid-table vrf TRANS instance-id 1`	Configures an association between a VRF table and a LISP instance ID, and enters eid-table configuration submode. • In this example, the VRF table TRANS and instance-id 1 are associated together.
Step 11	**database-mapping** *EID-prefix/prefix-length locator* **priority** *priority* **weight** *weight* **Example:** `Router(config-router-lisp-eid-table)# database-mapping 10.1.1.0/24 172.16.1.2 priority 1 weight 100`	Configures an EID-to-RLOC mapping relationship and its associated traffic policy for this LISP site. • In this example, the EID prefix 10.1.1.0/24 within instance-id 1 at this site is associated with the local IPv4 RLOC 172.16.1.2, as well as with the neighbor xTR RLOC 172.6.1.6.
Step 12	Repeat Step 11 until all EID-to-RLOC mappings within this eid-table vrf and instance ID for the LISP site are configured. **Example:** `Router(config-router-lisp-eid-table)# database-mapping 10.1.1.0/24 172.16.1.6 priority 1 weight 100`	Configures an EID-to-RLOC mapping relationship and its associated traffic policy for this LISP site.
Step 13	**ipv4 etr map-server** *map-server-address* **key** *key-type authentication-key* **Example:** `Router(config-router-lisp-eid-table)# ipv4 etr map-server 172.16.1.2 key 0 TRANS-key`	Configures a locator address for the LISP map server and an authentication key for which this router, acting as an IPv4 LISP ETR, will use to register with the LISP mapping system. • In this example, the map server and authentication-key are specified here, within the eid-table subcommand mode, so that the authentication key is associated only with this instance ID, within this VPN.

	Command or Action	Purpose
		Note The map server must be configured with EID prefixes and instance-ids matching the one(s) configured on this ETR, as well as an identical authentication key. **Note** The locator address of the map server may be an IPv4 or IPv6 address. In this example, because each xTR has only IPv4 RLOC connectivity, the map server is reachable using its IPv4 locator addresses. (See the *LISP Command Reference Guide* for more details.)
Step 14	Repeat Step 13 to configure another locator address for the same LISP map server **Example:** `Router(config-router-lisp-eid-table)# ipv4 etr map-server 172.16.1.6 key 0 TRANS-key`	Configures a locator address for the LISP map server and an authentication key for which this router, acting as an IPv4 LISP ETR, will use to register with the LISP mapping system. • In this example, a redundant map server is configured. (Because the MS is co-located with the xTRs in this case, this command indicates that this xTR is pointing to itself for registration (and its neighbor xTR/MS/MR at the same site).
Step 15	**exit** **Example:** `Router(config-router-lisp-eid-table)# exit`	Exits eid-table configuration submode and returns to LISP configuration mode.
Step 16	**ipv4 itr map-resolver** *map-resolver-address* **Example:** `Router(config-router-lisp)# ipv4 itr map-resolver 172.16.1.2`	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. • In this example, the map resolver is specified within router lisp configuration mode and inherited into all eid-table instances since nothing is related to any single instance ID. In addition, redundant map resolvers are configured. (Because the MR is co-located with the xTRs in this case, this command indicates that this xTR is pointing to itself for mapping resolution (and its neighbor xTR/MS/MR at the same site). • The locator address of the map resolver may be an IPv4 or IPv6 address. In this example, because each xTR has only IPv4 RLOC connectivity, the map resolver is reachable using its IPv4 locator address. (See the *LISP Command Reference Guide* for more details.) **Note** Up to two map resolvers may be configured if multiple map resolvers are available. (See the *LISP Command Reference Guide* for more details.)
Step 17	Repeat Step 16 to configure another locator address for the LISP map resolver **Example:** `Router(config-router-lisp)# ipv4 itr map-resolver 172.16.1.6`	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. • In this example, a redundant map resolver is configured. (Because the MR is co-located with the xTRs in this case, this command indicates that this xTR is pointing to itself for mapping resolution (and its neighbor xTR/MS/MR at the same site).

	Command or Action	Purpose
		• The locator address of the map resolver may be an IPv4 or IPv6 address. In this example, because each xTR has only IPv4 RLOC connectivity, the map resolver is reachable using its IPv4 locator address. (See the *LISP Command Reference Guide* for more details.) **Note** Up to two map resolvers may be configured if multiple map resolvers are available. (See the *LISP Command Reference Guide* for more details.)
Step 18	**ipv4 itr** **Example:** `Router(config-router-lisp)# ipv4 itr`	Enables LISP ITR functionality for the IPv4 address family.
Step 19	**ipv4 etr** **Example:** `Router(config-router-lisp)# ipv4 etr`	Enables LISP ETR functionality for the IPv4 address family.
Step 20	**exit** **Example:** `Router(config-router-lisp)# exit`	Exits LISP configuration mode and returns to global configuration mode.

Step 21	**ip route** *ipv4-prefix next-hop*
	Example:
	`Router(config)# ip route 0.0.0.0 0.0.0.0 172.16.1.1`
		Configures a default route to the upstream next hop for all IPv4 destinations.
		• All IPv4 EID-sourced packets destined to both LISP and non-LISP sites are forwarded in one of two ways:
		  • LISP-encapsulated to a LISP site when traffic is LISP-to-LISP
		  • natively forwarded when traffic is LISP-to-non-LISP
		• Packets are deemed to be a candidate for LISP encapsulation when they are sourced from a LISP EID and the destination matches one of the following entries:
		  • a current map-cache entry
		  • a default route with a legitimate next-hop
		  • no route at all
		In this configuration example, because the xTR has IPv4 RLOC connectivity, a default route to the upstream SP is used for all IPv4 packets to support LISP processing.

