Inspur

CN12700 Series

INOS LISP Configuration Guide

(Release 8.x)

Inspur-Cisco Networking Technology Co.,Ltd. provides customers with comprehensive technical support and services. For any assistance, please contact our local office or company headquarters.
Website: http://www.inspur.com/
Technical Support Tel: 400-691-1766
Technical Support Email:inspur_network@inspur.com
Technical Document Support Email:inspur_network@inspur.com
Address: 1036 Langchao Road, Lixia District, Jinan City, Shandong Province
Postal code: 250101

Notice

Preface

Objectives

This guide describes main functions of the CN12700 Series. To have a quick grasp of the CN12700 Series, please read this manual carefully.

Versions

The following table lists the product versions related to this document.

Product name	Version
CN12700 Series	

Conventions

Symbol conventions

The symbols that may be found in this document are defined as follows.

Symbol	Description
Warning	Indicates a hazard with a medium or low level of risk which, if not avoided, could result in minor or moderate injury.
Caution	Indicates a potentially hazardous situation that, if not avoided, could cause equipment damage, data loss, and performance degradation, or unexpected results.
Note	Provides additional information to emphasize or supplement important points of the main text.
Tip	Indicates a tip that may help you solve a problem or save time.

General conventions

Convention	Description
Boldface	Names of files, directories, folders, and users are in **boldface**. For example, log in as user **root**.
Italic	Book titles are in *italics*.
`Lucida Console`	Terminal display is in `Lucida Console`.

Command conventions

Convention	Description
Boldface	The keywords of a command line are in **boldface**.
Italic	Command arguments are in *italics*.
[]	Items (keywords or arguments) in square brackets [ ] are optional.
{ x | y | ... }	Alternative items are grouped in braces and separated by vertical bars. One is selected.
[x | y | ...]	Optional alternative items are grouped in square brackets and separated by vertical bars. One or none is selected.
{ x | y | ... } *	Alternative items are grouped in braces and separated by vertical bars. A minimum of one or a maximum of all can be selected.
[x | y | ...] *	The parameter before the & sign can be repeated 1 to n times.

GUI conventions

Convention	Description
Boldface	Buttons, menus, parameters, tabs, windows, and dialog titles are in **boldface**. For example, click **OK**.
>	Multi-level menus are in boldface and separated by the ">" signs. For example, choose **File** > **Create** > **Folder**.

Keyboard operation

Format	Description
Key	Press the key. For example, press **Enter** and press **Tab**.

Format	Description
Key 1+Key 2	Press the keys concurrently. For example, pressing **Ctrl+C** means the two keys should be pressed concurrently.
Key 1, Key 2	Press the keys in turn. For example, pressing **Alt**, **A** means the two keys should be pressed in turn.

Mouse operation

Action	Description
Click	Select and release the primary mouse button without moving the pointer.
Double-click	Press the primary mouse button twice continuously and quickly without moving the pointer.
Drag	Press and hold the primary mouse button and move the pointer to a certain position.

Change history

Updates between document versions are cumulative. Therefore, the latest document version contains all updates made to previous versions.

Issue 01 (2020-02-24)

Initial commercial release

Contents

Contents ... i

Figures ... iii

Tables ... v

CHAPTER 1 Configuring Locator ID Separation Protocol ... 1

1.1 Information About Locator/ID Separation Protocol ... 1

1.2 Information About LISP ... 1

1.3 LISP Devices Overview ... 2

1.4 Licensing Requirements for LISP ... 3

1.5 LISP Guidelines and Limitations ... 3

1.6 Default Settings for LISP ... 3

1.7 Configuring Locator/ID Separation Protocol ... 3

1.8 Additional References ... 12

1.9 Feature History for LISP ... 13

CHAPTER 2 Configuring LISP ESM Multihop Mobility ... 14

2.1 Finding Feature Information ... 14

2.2 Information About LISP ESM Multihop Mobility ... 14

2.3 Licensing Requirements for LISP ... 14

2.4 Guidelines and Limitations for LISP ESM Multihop Mobility ... 14

2.5 Default Settings for LISP ... 15

2.6 Configuring LISP ESM Multihop Mobility ... 15

2.7 Configuration Examples for LISP ESM Multihop Mobility ... 20

2.8 Additional References ... 24

2.9 Feature Information for LISP ESM Multihop Mobility ... 25

CHAPTER 3 LISP Instance-ID Support ... 26

3.1 Information about LISP Instance-ID Support ... 26

3.2 How to Configure LISP Instance-ID Support ... 32

3.3 Configuration Examples for LISP Instance-ID Support ... 63

3.4 Feature History for Configuring LISP Instance ID ... 74

CHAPTER 4 Configuring LISP Delegate Database Tree (DDT) ... 75

4.1 LISP Delegate Database Tree (DDT)....75
4.2 Overview of DDT....75
4.3 Restrictions for LISP Delegate Database Tree (DDT)....75
4.4 Configuring LISP Delegate Database Tree (DDT)....75
4.5 Configuration Examples for LISP Delegate Database Tree (DDT)....77
4.6 Feature History for Delegate Database Tree....77
CHAPTER 5 Configuring LISP Multicast....78
5.1 LISP Multicast....78
5.2 Finding Feature Information....78
5.3 Restrictions for LISP Multicast....78
5.4 Configuration Example for LISP Multicast....81
5.5 Feature History for LISP Multicast....82
CHAPTER 6 LISP Support for Disjointed RLOC Domains....83
6.1 LISP Support for Disjointed RLOC Domains....83
CHAPTER 7 Configuring LISP Extranets....91
7.1 Finding Feature Information....91
7.2 Feature History for LISP Extranets....91
7.3 Information About LISP Extranets....91
7.4 Licensing Requirements for LISP....93
7.5 Guidelines and Limitations for LISP Extranets....93
7.6 Configuring LISP Extranets....94
CHAPTER 8 Redistribution of RIB Routes into LISP....101
8.1 Finding Feature Information....101
8.2 Feature History for Redistribution of RIB Routes into LISP....101
8.3 Information About Redistribution of RIB Routes into LISP....101
8.4 Configuring Database Application for Redistribution of RIB Routes into LISP....102
8.5 Configuring Map-cache Application for Redistribution of RIB Routes into LISP....103
8.6 Example: Redistribution of RIB Routes in LISP....104
CHAPTER 9 Configuration Limits for LISP....110
9.1 Configuration Limits for LISP....110