	Command or Action	Purpose
Step 22	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

Example:

The examples below show the complete configuration for the HQ-RTR-1 and HQ-RTR-2 (xTR/MS/MR located at the HQ Site), and Site2-xTR LISP devices illustrated in the figure above and in this task. Note that both HQ-RTR-1 and HQ-RTR-2 are provided in order to illustrate the proper method for configuring a LISP multihomed site.

Example configuration for HQ-RTR-1 with an xTR, a map server and a map resolver:

```
hostname HQ-RTR-1
!
vrf definition TRANS
 address-family ipv4
 exit
!
vrf definition SOC
address-family ipv4
 exit
!
vrf definition FIN
address-family ipv4
 exit
!
interface Loopback0
 description Management Loopback (in default space)
 ip address 172.31.1.11 255.255.255.255
!
interface GigabitEthernet0/0/0
 description WAN Link to IPv4 Core
 ip address 172.16.1.2 255.255.255.252
 negotiation auto
!
interface GigabitEthernet0/0/1
 vrf forwarding TRANS
 ip address 10.1.1.1 255.255.255.0
 negotiation auto
!
interface GigabitEthernet0/0/2
 vrf forwarding SOC
 ip address 10.2.1.1 255.255.255.0
 negotiation auto
!
interface GigabitEthernet0/0/3
 vrf forwarding FIN
 ip address 10.3.1.1 255.255.255.0
 negotiation auto
!
router lisp
 eid-table default instance-id 0
  database-mapping 172.31.1.11/32 172.16.1.2 priority 1 weight 50
  database-mapping 172.31.1.11/32 172.16.1.6 priority 1 weight 50
  ipv4 etr map-server 172.16.1.2 key DEFAULT-key
  ipv4 etr map-server 172.16.1.6 key DEFAULT-key
  exit
```

```
 !
 eid-table vrf TRANS instance-id 1
  database-mapping 10.1.1.0/24 172.16.1.2 priority 1 weight 50
  database-mapping 10.1.1.0/24 172.16.1.6 priority 1 weight 50
  ipv4 etr map-server 172.16.1.2 key TRANS-key
  ipv4 etr map-server 172.16.1.6 key TRANS-key
  exit
 !
 eid-table vrf SOC instance-id 2
  database-mapping 10.2.1.0/24 172.16.1.2 priority 1 weight 50
  database-mapping 10.2.1.0/24 172.16.1.6 priority 1 weight 50
  ipv4 etr map-server 172.16.1.2 key SOC-key
  ipv4 etr map-server 172.16.1.6 key SOC-key
  exit
 !
 eid-table vrf FIN instance-id 3
  database-mapping 10.3.1.0/24 172.16.1.2 priority 1 weight 50
  database-mapping 10.3.1.0/24 172.16.1.6 priority 1 weight 50
  ipv4 etr map-server 172.16.1.2 key FIN-key
  ipv4 etr map-server 172.16.1.6 key FIN-key
  exit
 !
 site DEFAULT
  authentication-key DEFAULT-key
  eid-prefix 172.31.1.0/24 accept-more-specifics
  exit
 !
 site TRANS
  authentication-key TRANS-key
  eid-prefix instance-id 1 10.1.0.0/16 accept-more-specifics
  exit
 !
 site SOC
  authentication-key SOC-key
  eid-prefix instance-id 2 10.2.0.0/16 accept-more-specifics
  exit
 !
 site FIN
  authentication-key FIN-key
  eid-prefix instance-id 3 10.3.0.0/16 accept-more-specifics
  exit
 !
 ipv4 map-server
 ipv4 map-resolver
 ipv4 itr map-resolver 172.16.1.2
 ipv4 itr map-resolver 172.16.1.6
 ipv4 itr
 ipv4 etr
 exit
!
ip route 0.0.0.0 0.0.0.0 172.16.1.1
```

Example configuration for HQ-RTR-2 with an xTR, a map server and a map resolver:

```
hostname HQ-RTR-2
!
vrf definition TRANS
address-family ipv4
 exit
!
vrf definition SOC
address-family ipv4
 exit
!
vrf definition FIN
address-family ipv4
 exit
!
interface Loopback0
 description Management Loopback (in default space)
 ip address 172.31.1.12 255.255.255.255
!
interface GigabitEthernet0/0/0
```

```
 description WAN Link to IPv4 Core
 ip address 172.16.1.6 255.255.255.252
 negotiation auto
!
interface GigabitEthernet0/0/1
 vrf forwarding TRANS
 ip address 10.1.1.2 255.255.255.0
 negotiation auto
!
interface GigabitEthernet0/0/2
 vrf forwarding SOC
 ip address 10.2.1.2 255.255.255.0
 negotiation auto
!
interface GigabitEthernet0/0/3
 vrf forwarding FIN
 ip address 10.3.1.2 255.255.255.0
 negotiation auto
!
router lisp
 eid-table default instance-id 0
  database-mapping 172.31.1.12/32 172.16.1.2 priority 1 weight 50
  database-mapping 172.31.1.12/32 172.16.1.6 priority 1 weight 50
  ipv4 etr map-server 172.16.1.2 key DEFAULT-key
  ipv4 etr map-server 172.16.1.6 key DEFAULT-key
  exit
 !
 eid-table vrf TRANS instance-id 1
  database-mapping 10.1.1.0/24 172.16.1.2 priority 1 weight 50
  database-mapping 10.1.1.0/24 172.16.1.6 priority 1 weight 50
  ipv4 etr map-server 172.16.1.2 key TRANS-key
  ipv4 etr map-server 172.16.1.6 key TRANS-key
  exit
 !
 eid-table vrf SOC instance-id 2
  database-mapping 10.2.1.0/24 172.16.1.2 priority 1 weight 50
  database-mapping 10.2.1.0/24 172.16.1.6 priority 1 weight 50
  ipv4 etr map-server 172.16.1.2 key SOC-key
  ipv4 etr map-server 172.16.1.6 key SOC-key
  exit
 !
 eid-table vrf FIN instance-id 3
  database-mapping 10.3.1.0/24 172.16.1.2 priority 1 weight 50
  database-mapping 10.3.1.0/24 172.16.1.6 priority 1 weight 50
  ipv4 etr map-server 172.16.1.2 key FIN-key
  ipv4 etr map-server 172.16.1.6 key FIN-key
  exit
 !
 site DEFAULT
  authentication-key DEFAULT-key
  eid-prefix 172.31.1.0/24 accept-more-specifics
  exit
 !
 site TRANS
  authentication-key TRANS-key
  eid-prefix instance-id 1 10.1.0.0/16 accept-more-specifics
  exit
 !
 site SOC
  authentication-key SOC-key
  eid-prefix instance-id 2 10.2.0.0/16 accept-more-specifics
  exit
 !
 site FIN
  authentication-key FIN-key
  eid-prefix instance-id 3 10.3.0.0/16 accept-more-specifics
  exit
 !
 ipv4 map-server
 ipv4 map-resolver
 ipv4 itr map-resolver 172.16.1.2
 ipv4 itr map-resolver 172.16.1.6
 ipv4 itr
```

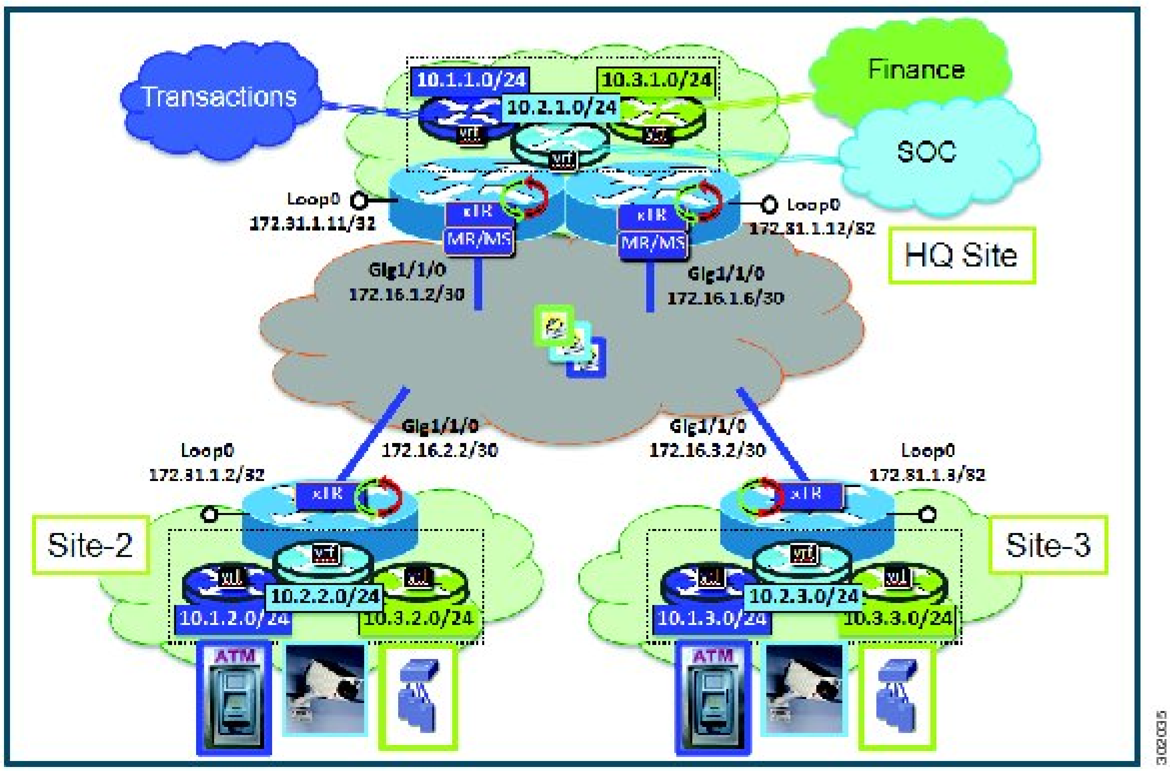
```
 ipv4 etr
 exit
!
ip route 0.0.0.0 0.0.0.0 172.16.1.5
```

Configure a Remote Site for Large-Scale LISP Shared Model Virtualization

Perform this task to enable and configure LISP ITR/ETR (xTR) functionality at a remote site to implement LISP shared model virtualization as part of a large-scale, multiple-site LISP topology.

The configuration demonstrated in this task is part of a more complex, larger scale LISP virtualization solution. In this task, the configuration applies to one of the remote sites shown in the figure below. In this task, the remote site routers only act as xTRs, and use the MS/MRs at the HQ site for LISP control plane support.