Figures

Figure 1 : Inspur INOS LISP Deployment Environment.......... 1
Figure 2 : LISP ESM Multihop Topology.......... 20
Figure 3 : Sample Topology.......... 20
Figure 4 : LISP Deployment Environment.......... 26
Figure 5 : Device Level Virtualization.......... 27
Figure 6 : Path Level Virtualization.......... 27
Figure 7 : Default (Nonvirtualized) LISP Model.......... 28
Figure 8 : LISP Shared Model Virtualization resolves EIDs within VRFs tied to Instance IDs. The default (global) routing table is the shared space.......... 28
Figure 9 : LISP shared model single tenancy use case. A customers uses its own xTR and shares a common core network and mapping system.......... 29
Figure 10: LISP shared model multitenancy use case. Customer's use shared xTRs and share a common core network and mapping system.......... 29
Figure 11: LISP parallel model virtualization resolves an EID and associated RLOCs within the same or a different VRF. In this example, both EID and RLOC addresses are resolved in the same VRF, but multiple (parallel) segmentation is configured on the sa.......... 30
Figure 12: LISP shared and parallel models may be combined for maximum flexibility.......... 30
Figure 13: LISP parallel model multitenancy case. Shared xTRs use virtualized core networks and mapping systems. LISP instance IDs segment the LISP data plane and control plane.......... 31
Figure 14: Simple LISP Site with Virtualized IPv4 and IPv6 EIDs and a Shared IPv4 core.......... 32
Figure 15: Large Scale LISP Site with Virtualized IPv4 EIDs and a Shared IPv4 Core.......... 41
Figure 16 : Large Scale LISP Site with Virtualized IPv4 EIDs and a Shared IPv4 Core.......... 49
Figure 17 : Simple LISP Site with One IPv4 RLOC and One IPv4 EID.......... 54
Figure 18 : Simple LISP Site with One IPv4 RLOC and One IPv4 EID.......... 60
Figure 19Figure 19: Sample topology—LISP Support for Disjointed RLOC Domains.......... 84
Figure 20 : Inspur INOS LISP Extranets Topology.......... 92
Figure 21 : Packet Flow in LISP VRF Leaking.......... 92
Figure 22 : LISP Local Extranets.......... 93
Figure 23 : LISP Extranet Topology.......... 94

Figure 24 : LISP Extranets Topology......95
Figure 25 : Local LISP Extranets......98
Figure 26 : Redistribution of RIB Routes in a LISP Topology......104
Figure 27 : Redistribute RIB Routes into Database Topology......105
Figure 28 : Redistribute RIB Routes into Map-cache Topology......107

Tables

Table 1 : LISP Default Settings 3
Table 2 : Feature History for LISP 13
Table 3 : LISP Default Settings 15
Table 4 : Feature History for Configuring LISP Instance ID 74
Table 5 : Feature History for LISP Delegate Database Tree 77
Table 6 : Feature History for LISP Multicast 82
Table 7 : Feature History for LISP Support for Disjointed RLOC Domains 90
Table 8: Feature History for LISP Extranets 91
Table 9 : Feature History for Redistribution of RIB Routes into LISP 101
Table 10 : Supported RIB Sources for Route Import 101

CHAPTER 1 Configuring Locator ID Separation Protocol

This chapter describes how to configure the basic Inspur INOS Locator/ID Separation Protocol (LISP) functionality on all LISP-related devices, including the Ingress Tunnel Router (ITR), Egress Tunnel Router, Proxy ITR (PITR), Proxy ETR (PETR), Map Resolver (MR), Map Server (MS), and LISP-ALT device.

This chapter contains the following sections:

- Information About Locator/ID Separation Protocol.
- Information About LISP.
- LISP Devices Overview.
- Licensing Requirements for LISP.
- LISP Guidelines and Limitations.
- Default Settings for LISP.
- Configuring Locator/ID Separation Protocol.
- Additional References.
- Feature History for LISP.

1.1 Information About Locator/ID Separation Protocol

The Locator/ID Separation Protocol (LISP) network architecture and protocol implements a new semantic for IP addressing by creating two new 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 improves routing system scalability, multihoming efficiency, and ingress traffic engineering. LISP end site support is configured on devices such as Inspur routers.

1.2 Information About LISP

In the current Internet routing and addressing architecture, the IP address is used as a single namespace that simultaneously expresses two functions about a device: its identity and how it is attached to the network. One very visible and detrimental result of this single namespace is demonstrated by the rapid growth of the Internet's default-free zone (DFZ) as a consequence of multi-homing, traffic engineering (TE), nonaggregatable address allocations, and business events such as mergers and acquisitions.

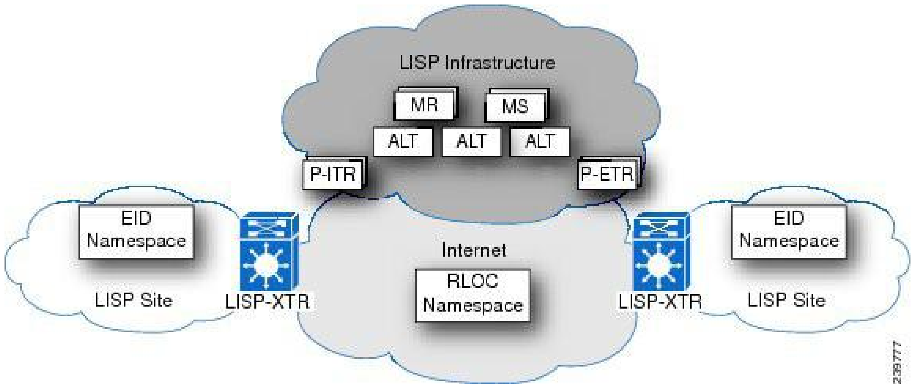
LISP changes current IP address semantics by creating two new namespaces: Endpoint Identifiers (EIDs) that are assigned to end-hosts and Routing Locators (RLOCs) that are assigned to devices (primarily routers) that make up the global routing system. These two namespaces provide the following advantages:

- Improved routing system scalability by using topologically aggregated RLOCs
- Provider independence for devices numbered out of the EID space
- Multihoming of endsites with improved traffic engineering
- IPv6 transition functionality

LISP is deployed primarily in network edge devices. It requires no changes to host stacks, Domain Name Service (DNS), or local network infrastructure, and little to no major changes to existing network infrastructures.

Figure 1 : Inspur INOS LISP Deployment Environment

This figure shows a LISP deployment environment. Three essential environments exist in a LISP environment: LISP sites (EID namespace), non-LISP sites (RLOC namespace), and LISP Mapping Service (infrastructure).



The LISP EID namespace represents customer end sites as they are defined today. The only difference is that the IP addresses used within these LISP sites are not advertised within the non-LISP, Internet (RLOC namespace). End customer LISP functionality is deployed exclusively on CE routers that function within LISP as Ingress Tunnel Router (ITR) and Egress Tunnel Router (ETR) devices.

To fully implement LISP with support for Mapping Services and Internet interworking, you might need to deploy additional LISP infrastructure components such as Map Server (MS), Map Resolver (MR), Proxy Ingress Tunnel Router (PITR), Proxy Egress Tunnel Router (PETR), and Alternative Topology (ALT).

1.3 LISP Devices Overview

The following devices are found in a full LISP deployment:

1.3.1 LISP Site Devices

The LISP site devices are as follows:

Ingress Tunnel Router (ITR)—This device is deployed as a LISP site edge device. It receives packets from site-facing interfaces (internal hosts) and either LISP encapsulates packets to remote LISP sites or the ITR natively forwards packets to non-LISP sites.

Egress Tunnel Router (ETR)—This device is deployed as a LISP site edge device. It receives packets from core-facing interfaces (the Internet) and either decapsulates LISP packets or delivers them to local EIDs at the site.

1.3.2 LISP Infrastructure

The LISP infrastructure devices are as follows:

Map Server (MS)—This device is deployed as a LISP Infrastructure component. It must be configured to permit a LISP site to register to it by specifying for each LISP site the EID prefixes for which registering ETRs are authoritative. An authentication key must match the key that is configured on the ETR. An MS receives Map-Register control packets from ETRs. When the MS is configured with a service interface to the LISP ALT, it injects aggregates for the EID prefixes for registered ETRs into the ALT. The MS also receives Map-Request control packets from the ALT, which it then encapsulates to the registered ETR that is authoritative for the EID prefix being queried.