Figure 29: Large Scale LISP Site with Virtualized IPv4 EIDs and a Shared IPv4 Core



The components illustrated in the topology shown in the figure above are described below:

- **LISP remote sites:**
  - Each remote site CPE router functions as a LISP ITR and ETR (xTR).
  - Each LISP xTRs has the same three VRFs as the HQ Site: TRANS, SOC, and FIN. Each VRF contains only IPv4 EID-prefixes.
  - Each remote site LISP xTR has a single RLOC connection to a shared IPv4 core network.

Before You Begin

The configuration below assumes that the referenced VRFs were created using the **vrf definition** command and that the Configure a Large-Scale LISP Shared Model Virtualization task has been performed at one or more central (headquarters) sites.

SUMMARY STEPS

1. **configure terminal**
2. **router lisp**
3. **eid-table vrf***vrf-name* **instance-id** *instance-id*
4. **database-mapping** *EID-prefix/prefix-length locator* **priority** *priority* **weight** *weight*
5. **ipv4 etr map-server** *map-server-address* **key** *key-type authentication-key*
6. Repeat Step 13 to configure another locator address for the same LISP map server
7. **exit**
8. **ipv4 itr map-resolver** *map-resolver-address*
9. Repeat Step 16 to configure another locator address for the LISP map resolver
10. **ipv4 itr**
11. **ipv4 etr**
12. **exit**
13. **ip route** *ipv4-prefix next-hop*
14. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 2	**router lisp** **Example:** `Router(config)# router lisp`	Enters LISP configuration mode (software only).
Step 3	**eid-table vrf***vrf-name* **instance-id** *instance-id* **Example:** `Router(config-router-lisp)# eid-table vrf TRANS instance-id 1`	Configures an association between a VRF table and a LISP instance ID, and enters eid-table configuration submode. • In this example, the VRF table TRANS and instance-id 1 are associated together.
Step 4	**database-mapping** *EID-prefix/prefix-length locator* **priority** *priority* **weight** *weight*	Configures an EID-to-RLOC mapping relationship and its associated traffic policy for this LISP site.

	Command or Action	Purpose
	Example: `Router(config-router-lisp-eid-table)# database-mapping 10.1.2.0/24 172.16.2.2 priority 1 weight 100`	• In this example, the EID prefix 10.1.2.0/24 within instance-id 1 at this site is associated with the local IPv4 RLOC 172.16.2.2. **Note** Repeat this step until all EID-to-RLOC mappings within this eid-table vrf and instance ID for the LISP site are configured.
Step 5	**ipv4 etr map-server** *map-server-address* **key** *key-type authentication-key* **Example:** `Router(config-router-lisp-eid-table)# ipv4 etr map-server 172.16.1.2 key 0 TRANS-key`	Configures a locator address for the LISP map server and an authentication key for which this router, acting as an IPv4 LISP ETR, will use to register with the LISP mapping system. • In this example, the map server and authentication-key are specified here, within the eid-table subcommand mode, so that the authentication key is associated only with this instance ID, within this VPN. **Note** The map server must be configured with EID prefixes and instance-ids matching the one(s) configured on this ETR, as well as an identical authentication key. **Note** The locator address of the map server may be an IPv4 or IPv6 address. In this example, because each xTR has only IPv4 RLOC connectivity, the map server is reachable using its IPv4 locator addresses. (See the *LISP Command Reference Guide* for more details.)
Step 6	Repeat Step 13 to configure another locator address for the same LISP map server **Example:** `Router(config-router-lisp-eid-table)# ipv4 etr map-server 172.16.1.6 key 0 TRANS-key`	Configures a locator address for the LISP map server and an authentication key for which this router, acting as an IPv4 LISP ETR, will use to register with the LISP mapping system. • In this example, a redundant map server is configured. (Because the MS is co-located with the xTRs in this case, this command indicates that this xTR is pointing to itself for registration (and its neighbor xTR/MS/MR at the same site).
Step 7	**exit** **Example:** `Router(config-router-lisp-eid-table)# exit`	Exits eid-table configuration submode and returns to LISP configuration mode.
Step 8	**ipv4 itr map-resolver** *map-resolver-address* **Example:** `Router(config-router-lisp)# ipv4 itr map-resolver 172.16.1.2`	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. • In this example, the map resolver is specified within router lisp configuration mode and inherited into all eid-table instances since nothing is related to any single instance ID. In addition, redundant map resolvers are configured. (Because the MR is co-located with the xTRs in this case, this command indicates that this xTR is pointing to itself for mapping resolution (and its neighbor xTR/MS/MR at the same site). • The locator address of the map resolver may be an IPv4 or IPv6 address. In this example, because each xTR has only IPv4 RLOC connectivity,

	Command or Action	Purpose
		the map resolver is reachable using its IPv4 locator address. (See the *LISP Command Reference Guide* for more details.) **Note** Up to two map resolvers may be configured if multiple map resolvers are available. (See the *LISP Command Reference Guide* for more details.)
Step 9	Repeat Step 16 to configure another locator address for the LISP map resolver **Example:** `Router(config-router-lisp)# ipv4 itr map-resolver 172.16.1.6`	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. • In this example, a redundant map resolver is configured. (Because the MR is co-located with the xTRs in this case, this command indicates that this xTR is pointing to itself for mapping resolution (and its neighbor xTR/MS/MR at the same site). • The locator address of the map resolver may be an IPv4 or IPv6 address. In this example, because each xTR has only IPv4 RLOC connectivity, the map resolver is reachable using its IPv4 locator address. (See the *LISP Command Reference Guide* for more details.) **Note** Up to two map resolvers may be configured if multiple map resolvers are available. (See the *LISP Command Reference Guide* for more details.)
Step 10	**ipv4 itr** **Example:** `Router(config-router-lisp)# ipv4 itr`	Enables LISP ITR functionality for the IPv4 address family.
Step 11	**ipv4 etr** **Example:** `Router(config-router-lisp)# ipv4 etr`	Enables LISP ETR functionality for the IPv4 address family.
Step 12	**exit** **Example:** `Router(config-router-lisp)# exit`	Exits LISP configuration mode and returns to global configuration mode.

Step 13	**ip route** *ipv4-prefix next-hop*
	
	Example:
	
	`Router(config)# ip route 0.0.0.0 0.0.0.0 172.16.2.1`
		Configures a default route to the upstream next hop for all IPv4 destinations.
		
		• All IPv4 EID-sourced packets destined to both LISP and non-LISP sites are forwarded in one of two ways:
		
		  • LISP-encapsulated to a LISP site when traffic is LISP-to-LISP
		
		  • natively forwarded when traffic is LISP-to-non-LISP

	Command or Action	Purpose
		• Packets are deemed to be a candidate for LISP encapsulation when they are sourced from a LISP EID and the destination matches one of the following entries: • a current map-cache entry • a default route with a legitimate next-hop • no route at all In this configuration example, because the xTR has IPv4 RLOC connectivity, a default route to the upstream SP is used for all IPv4 packets to support LISP processing.
Step 14	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

Example:

The example below show the complete configuration for the remote site device illustrated in the figure above and in this task. Note that only one remote site configuration is shown here.

Example configuration for Site 2 with an xTR, and using the map server and a map resolver from the HQ site:

```
hostname Site2-xTR
!
vrf definition TRANS
address-family ipv4
 exit
!
vrf definition SOC
address-family ipv4
 exit
!
vrf definition FIN
address-family ipv4
 exit
!
interface Loopback0
 description Management Loopback (in default space)
 ip address 172.31.1.2 255.255.255.255
!
interface GigabitEthernet0/0/0
 description WAN Link to IPv4 Core
 ip address 172.16.2.2 255.255.255.252
 negotiation auto
!
interface GigabitEthernet0/0/1
 vrf forwarding TRANS
 ip address 10.1.2.1 255.255.255.0
 negotiation auto
!
interface GigabitEthernet0/0/2
 vrf forwarding SOC
```

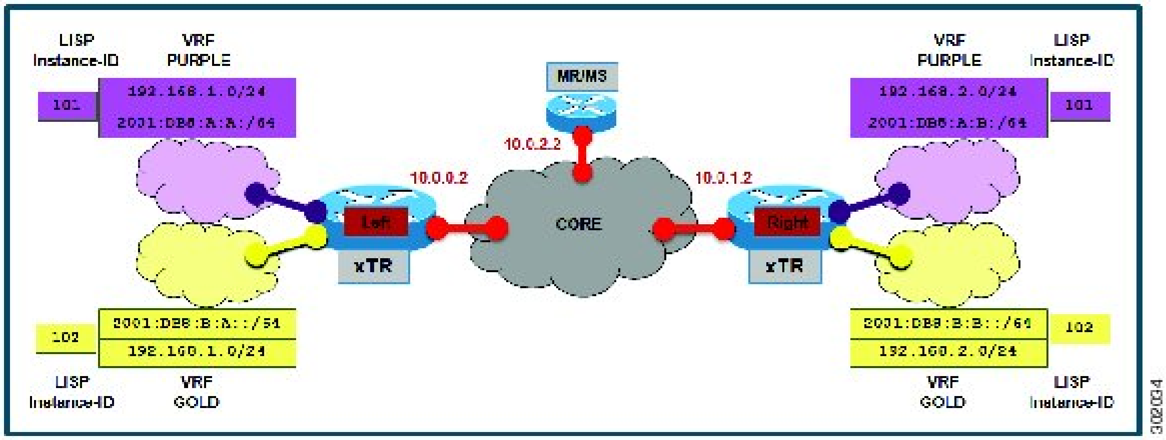
```
 ip address 10.2.2.1 255.255.255.0
 negotiation auto
!
interface GigabitEthernet0/0/3
 vrf forwarding FIN
 ip address 10.3.2.1 255.255.255.0
 negotiation auto
!
router lisp
 eid-table default instance-id 0
  database-mapping 172.31.1.2/32 172.16.2.2 priority 1 weight 100
  ipv4 etr map-server 172.16.1.2 key DEFAULT-key
  ipv4 etr map-server 172.16.1.6 key DEFAULT-key
  exit
 !
 eid-table vrf TRANS instance-id 1
  database-mapping 10.1.2.0/24 172.16.2.2 priority 1 weight 100
  ipv4 etr map-server 172.16.1.2 key TRANS-key
  ipv4 etr map-server 172.16.1.6 key TRANS-key
  exit
 !
 eid-table vrf SOC instance-id 2
  database-mapping 10.2.2.0/24 172.16.2.2 priority 1 weight 100
  ipv4 etr map-server 172.16.1.2 key SOC-key
  ipv4 etr map-server 172.16.1.6 key SOC-key
  exit
 !
 eid-table vrf FIN instance-id 3
  database-mapping 10.3.2.0/24 172.16.2.2 priority 1 weight 100
  ipv4 etr map-server 172.16.1.2 key FIN-key
  ipv4 etr map-server 172.16.1.6 key FIN-key
  exit
 !
 ipv4 itr map-resolver 172.16.1.2
 ipv4 itr map-resolver 172.16.1.6
 ipv4 itr
 ipv4 etr
 exit
!
ip route 0.0.0.0 0.0.0.0 172.16.2.1
```

Verifying and Troubleshooting LISP Virtualization

After configuring LISP, verifying and troubleshooting LISP configuration and operations may be performed by following the optional steps described below. Note that certain verification and troubleshooting steps may only apply to certain types of LISP devices.