Map Resolver (MR)—This device is deployed as a LISP Infrastructure device. It receives Map-Requests encapsulated to it from ITRs. When configured with a service interface to the LISP ALT, the MR forwards Map Requests to the ALT. The MR also sends Negative Map-Replies to ITRs in response to queries for non-LISP addresses.

Alternative Topology (ALT)—This is a logical topology and is deployed as part of the LISP Infrastructure to provide scalable EID prefix aggregation. Because the ALT is deployed as a dual-stack (IPv4 and IPv6) Border Gateway Protocol (BGP) over Generic Routing Encapsulation (GRE) tunnels, you can use ALT-only devices with basic router hardware or other off-the-shelf devices that can support BGP and GRE.

1.3.3 LISP Internetworking Devices

The LISP internetworking devices are as follows:

Proxy ITR (PITR)—This device is a LISP infrastructure device that provides connectivity between non-LISP sites and LISP sites. A PITR advertises coarse-aggregate prefixes for the LISP EID namespace into the Internet, which attracts non-LISP traffic destined to LISP sites. The PITR then encapsulates and forwards this traffic

to LISP sites. This process not only facilitates LISP/non-LISP internetworking but also allows LISP sites to see LISP ingress traffic engineering benefits from non-LISP traffic.

Proxy ETR (PETR)—This device is a LISP infrastructure device that allows IPv6 LISP sites without native IPv6 RLOC connectivity to reach LISP sites that only have IPv6 RLOC connectivity. In addition, the PETR can also be used to allow LISP sites with Unicast Reverse Path Forwarding (URPF) restrictions to reach non-LISP sites.

1.4 Licensing Requirements for LISP

The following table shows the LISP licensing requirements:

Product	**License Requirement**
Inspur INOS	This feature requires the LAN_ENTERPRISE_SERVICES_PKG license. For a complete explanation of the Inspur INOS licensing scheme, see the *Inspur INOS Licensing Guide*.

1.5 LISP Guidelines and Limitations

LISP has the following configuration guidelines and limitations:

• Use an Overlay Transport Virtualization (OTV) or another LAN extension mechanism to filter the HSRP hello messages across the data centers to create an active-active HSRP setup and provide egress path optimization for the data center hosts.

• Make sure that the HSRP group and the HSRP Virtual IP address in all data centers in the extended LAN are the same. Keeping the HSRP group number consistent across locations guarantees that the same MAC address is always used for the virtual first-hop gateway.

• LISP VM mobility across subnets requires that the same MAC address is configured across all HSRP groups that allow dynamic EIDs to roam. You must enable the Proxy Address Resolution Protocol (ARP) for the interfaces that have VM mobility enabled across subnets.

1.6 Default Settings for LISP

This table lists the default settings for LISP parameters.

Table 1 : LISP Default Settings

Parameters	**Default**
feature lisp command	Disabled

1.7 Configuring Locator/ID Separation Protocol

1.7.1 Enabling the LISP Feature

You can enable the LISP feature on the Inspur INOS device.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**feature lisp** **Example:** `switch(config)# feature lisp`	Enables the LISP feature set if it is not already configured.

1.7.2 Configuring LISP ITR/ETR (xTR) Functionality

Configuring LISP ITR/ETR (xTR)

You can enable and configure a LISP xTR with a LISP Map-Server and Map-Resolver for mapping services for both IPv4 and IPv6 address families.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	{ **ip** | **ipv6** } **lisp itr**
	Example:
	`switch(config)# ip lisp itr`
	Example:
	`switch(config)# ipv6 lisp itr`
		Enables LISP ITR functionality.

Step 3	{ **ip** | **ipv6** } **lisp etr**
	Example:
	`switch(config)# ip lisp etr`
	Example:
	`switch(config)# ipv6 lisp etr`
		Enables LISP ETR functionality.

Step 4	(Optional) { **ip** | **ipv6** } **lisp itr-etr**
	Example:
	`switch(config)# ip lisp itr-etr`
	Example:
	`switch(config)# ipv6 lisp itr-etr`
		Enables both the LISP ITR and the LISP ETR functionality. When both ITR and ETR functionality is being enabled on the same device, the configuration can be simplified by using this command instead of the { **ip** | **ipv6** } **lisp itr** and { **ip** | **ipv6** } **lisp etr** commands separately.

<table>
<tr><td>Step 5</td><td>{ip | ipv6} lisp itr map-resolver map-resolver-address
Example:
<code>switch(config)# ip lisp itr map-resolver 10.10.10.1</code>
Example:
<code>switch(config)# ipv6 lisp itr map-resolver 10.10.10.1</code></td><td>Configures the locator address of the Map-Resolver to which this router sends Map-Request messages for IPv4 or IPv6 EIDs.
Note The locator address of the Map-Resolver can be an IPv4 or IPv6 address. See the Inspur CN12700 Series INOS LISP Command Reference for more details.</td></tr>
<tr><td>Step 6</td><td>{ip | ipv6} database-mapping EID-prefix/prefixlength locator prioritypriority weight weight
Example:
<code>switch(config)# ip lisp database-mapping 10.10.10.0/24
172.16.1.1 priority 1 weight 100</code>
Example:
<code>switch(config)# ipv6 lisp database-mapping
2001:db8:bb::/48 172.16.1.1 priority 1 weight 100</code></td><td>Configures an EID-to-RLOC mapping relationship and associated traffic policy for all IPv4 or IPv6 EID prefix(es) for this LISP site.
Note If the site has multiple locators associated with the same EID-prefix block, enter multiple ip lisp database-mapping commands to configure all of the locators for a given EID-prefix block.
If the site is assigned multiple EID-prefix blocks, enter the ip lisp database-mapping command for each EID-prefix block assigned to the site and for each locator by which the EID-prefix block is reachable.
If the site has multiple ETRs, you must configure all ETRs with the ip lisp database-mapping and ipv6 lisp database-mapping commands ensuring the options used are consistent.</td></tr>
<tr><td>Step 7</td><td>{ip | ipv6} lisp etr map-server map-server-address key key-type authentication-key
Example:
<code>switch(config)# ip lisp etr map-server 172.16.1.2
key 0 123456789</code>
Example:
<code>switch(config)# ipv6 lisp etr map-server 172.16.1.2
key 0 123456789</code></td><td>Configures the locator address of the LISP Map-Server to which this router, acting as an IPv4 or IPv6 LISP ETR, registers.
Note The Map-Server must be configured with EID prefixes that match the EID-prefixes configured on this ETR, and a key matching the one configured on this ETR.
The locator address of the Map-Server may be an IPv4 or IPv6 address. See the Inspur CN12700 Series INOS LISP Command Reference for more details.</td></tr>
</table>

Step 8	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 9	(Optional) **show** {**ip** | **ipv6**} **lisp**
	Example:
	`switch# show ip lisp`
	Example:
	`switch# show ipv6 lisp`
		Displays all configured IPv4 or IPv6 LISP configuration parameters.

What to do next

Complete the optional LISP xTR parameters as needed.