In this task, the topology is shown in the figure below and the configuration is from the "Configure Simple LISP Shared Model Virtualization" task, but the commands are applicable to both LISP shared and parallel model virtualization.

Figure 30: Simple LISP Site with Virtualized IPv4 and IPv6 EIDs and a Shared IPv4 Core



> **Note**
> The following examples do not show every available command and every available output display. Refer to the *Cisco IOS LISP Command Reference* for detailed explanations of each command.

SUMMARY STEPS

1. **enable**
2. **show running-config | section router lisp**
3. **show** [**ip** | **ipv6**] **lisp**
4. **show** [**ip** | **ipv6**] **lisp map-cache**
5. **show** [**ip** | **ipv6**] **lisp database** [**eid-table vrf** *vrf-name*]
6. **show lisp site** [**name** *site-name*]
7. **lig** {[**self** {**ipv4** | **ipv6**}] | {*hostname* | *destination-EID*}
8. **ping** {*hostname* | *destination-EID*}
9. **clear** [**ip** | **ipv6**] **lisp map-cache**

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode. Enter your password if prompted.

Example:

```
Router> enable
```

Step 2 **show running-config | section router lisp**

The **show running-config | section router lisp** command is useful for quickly verifying the LISP configuration on the device. This command applies to any LISP device. The following is sample output from the **show running-config | section router lisp** command when a simple LISP site is configured with virtualized IPv4 and IPv6 EID prefixes and a shared IPv4 core:

Example:

```
Router# show running-config | section router lisp

router lisp
 eid-table vrf PURPLE instance-id 101
  database-mapping 192.168.1.0/24 10.0.0.2 priority 1 weight 1
  database-mapping 2001:DB8:A:A::/64 10.0.0.2 priority 1 weight 1
 eid-table vrf GOLD instance-id 102
  database-mapping 192.168.1.0/24 10.0.0.2 priority 1 weight 1
  database-mapping 2001:DB8:B:A::/64 10.0.0.2 priority 1 weight 1
 exit
 !
 ipv4 itr map-resolver 10.0.2.2
 ipv4 itr
 ipv4 etr map-server 10.0.2.2 key Left-key
 ipv4 etr
 ipv6 itr map-resolver 10.0.2.2
 ipv6 itr
 ipv6 etr map-server 10.0.2.2 key Left-key
 ipv6 etr
 exit
```

Step 3 **show** [**ip** | **ipv6**] **lisp**

The **show ip lisp** and **show ipv6 lisp** commands are useful for quickly verifying the operational status of LISP as configured on the device, as applicable to the IPv4 and IPv6 address families respectively. This command applies to any LISP device.

Example:

The first example shows a summary of LISP operational status and IPv6 address family information by EID table:

```
Router# show ipv6 lisp eid-table summary

Instance count: 2
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
PURPLE                LISP0.101     1      0      1  0.0%  0.0% ITR-ETR
GOLD                  LISP0.102     1      0      1  0.0%  0.0% ITR-ETR
```

Example:

The second example shows LISP operational status and IPv6 address family information for the VRF named PURPLE:

```
Router# show ipv6 lisp eid-table vrf PURPLE
```

```
  Instance ID:                       101
  Router-lisp ID:                    0
  Locator table:                     default
  EID table:                         PURPLE
  Ingress Tunnel Router (ITR):       enabled
  Egress Tunnel Router (ETR):        enabled
  Proxy-ITR Router (PITR):           disabled
  Proxy-ETR Router (PETR):           disabled
  Map Server (MS):                   disabled
  Map Resolver (MR):                 disabled
  Map-Request source:                2001:DB8:A:A::1
  ITR Map-Resolver(s):               10.0.2.2
  ETR Map-Server(s):                 10.0.2.2 (00:00:24)
  ITR use proxy ETR RLOC(s):         none
```

Example:

The third example shows LISP operational status and IPv6 address family information for the instance ID of 101:

```
Router# show ipv6 lisp instance-id 101

  Instance ID:                       101
  Ingress Tunnel Router (ITR):       enabled
  Egress Tunnel Router (ETR):        enabled
  Proxy-ITR Router (PITR):           disabled
  Proxy-ETR Router (PETR):           disabled
  Map Server (MS):                   disabled
  Map Resolver (MR):                 disabled
  Map-Request source:                2001:DB8:A:A::1
  ITR Map-Resolver(s):               10.0.2.2
  ETR Map-Server(s):                 10.0.2.2 (00:00:11)
  ITR Solicit Map Request (SMR):     accept and process
    Max SMRs per map-cache entry:    8 more specifics
    Multiple SMR suppression time:   60 secs
  ETR accept mapping data:           disabled, verify disabled
  ETR map-cache TTL:                 1d00h
```

Step 4 **show** [**ip** | **ipv6**] **lisp map-cache**

The **show ip lisp map-cache** and **show ipv6 lisp map-cache** commands are useful for quickly verifying the operational status of the map cache on a device configured as an ITR or PITR, as applicable to the IPv4 and IPv6 address families respectively.

Example:

The following example shows IPv6 mapping cache information based on a configuration when a simple LISP site is configured with virtualized IPv4 and IPv6 EID prefixes and a shared IPv4 core. This example output assumes that a map-cache entry has been received for another site with the IPv6 EID prefix 2001:db8:b:b::/64.

```
Router# show ip lisp map-cache eid-table vrf GOLD

LISP IPv6 Mapping Cache for EID-table vrf GOLD (IID 102), 2 entries

::/0, uptime: 01:09:52, expires: never, via static send map-request
  Negative cache entry, action: send-map-request
2001:DB8:B:B::/64, uptime: 00:00:10, expires: 23:59:42, via map-reply, complete
  Locator   Uptime    State      Pri/Wgt
  10.0.1.2  00:00:10  up           1/1
```

Step 5 **show** [**ip** | **ipv6**] **lisp database** [**eid-table vrf** *vrf-name*]

The **show ip lisp database** and **show ipv6 lisp database** commands are useful for quickly verifying the operational status of the database mapping on a device configured as an ETR, as applicable to the IPv4 and IPv6 address families respectively.

Example:

The following example shows IPv6 mapping database information for the VRF named GOLD.

```
Router# show ipv6 lisp database eid-table vrf GOLD

LISP ETR IPv6 Mapping Database for EID-table vrf GOLD (IID 102), LSBs: 0x1, 1 entries

EID-prefix: 2001:DB8:B:A::/64
  10.0.0.2, priority: 1, weight: 1, state: site-self, reachable
```

Step 6 **show lisp site** [**name** *site-name*]

The **show lisp site** command is useful for quickly verifying the operational status of LISP sites, as configured on a map server. This command only applies to a device configured as a map server. The following example output is based on a configuration when a simple LISP site is configured with virtualized IPv4 and IPv6 EID prefixes and shows the information for the instance ID of 101.

Example:

```
Router# show lisp site instance-id 101

LISP Site Registration Information

Site Name      Last      Up   Who Last            Inst     EID Prefix
               Register       Registered          ID
Left           00:00:36  yes  10.0.0.2            101      192.168.1.0/24
               00:00:43  yes  10.0.0.2            101      2001:DB8:A:A::/64
Right          00:00:31  yes  10.0.1.2            101      192.168.2.0/24
               00:00:02  yes  10.0.1.2            101      2001:DB8:A:B::/64
```

Example:

This second example shows LISP site information for the IPv6 EID prefix of 2001:db8:a:a:/64 and instance ID of 101.

```
Router# show lisp site 2001:db8:a:a:/64 instance-id 101

LISP Site Registration Information

Site name: Left
Allowed configured locators: any
Requested EID-prefix:
  EID-prefix: 2001:DB8:A:A::/64 instance-id 101
    First registered:     02:41:55
    Routing table tag:    0
    Origin:               Configuration
    Registration errors:
      Authentication failures:   4
      Allowed locators mismatch: 0
    ETR 10.0.0.2, last registered 00:00:22, no proxy-reply, no map-notify
                TTL 1d00h
      Locator   Local  State      Pri/Wgt
      10.0.0.2  yes    up           1/1
```

Step 7 **lig** {[**self** {**ipv4** | **ipv6**}] | {*hostname* | *destination-EID*}

The LISP Internet Groper (lig) command is useful for testing the LISP control plane. The **lig** command can be used to query for the indicated destination hostname or EID, or the routers local EID-prefix. This command provides a simple means of testing whether a destination EID exists in the LISP mapping database system, or your site is registered with the mapping database system. This command is applicable for both the IPv4 and IPv6 address families and applies to any LISP device that maintains a map cache (for example, if configured as an ITR or PITR). The following example output is based on a configuration when a simple LISP site is configured with virtualized IPv4 and IPv6 EID prefixes and shows the information for the instance ID of 101 and the IPv4 EID prefix of 192.168.2.1.

Example:

```
Router# lig instance-id 101 192.168.2.1

Mapping information for EID 192.168.2.1 from 10.0.1.2 with RTT 12 msecs
192.168.2.0/24, uptime: 00:00:00, expires: 23:59:52, via map-reply, complete
  Locator   Uptime    State      Pri/Wgt
  10.0.1.2  00:00:00  up           1/1
```

Example:

This second example output shows information about the VRF named PURPLE:

```
Router# lig eid-table vrf PURPLE self

Mapping information for EID 192.168.1.0 from 10.0.0.1 with RTT 20 msecs
192.168.1.0/24, uptime: 00:00:00, expires: 23:59:52, via map-reply, self
  Locator   Uptime    State      Pri/Wgt
  10.0.0.1  00:00:00  up, self     1/1
```

Step 8 **ping** {*hostname* | *destination-EID*}

The **ping** command is useful for testing basic network connectivity and reachability and/or liveness of a destination EID or RLOC address. When using **ping** it is important to be aware that because LISP uses an encapsulation, you should always specify a source address; never allow the **ping** application to assign its own default source address. This is because there are four possible ways to use **ping**, and without explicitly indicating the source address, the wrong one may be used by the application leading to erroneous results that complicate operational verification or troubleshooting. The four possible uses of **ping** include:

- RLOC-to-RLOC—Sends "echo□? packets out natively (no LISP encap) and receive the "echo-reply□? back natively. This can be used to test the underlying network connectivity between locators of various devices, such as xTR to Map-Server or Map-Resolver.
- EID-to-EID—Sends "echo□? packets out LISP-encaped and receive the "echo-reply□? back LISP-encaped. This can be used to test the LISP data plane (encapsulation) between LISP sites.
- EID-to-RLOC—Sends "echo□? packets out natively (no LISP encap) and receive the "echo-reply" back LISP-encaped through a PITR mechanism. This can be used to test the PITR infrastructure.
- RLOC-to-EID - Sends "echo□? packets out LISP-encaped and receive the "echo-reply□? back natively. This can be used to test PETR capabilities.