Configuring Optional LISP ITR/ETR (xTR) Functionality

You can configure optional capability for the LISP xTR.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	(Optional) {**ip** | **ipv6**} **lisp etr accept-map-request-mapping** [**verify**]
	Example:
	`switch(config)# ip lisp etr accept-map-request verify`
	Example:
	`switch(config)# ipv6 lisp etr accept-map-request verify`
		Configures the LISP ETR to cache IPv4 or IPv6 mapping data contained in a Map-Request message received from the Map-Server on behalf of a LISP ITR.
		The **verify** keyword allows the mapping data to be cached but not used for forwarding packets until the ETR can send its own Map-Request to one of the locators from the mapping data record and receive a Map-Reply with the same data in response. By default, the router does not cache mapping data contained in a Map-Request message.

Step 3	(Optional) {**ip** | **ipv6**} **lisp ip lisp etr map-cache-ttl** *time-to-live*
	Example:
	`switch(config)# ip lisp etr map-cache-ttl 720`
	Example:
	`switch(config)# ipv6 lisp etr map-cache-ttl 720`
		Configures the time-to-live (TTL) value, in minutes, inserted into LISP Map-Reply messages sent by this ETR.

Step 4	(Optional) {**ip** | **ipv6**} **lisp** **map-cache-limit** *cache-limit* [**reserve-list** *list*]
	Example:
	`switch(config)# ip lisp map-cache-limit 2000`
	Example:
	`switch(config)# ipv6 lisp map-cache-limit 2000`
		Configures the maximum number of LISP map-cache entries allowed to be stored. By default, the LISP map-cache limit is 1000 entries.

Step 5	(Optional) {**ip** | **ipv6**} **lisp** **map-request-source** *source-address*
	Example:
	`switch(config)# ip lisp map-request-source 172.16.1.1`
	Example:
	`switch(config)# ipv6 lisp map-request-source 2001:db8:0a::1`
		Configures the address to be used as the source address for LISP Map-Request messages. By default, one of the locator addresses configured with the **ip lisp database-mapping** or **ipv6 lisp database-mapping** command is used as the default source address for LISP Map-Request messages.

Step 6	(Optional) {**ip** | **ipv6**} **lisp** **path-mtu-discovery** {**min** *lower-bound* | **max** *upper-bound*}
	Example:
	`switch(config)# ip lisp path-mtu-discovery min 1200`
	Example:
	`switch(config)# ipv6 lisp path-mtu-discovery min 1200`
		Configures the minimum and maximum MTU settings for the LISP router for path-mtu-discovery. By default, path-mtu-discovery is enabled by the LISP router.
		Caution Disabling the use of path-mtu-discovery is not recommended.

Step 7	(Optional) [**no**] **lisp loc-reach-algorithm** {**tcp-count** | **echo-nonce** | **rloc-probing**} **Example:** `switch(config)# lisp loc-reach-algorithm rloc-probing`	Enables or disables the use of a LISP locator reachability algorithm. Locator reachability algorithms are address-family independent. By default, all locator reachability algorithms are disabled.

Step 8	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 9	(Optional) **show** {**ip** | **ipv6**} **lisp**
	Example:
	`switch# show ip lisp`
	Example:
	`switch# show ipv6 lisp`
		Displays all configured IPv4 or IPv6 LISP configuration parameters.

1.7.3 Configuring LISP-ALT Functionality

You can enable and configure LISP-ALT (ALT) functionality for both IPv4 and IPv6 address families.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	{**ip** | **ipv6**} **lisp alt-vrf** *vrf-name*
	Example:
	`switch(config)# ip lisp alt-vrf lisp`
	Example:
	`switch(config)# ipv6 lisp alt-vrf lisp`
		Configures LISP to use the LISP-ALT VRF vrf-name.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 4	(Optional) **show** {**ip** | **ipv6**} **lisp**
	Example:
	`switch# show ip lisp`
	Example:
	`switch# show ipv6 lisp`
		Displays all configured IPv4 or IPv6 LISP configuration parameters.

1.7.4 Configuring Required LISP Map-Resolver Functionality

You can enable and configure LISP Map-Resolver (MR) functionality for both IPv4 and IPv6 address families.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	{**ip** | **ipv6**} **lisp map-resolver**
	Example:
	`switch(config)# ip lisp map-resolver`
	Example:
	`switch(config)# ipv6 lisp map-resolver`
		Enables LISP Map-Resolver functionality on the device.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 4	(Optional) **show** {**ip** | **ipv6**} **lisp**
	Example:
	`switch# show ip lisp`
	Example:
	`switch# show ipv6 lisp`
		Displays all configured IPv4 or IPv6 LISP configuration parameters.

1.7.5 Configuring LISP Map-Server Functionality

Configuring Required LISP Map-Server Functionality

You can enable and configure LISP Map-Server (MS) functionality for both IPv4 and IPv6 address families.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	{**ip** | **ipv6**} **lisp map-server**
	Example:
	`switch(config)# ip lisp map-server`
	Example:
	`switch(config)# ipv6 lisp map-server`
		Enables LISP Map-Server functionality on the device.

Step 3	**lisp site** *site-name*
	Example:
	`switch(config)# lisp site Customer1`
	`switch(config-lisp-site)#`
		Creates the site name and enters LISP site configuration mode.

Step 4	**description** *description* **Example:** `switch(config-lisp-site)# description LISP Site Customer1`	Enters a description for the LISP site being configured.
Step 5	**authentication-key** *key-type password* **Example:** `switch(config-lisp-site)# authentication-key 0 123456789`	Enters the authentication key type and password for the LISP site being configured. **Note** The password must match the one configured on the ETR in order for the ETR to successfully register.

Step 6	**eid-prefix** *EID-prefix* [**route-tag** *tag*]
	Example:
	`switch(config-lisp-site)# eid-prefix 192.168.1.0/24 route-tag 12345`
	Example:
	`switch(config-lisp-site)# eid-prefix 2001:db8:aa::/48 route-tag 12345`
		Enters the EID-prefix for which the LISP site being configured is authoritative and optionally adds a route-tag.

Step 7	**end**
	Example:
	`switch(config-lisp-site)# end`
	`switch#`
		Exits LISP site configuration mode.

Step 8	(Optional) **show** {**ip** | **ipv6**} **lisp**
	Example:
	`switch# show ip lisp`
	Example:
	`switch# show ipv6 lisp`
		Displays all configured IPv4 or IPv6 LISP configuration parameters.

What to do next

Complete the optional LISP Map-Server configuration items as needed.

1.7.6 Configuring Optional LISP Map-Server Functionality

You can configure optional LISP Map-Server functionality.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**lisp site** *site-name*
	Example:
	`switch(config)# lisp site Customer1`
	`switch(config-lisp-site)#`
		Enters LISP site configuration mode for the indicated site. If the site does not exist, it will be created.

Step 3	(Optional) **allowed-locators** *rloc1* [*rloc2* [...]] **Example:** `switch(config-lisp-site)# allowed-locators 172.16.8.1 2001:db8:aa::1`	Enters the locators that are to be allowed to be included in the Map-Register message for the LISP site being configured. **Note** When the **allowed-locators** command is configured, all locators listed on the Map-Server within the LISP site configuration must also appear in the Map-Register message sent by the ETR for the Map-Register message to be accepted.