The **ping** command is applicable to the IPv4 and IPv6 address families respectively, and can be used on any LISP device in some manner. (The ability to do LISP encapsulation, for example, requires the device to be configured as an ITR or PITR.)

The following example output from the **ping** command is based on a configuration when a simple LISP site is configured with virtualized IPv4 and IPv6 EID prefixes. (Note that ping is not a LISP command and does not know about an EID table or an instance ID. When virtualization is included, output limiters can only be specified by VRF.)

Example:

```
Router# ping vrf PURPLE 2001:DB8:a:b::1 source 2001:DB8:a:a::1 rep 100

Type escape sequence to abort.
Sending 100, 100-byte ICMP Echos to 2001:DB8:A:B::1, timeout is 2 seconds:
Packet sent with a source address of 2001:DB8:A:A::1%PURPLE
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 100 percent (100/100), round-trip min/avg/max = 0/0/1 ms
```

Example:

```
Router# ping vrf GOLD

Protocol [ip]: ipv6
Target IPv6 address: 2001:db8:b:b::1
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands? [no]: y
Source address or interface: 2001:db8:b:a::1
.
.
.
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:B:B::1, timeout is 2 seconds:
Packet sent with a source address of 2001:DB8:B:A::1%GOLD
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/0/0 ms
```

Step 9 **clear** [**ip** | **ipv6**] **lisp map-cache**

The **clear ip lisp map-cache** and **clear ipv6 lisp map-cache** commands remove all IPv4 or IPv6 dynamic LISP map-cache entries stored by the router. This can be useful trying to quickly verify the operational status of the LISP control plane. This command applies to a LISP device that maintains a map cache (for example, if configured as an ITR or PITR).

Example:

The following example displays IPv4 mapping cache information for instance ID 101, shows the command used to clear the mapping cache for instance ID 101, and displays the show information after clearing the cache.

```
Router# show ip lisp map-cache instance-id 101

LISP IPv4 Mapping Cache for EID-table vrf PURPLE (IID 101), 2 entries

0.0.0.0/0, uptime: 00:25:17, expires: never, via static send map-request
  Negative cache entry, action: send-map-request
192.168.2.0/24, uptime: 00:20:13, expires: 23:39:39, via map-reply, complete
  Locator   Uptime    State      Pri/Wgt
  10.0.1.2  00:20:13  up           1/1

Router# clear ip lisp map-cache instance-id 101

Router# show ip lisp map-cache instance-id 101

LISP IPv4 Mapping Cache, 1 entries

0.0.0.0/0, uptime: 00:00:02, expires: never, via static send map-request
  Negative cache entry, action: send-map-request
```

Configuration Examples for LISP Shared Model Virtualization

Complete configuration examples are available within each task under the “How to Configure LISP Shared Model Virtualization” section.

Additional References

Related Documents

Document Title	Location
Cisco IOS IP Routing: LISP Command Reference	http://www.cisco.com/en/US/docs/ios-xml/ios/iproute_lisp/command/ip-lisp-cr-book.html
Enterprise IPv6 Transitions Strategy Using the Locator/ID Separation Protocol	Cisco LISP Software Image Download Page
Cisco IOS LISP0 Virtual Interface, Application Note, Version 1.0	Cisco LISP Software Image Download Page
Cross-Platform Release Notes for Cisco IOS Release 15.2M&T	http://www.cisco.com/en/US/docs/ios/15_2m_and_t/release/notes/15_2m_and_t.html

Standards

Standard	Title
IANA Address Family Numbers	http://www.iana.org/assignments/address-family-numbers/address-family-numbers.xml

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
draft-ietf-lisp-22	Locator/ID Separation Protocol (LISP) http://tools.ietf.org/html/draft-ietf-lisp-22
draft-ietf-lisp-ms-16	LISP Map Server http://tools.ietf.org/html/draft-ietf-lisp-ms-16

RFC	Title
draft-ietf-lisp-alt-10	LISP Alternative Topology (LISP+ALT) http://tools.ietf.org/html/draft-ietf-lisp-alt-10
draft-ietf-lisp-LCAF-06	LISP Canonical Address Format (LCAF) http://tools.ietf.org/wg/lisp/
draft-ietf-lisp-interworking-06	Interworking LISP with IPv4 and IPv6 http://tools.ietf.org/html/draft-ietf-lisp-interworking-06
draft-ietf-lisp-lig-06	LISP Internet Groper (LIG) http://tools.ietf.org/html/draft-ietf-lisp-lig-06
draft-ietf-lisp-mib-03	LISP MIB http://tools.ietf.org/wg/lisp/draft-ietf-lisp-mib/

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for LISP Shared Model Virtualization

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

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

Table 3: Feature Information for LISP Shared Model Virtualization

Feature Name	Releases	Feature Information
LISP Shared Model Virtualization	15.2(2)T 15.1(1)SY1	LISP Shared Model Virtualization feature uses Endpoint Identifier (EID) spaces that are created by binding VRFs associated with an EID space to Instance IDs. A common, “shared” locator space is used by all virtualized EIDs.

CHAPTER 4

LISP Host Mobility Across Subnet

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

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

Information About LISP Host Mobility Across Subnet

Overview of LISP Host Mobility Across Subnet

You can use LISP Host Mobility Across Subnet commands to deploy extended subnets and across subnets. A detailed configuration guide and examples are under development and will appear here soon. Meanwhile, please refer to the *LISP Command Reference*.