Step 4	**end**
	Example:
	`switch(config-lisp-site)# end`
	`switch#`
		Exits LISP site configuration mode.

Step 5	(Optional) **show** {**ip** | **ipv6**} **lisp**
	Example:
	`switch# show ip lisp`
	Example:
	`switch# show ipv6 lisp`
		Displays all configured IPv4 or IPv6 LISP configuration parameters.

1.7.7 Configuring Required LISP Proxy-ITR Functionality

You can enable and configure LISP Proxy-ITR functionality for both IPv4 and IPv6 address families.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	{**ip** | **ipv6**} **proxy-itr** *locator* [*other-address-family-locator*]
	Example:
	`switch(config)# ip lisp proxy-itr 172.16.8.1`
	Example:
	`switch(config)# ipv6 lisp proxy-itr 2001:db8:aa::1`
		Configures LISP Proxy-ITR functionality on the device. The *locator* address is used as a source address for encapsulating data packets or Map-Request messages. Optionally, you can provide an address for the other address family (for example, IPv6 for the **ip proxy-itr** command).

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 4	(Optional) **show** {**ip** | **ipv6**} **lisp**
	Example:
	`switch# show ip lisp`
	Example:
	`switch# show ipv6 lisp`
		Displays all configured IPv4 or IPv6 LISP configuration parameters.

1.7.8 Configuring Required LISP Proxy-ETR Functionality

You can enable and configure LISP Proxy-ETR functionality for both IPv4 and IPv6 address families.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	{**ip** | **ipv6**} **proxy-etr**
	Example:
	`switch(config)# ip lisp proxy-etr`
	Example:
	`switch(config)# ipv6 lisp proxy-etr`
		Configures LISP Proxy-ETR functionality.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 4	(Optional) **show** {**ip** | **ipv6**} **lisp**
	Example:
	`switch# show ip lisp`
	Example:
	`switch# show ipv6 lisp`
		Displays all configured IPv4 or IPv6 LISP configuration parameters.

Related Topics

Configuring LISP-ALT Functionality.

1.8 Additional References

This section includes additional information related to implementing LISP.

1.8.1 Related Documents

Related Topic	Document Title
Inspur INOS licensing	*Inspur INOS Licensing Guide*

1.8.2 Standards

Standard	Title
No new or modified standards are supported by this release.	

1.9 Feature History for LISP

Table 2 : Feature History for LISP

Feature Name	Releases	Feature Information
LISP-ALT functionality	8.3(1)	This functionality is no longer required to configure other LISP features.
Locator/ID Separation Protocol (LISP)	8.3(1)	This feature is introduced.

CHAPTER 2 Configuring LISP ESM Multihop Mobility

This chapter describes how to configure the Extended Subnet Mode (ESM) multihop mobility feature to separate the Locator/ID Separation Protocol (LISP) dynamic host detection function from the LISP encapsulation/decapsulation function within a LISP topology.

This chapter contains the following sections:

- Finding Feature Information.
- Information About LISP ESM Multihop Mobility.
- Licensing Requirements for LISP.
- Guidelines and Limitations for LISP ESM Multihop Mobility.
- Default Settings for LISP.
- Configuring LISP ESM Multihop Mobility.
- Configuration Examples for LISP ESM Multihop Mobility.
- Additional References.
- Feature Information for LISP ESM Multihop Mobility.

2.1 Finding Feature Information

Your software release might not support all the features documented in this module. 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

2.2 Information About LISP ESM Multihop Mobility

2.3 Licensing Requirements for LISP

The following table shows the LISP licensing requirements:

Product	**License Requirement**
Inspur INOS	This feature requires the LAN_ENTERPRISE_SERVICES_PKG license. For a complete explanation of the Inspur INOS licensing scheme, see the *Inspur INOS Licensing Guide*.

2.4 Guidelines and Limitations for LISP ESM Multihop Mobility

LISP ESM multihop mobility has the following guidelines and limitations:

- Locator/ID Separation Protocol (LISP) multihop mobility is supported only in Extended Subnet Mode (ESM) and it is recommended in combination with Overlay Transport Virtualization (OTV).
- ESM multihop mobility requires OTV First Hop Redundancy Protocol (FHRP) isolation to avoid hair-pinning of traffic across the OTV Data Center Interconnect (DCI) framework.
- ESM multihop mobility does not support Network Address Translated (NAT'd) endpoint identifiers (EIDs).
- To properly route traffic between extended VLANs when the source and destination hosts are detected by FHRs at different data centers, we recommend one of the following designs:
    - Establish a routing protocol adjacency between the first-hop routers (FHRs) in the different data centers over a dedicated extended VLAN; redistribute host routes from LISP into the routing protocol for discovered hosts at each data center FHR.
    - Separate each mobile VLAN in a VRF and configure the LISP FHR within the related virtual routing and

forwarding (VRF) context. Set up an external site gateway xTR to act as router for all of the mobile VLANs (VRFs).

2.5 Default Settings for LISP

This table lists the default settings for LISP parameters.

Table 3 : LISP Default Settings

Parameters	Default
feature lisp command	Disabled

2.6 Configuring LISP ESM Multihop Mobility

This section includes the following topics:

2.6.1 Configuring the First-Hop Device

Before you begin

- Ensure that LISP is enabled on the Inspur INOS device.
- Ensure that you are in the correct VDC.
- Ensure that you have enabled the VLAN interfaces feature.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **ip lisp etr**	Configures a Inspur INOS device to act as an IPv4 Locator/ID Separation Protocol (LISP) Egress Tunnel Router (ETR),
Step 3	(Optional) switch(config)# **vrf context** *vrf-name*	Creates a new VRF and enters VRF configuration mode to configure the first-hop router (FHR) function within the specified VRF routing context instead of using the default VRF. The value of the *vrf- name* is any case-sensitive, alphanumeric string of up to 32 characters. **Note** This approach implements a mobility design where eachmobile VLAN is a member of a distinct VRF and an external site gateway xTR acts as router for all of the mobile VLANs (VRFs).

Step 4	switch(config)# **lisp dynamic-eid** *dynamic-EID-policy-name*	Configures a LISP Virtual Machine (VM) Mobility (dynamic-EID roaming) policy and enters the LISP dynamic-EID configuration mode.
Step 5	switch(config-lisp-dynamic-eid)# **database-mapping** *dynamic-EID-prefix locator* **priority** *priority* **weight** *weight*	Configures a IPv4 or IPv6 dynamic-endpoint identifier to Routing Locator (EID-to-RLOC) mapping relationship and its associated traffic policy. **Note** If you configured the **vrf context** command, the IP prefix specified for the *dynamic-EID-prefix locator* argument must belong to a local interface that is member of the same VRF.
Step 6	(Optional) switch(config-lisp-dynamic-eid)# **instance-id** *iid*	Configures an association between the dynamic EID policy and a LISP Instance ID. The *iid* must match the instance ID configured on the gateway xTR. The range is from 1 to 16777215. The default value is 0.
Step 7	switch(config-lisp-dynamic-eid)# **eid-notify** *ip-address* **key** *password*	Enables sending of dynamic endpoint identifier (EID) presence notifications to a gateway xTR with the specified IP address along with the authentication key used with the gateway xTR.
Step 8	switch(config-lisp-dynamic-eid)# **map-notify-group** *ipv4-group-address*	Configures a discovering LISP-Virtual Machine (VM) switch to send a Map-Notify message to other LISP-VM switches within the same data center site so that they can also determine the location of the dynamic-EID.
Step 9	Repeat the preceding steps for each first-hop device to be configured.	—
Step 10	switch(config-lisp-dynamic-eid)# **exit**	Exits the LISP dynamic-EID configuration mode and returns to global configuration mode.
Step 11	switch (config)# **interface vlan***vlan-id*	Creates or modifies a VLAN and enters interface configuration mode.
Step 12	(Optional) switch(config)# **vrf member** *vrf-name*	This step is required if you configured the **vrf context** command. Adds the interface being configured to a VRF when the FHR is configured within a VRF context.
Step 13	switch(config-if)# **lisp mobility** *dynamic-EID-policy-name*	Configures an interface on an Ingress Tunnel Router (ITR) to participate in Locator/ID Separation Protocol (LISP) virtual machine (VM)-mobility (dynamic-EID roaming) for the referenced dynamic-EID policy.

Step 14	switch(config-if)# **lisp-extended subnet-mode**	Configures an interface to create a dynamic-endpoint identifier (EID) state for hosts attached on their own subnet in order to track the movement of EIDs from one part of the subnet to another part of the same subnet.
Step 15	switch(config-if)# **ip router ospf** *instance-tag* **area** *area-id*	Species the Open Shortest Path First (OSPF) instance and area for an interface
Step 16	switch(config-if)# **ip ospf passive-interface**	Suppresses Open Shortest Path First (OSPF) routing updates on an interface to avoid establishing adjacency over the LAN extension.
Step 17	switch(config-if)# **hsrp** *group-number*	Enters Hot Standby Router Protocol (HSRP) configuration mode and creates an HSRP group.
Step 18	switch(config-if-hsrp)# **ip address** *ip-address*	Creates a virtual IP address for the HSRP group. The IP address must be in the same subnet as the interface IP address.
Step 19	Repeat the preceding steps for each interface to be configured for multihop mobility.	—
Step 20	switch(config-if-hsrp)# **end**	Returns to privileged EXEC mode.

2.6.2 Configuring the Site Gateway xTR

Before you begin

- Ensure that LISP is enabled on the Inspur INOS device.
- Ensure that you are in the correct VDC.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	(Optional) switch# **lisp instance-id** *iid*	Configures an association between a VRF or the default VRF and a LISP instance ID. The value of the instance ID configured on the FHR, Site Gateway xTR, MSMR, and remote xTR must match. This command modifies the value of the instance ID (iid) from the default (0) to the specified value. The range of the *iid* argument is from 1 to 16777215.
Step 3	switch(config)# **ip lisp itr-etr**	Configures a Inspur INOS device to act as both an IPv4 LISP Ingress Tunnel Router (ITR) and Egress Tunnel Router (ETR), also known as an xTR.

Step 4	switch(config)# **ip lisp database-mapping** *EID-prefix* { *locator* | **dynamic** } **priority** *priority* **weight** *weight*	Configures an IPv4 endpoint identifier to Routing Locator (EID-to-RLOC) mapping relationship and its associated traffic policy.
Step 5	Repeat the preceding step for each locator.	`switch(config)# ip lisp database-mapping 192.168.0.0/16 10.0.1.2 priority 1 weight 5` `switch(config)# ip lisp database-mapping 192.168.0.0/16 10.0.2.2 priority 1 weight 5`
Step 6	switch(config)# **ip lisp itr map-resolver** *map-resolver-address*	Configures a Inspur INOS device to act as an IPv4 Locator/ID Separation Protocol (LISP) Map-Resolver (MR).
Step 7	switch(config)# **ip lisp etr map-server** *map-server-address* {[**key** *key-type authentication-key*] | **proxy-reply** }	Configures the IPv4 or IPv6 locator address of the Locator/ID Separation Protocol (LISP) Map-Server to be used by the egress tunnel router (ETR) when registering for IPv4 EIDs.
Step 8	switch(config)# **lisp dynamic-eid** *dynamic-EID-policy-name*	Configures a LISP Virtual Machine (VM) Mobility (dynamic-EID roaming) policy and enters the LISP dynamic-EID configuration mode.
Step 9	switch(config-lisp-dynamic-eid)# **database-mapping** *dynamic-EID-prefix locator* **priority** *priority* **weight** *weight*	Configures a IPv4 or IPv6 dynamic-endpoint identifier to Routing Locator (EID-to-RLOC) mapping relationship and its associated traffic policy.
Step 10	switch(config-lisp-dynamic-eid)# **eid-notify authentication-key** { **0** *unencrypted-password* | **6** *encrypted-password* | *password*}	Specifies an authentication key to validate the endpoint identifier (EID)-notify messages received from a device.
Step 11	Repeat the preceding three steps to enable sending EID presence notifications to each additional site gateway.	Exits LISP locator-set configuration mode and returns to LISP configuration mode.
Step 12	switch(config-lisp-dynamic-eid)# **end**	Returns to privileged EXEC mode.

2.6.3 Configuring xTR

Before you begin

- Ensure that LISP is enabled on the Inspur INOS device.
- Ensure that you are in the correct VDC.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.

Step 2	(Optional) switch# **lisp instance-id** *iid*	Configures an association between a VRF or the default VRF and a LISP instance ID. The value of the instance ID configured on the FHR, Site Gateway xTR, MSMR, and remote xTR must match. This command modifies the value of the instance ID (iid) from the default (0) to the specified value. The range of the *iid* argument is from 1 to 16777215.
Step 3	switch(config)# **ip lisp itr-etr**	Configures a Inspur INOS device to act as both an IPv4 LISP Ingress Tunnel Router (ITR) and Egress Tunnel Router (ETR), also known as an xTR.
Step 4	switch(config)# **ip lisp database-mapping** *EID-prefix* { *locator* | **dynamic** } **priority** *priority* **weight** *weight*	Configures an IPv4 endpoint identifier to Routing Locator (EID-to-RLOC) mapping relationship and its associated traffic policy.
Step 5	switch(config)# **ip lisp database-mapping** *EID-prefix* { *locator* | **dynamic** } **priority** *priority* **weight** *weight*	Configures an IPv4 endpoint identifier to Routing Locator (EID-to-RLOC) mapping relationship and its associated traffic policy.
Step 6	switch(config)# **ip lisp itr map-resolver** *map-resolver-address*	Configures a Inspur INOS device to act as an IPv4 Locator/ID Separation Protocol (LISP) Map-Resolver (MR).
Step 7	switch(config)# **ip lisp etr map-server** *map-server-address* {[**key** *key-type authentication-key*] | **proxy-reply** }	Configures the IPv4 or IPv6 locator address of the Locator/ID Separation Protocol (LISP) Map-Server to be used by the egress tunnel router (ETR) when registering for IPv4 EIDs.
Step 8	switch(config)# **exit**	Exits global configuration mode and returns to privileged EXEC mode.

2.6.4 Configuring the Map Server

Before you begin

- Ensure that LISP is enabled on the Inspur INOS device.
- Ensure that you are in the correct VDC.

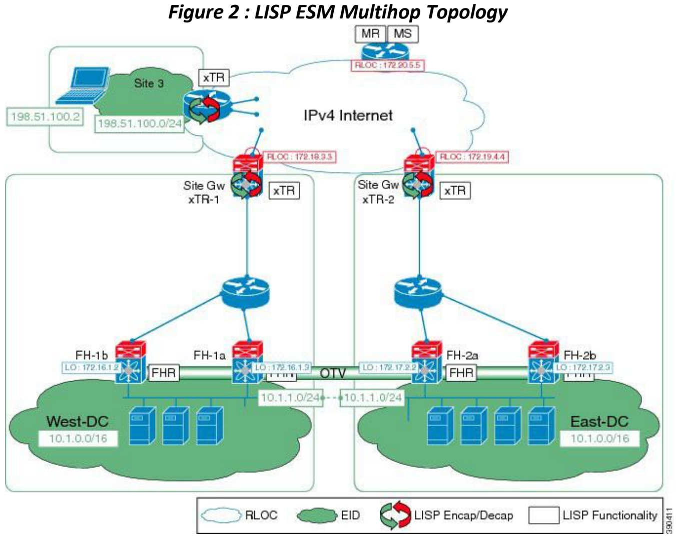
Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **ip lisp itr map-resolver** *map-resolver-address*	Configures a Inspur INOS device to act as an IPv4 Locator/ID Separation Protocol (LISP) Map-Resolver (MR).

Step 3	switch(config)# **ip lisp etr map-server** *map-server-address* {[**key** *key-type authentication-key*] | **proxy-reply** }	Configures the IPv4 or IPv6 locator address of the Locator/ID Separation Protocol (LISP) Map-Server to be used by the egress tunnel router (ETR) when registering for IPv4 EIDs.
Step 4	switch(config)# **lisp site** *site-name*	Configures a Locator/ID Separation Protocol (LISP) site and enter site configuration mode on a LISP Map-Server.
Step 5	switch(config-lisp-site)# **eid-prefix** [**instance-id** *iid*] { *EID-prefix* [**route-tag** *tag*]} [**accept-more-specifics**]	Configures a list of endpoint identifier (EID)-prefixes that are allowed in a Map-Register message sent by an egress tunnel router (ETR) when registering to the Map Server.
Step 6	switch(config-lisp-site)# **authentication-key** *key-type password*	Configures the password used to create the SHA-1 HMAC hash for authenticating the Map-Register message sent by an egress tunnel router (ETR) when registering to the Map-Server.
Step 7	Repeat the preceding three steps to configure each additional LISP site.	—
Step 8	switch(config-lisp-site)# **end**	Returns to privileged EXEC mode.

2.7 Configuration Examples for LISP ESM Multihop Mobility

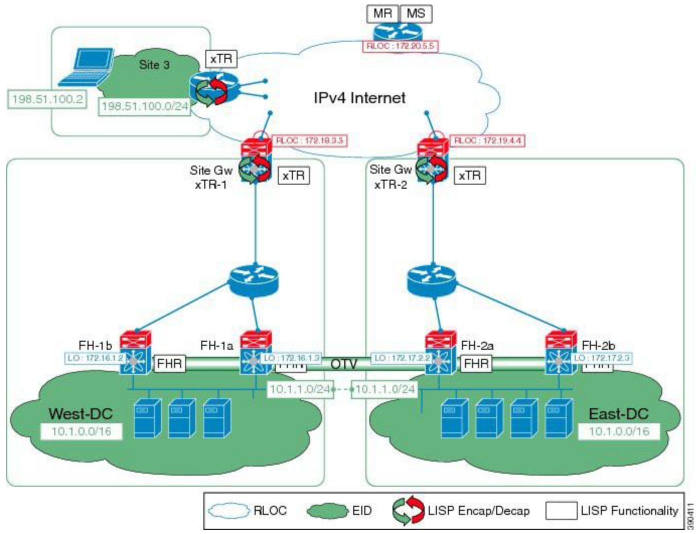
Figure 2 : LISP ESM Multihop Topology



This section includes the following examples for configuring the topology in the preceding figure:

2.7.1 Example: First-Hop Router Configuration

Figure 3 : Sample Topology



The following example shows how to configure the first hop "FH-1a" in the sample topology:

```
 ip lisp etr
 lisp dynamic-eid VLAN-11
  database-mapping 10.1.1.0/24 172.16.1.2 pr 10 w 50
  database-mapping 10.1.1.0/24 172.16.1.3 pr 10 w 50
  eid-notify 172.16.0.1 key 3 75095fe9112836e3
  map-notify-
 group 225.1.1.1
 lisp dynamic-eid
 VLAN-12
  database-mapping 10.1.2.0/24 172.16.1.2 pr 10 w 50
  database-mapping 10.1.2.0/24 172.16.1.3 pr 10 w 50
  eid-notify 172.16.0.1 key 3 75095fe9112836e3
  map-notify-group 225.1.1.2

 interface Vlan11
  lisp mobility
  VLAN-11 lisp
  extended-
  subnet-mode
  ip address
  10.1.1.3/24
  ip ospf passive-interface
  ip router ospf 100 area 0.0.0.1
  hsrp 1
ip 10.1.1.1

 interface Vlan12
  lisp mobility
  VLAN-12 lisp
  extended-
  subnet-mode


  ip address
  10.1.2.3/24
  ip ospf passive-
  interface
  ip router ospf 100 area 0.0.0.1
  hsrp 2
   ip 10.1.2.1
```

The following example shows how to configure the first hop "FH-2a" in the sample topology:

```
ip lisp etr
lisp dynamic-eid VLAN-
11
 database-mapping 10.1.1.0/24 172.17.2.2 pr 10 w 50
 database-mapping 10.1.1.0/24 172.17.2.3 pr 10 w 50
 eid-notify 172.17.0.1 key 3 6d018260cf71b07c
 map-notify-group
225.1.1.1 lisp
dynamic-eid VLAN-12
 database-mapping 10.1.2.0/24 172.17.2.2 pr 10 w 50
 database-mapping 10.1.2.0/24 172.17.2.3 pr 10 w 50
 eid-notify 172.17.0.1 key 3 6d018260cf71b07c
 map-notify-group
 225.1.1.2

interface Vlan11
 lisp mobility
 VLAN-11 lisp
 extended-subnet-
 mode ip address
 10.1.1.4/24
 ip ospf passive-
 interface
 ip router ospf 100 area 0.0.0.2
 hsrp 1
  ip 10.1.1.1

interface Vlan12
 lisp mobility
 VLAN-12 lisp
 extended-subnet-
 mode ip address
 10.1.2.4/24
 ip ospf passive-
 interface
 ip router ospf 100 area 0.0.0.2
 hsrp 2
  ip 10.1.2.1
```

The following additional configuration ensures that the FHRs can route traffic from other attached subnets to servers that belong to the mobile subnet site1 and are discovered in the opposite data center. For this purpose the FHRs are configured to establish an adjacency over a dedicated extended VLAN using a dedicated routing protocol instance and to redistribute host routes from LISP.

For FH-1a:

```
ip prefix-list DiscoveredServers seq 5 permit 10.1.0.0/22 ge 32

route-map LISP2EIGRP permit 10
 match ip address prefix-list DiscoveredServers

interface
 Vlan100
 no
 shutdown
 ip address
 10.255.0.1/30 ip
 router eigrp 100

router eigrp 100
 autonomous-system 100
 redistribute lisp route-map LISP2EIGRP
```

For FHA-2a:

```
ip prefix-list DiscoveredServers seq 5 permit 10.1.0.0/22 ge 32
```

```
route-map LISP2EIGRP permit 10
 match ip address prefix-list DiscoveredServers

interf
 ace
 Vlan1
 00 no
 shutd
 own
 ip address
 10.255.0.2/30
 ip router
 eigrp 100

router eigrp 100
 autonomous-system 100
 redistribute lisp route-map LISP2EIGRP
```

2.7.2 Example: Site Gateway xTR Configuration

The following example shows how to configure the site gateway "Site GW xTR-1" in the sample topology:

```
ip lisp itr-etr
ip lisp database-mapping 10.1.0.0/16 172.18.3.3 priority
10 weight 50 ip lisp itr map-resolver 172.20.5.5
ip lisp etr map-server 172.20.5.5 key 3
0b50279df3929e28 lisp dynamic-eid VLAN11
database-mapping 10.1.1.0/24 172.18.3.3 priority
 10 weight 50 eid-notify authentication-key 3
 75095fe9112836e3
 lisp dynamic-eid VLAN12
database-mapping 10.1.2.0/24 172.18.3.3 priority
 10 weight 50 eid-notify authentication-key 3
 75095fe9112836e3

interface
 Ethernet3/1
 description
 Inside DC West
 ip address
 172.16.0.1/30
 ip router ospf 1 area 0.0.0.1
```

The following example configuration is for the site gateway "Site GW xTR-2" in the sample topology:

```
ip lisp itr-etr
ip lisp database-mapping 10.2.2.0/24 172.19.4.4 priority
10 weight 50 ip lisp itr map-resolver 172.20.5.5
ip lisp etr map-server 172.20.5.5 key 3
0b50279df3929e28 lisp dynamic-eid VLAN11
database-mapping 10.1.1.0/24 172.19.4.4 priority
 10 weight 50 eid-notify authentication-key 3
 6d018260cf71b07c
 lisp dynamic-eid VLAN12
database-mapping 10.1.2.0/24 172.19.4.4 priority
 10 weight 50 eid-notify authentication-key 3
 6d018260cf71b07c

interface
 Ethernet3/1
 description
 Inside DC East
 ip address
 172.17.0.1/30
 ip router ospf 1 area 0.0.0.2
```

2.7.3 Example: xTR Configuration

The following example shows how to configure the xTR (at Site 3):

```
ip lisp itr-etr
ip lisp database-mapping 198.51.100.0/24 172.21.1.5 priority
10 weight 50 ip lisp itr map-resolver 172.20.5.5
ip lisp etr map-server 172.20.5.5 key 3 0b50279df3929e28
```

2.7.4 Example: MSMR Configuration

The following example shows how to configure the map server map resolver (MSMR) device in the sample topology:

```
ip lisp map-
resolver ip
lisp map-
server lisp
site
roaming1
 eid-prefix 10.1.0.0/16 accept-more-
 specifics authentication-key 3
 0b50279df3929e28
lisp site site2
 eid-prefix 10.2.2.0/24
 authentication-key 3
 0b50279df3929e28
lisp site site3
 eid-prefix 198.51.100.0/24
 authentication-key 3
 0b50279df3929e28
```

2.7.5 Example:Multi-Hop Mobility Interworking with Routing Protocols Configuration

The following example shows how to dynamically redistribute LISP host routes for discovered servers into OSPF at the first-hop router (FHR):

```
ip prefix-list lisp-pflist seq 10 permit
10.1.1.0/24 ge 32 route-map lisp-rmap permit 10
match ip address prefix-list
lisp-pflist router ospf 100
redistribute lisp route-map lisp-rmap
```

The following example shows how to automatically convert host routes from a routing protocol into LISP dynamic EID entries at a Site Gateway xTR (in lieu of an EID notification coming from a FHR):

```
ip lisp itr-etr
ip lisp database-mapping 10.1.0.0/16 172.18.3.3 priority 10
weight 50 ip lisp itr map-resolver 172.20.5.5
ip lisp etr map-server 172.20.5.5 key 3
0b50279df3929e28 lisp dynamic-eid site1
 database-mapping 10.1.1.0/24 172.18.3.3 priority 10
 weight 50 register-route-notifications
```

2.8 Additional References

This section includes additional information related to implementing LISP.

2.9 Feature Information for LISP ESM Multihop Mobility

Feature Name	Release	Feature Information
LISP ESM multihop mobility	8.3(1)	This feature was introduced. The LISP Extended Subnet Mode (ESM) Multihop Mobility feature separates the Locator/ID Separation Protocol (LISP) dynamic host detection function from the LISP encapsulation and decapsulation function within a LISP topology.
Dynamic-EID Route Import	8.3(1)	This feature was introduced. This feature provides the ability for a Site Gateway xTR to perform server presence detection upon receiving host routes updates.

CHAPTER 3 LISP Instance-ID Support

This chapter includes the following sections:

- Information about LISP Instance-ID Support.
- How to Configure LISP Instance-ID Support.
- Configuration Examples for LISP Instance-ID Support.

3.1 Information about LISP Instance-ID Support

3.1.1 Overview of LISP Instance ID

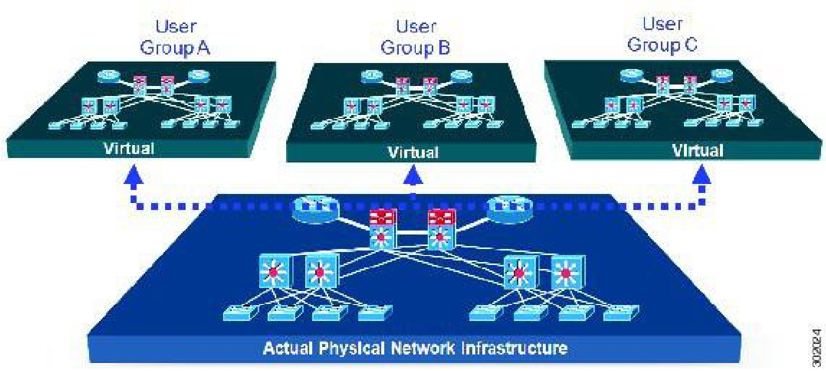
The LISP Instance ID provides a means of maintaining unique address spaces (or "address space segmentation") in the control and data plane. Instance IDs are numerical tags defined in the LISP canonical address format (LCAF). The Instance ID has been added to LISP to support virtualization.

When multiple organizations inside of a LISP site are using private addresses as Endpoint ID (EID) prefixes, their address spaces must remain segregated due to possible address duplication. An Instance ID in the address encoding can be used to create multiple segmented VPNs inside of a LISP site where you want to keep using EID-prefix-based subnets. The LISP Instance ID is currently supported in LISP ingress tunnel routers and egress tunnel routers (ITRs and ETRs, collectively known as xTRs), map server (MS) and map resolver (MR).

This chapter explains how to configure LISP xTRs with LISP MS and MR to implement virtualization. The content considers different site topologies and includes guidance to both shared and parallel LISP model configurations. It includes conceptual background and practical guidance, and provides multiple configuration examples.

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

Figure 4 : LISP Deployment Environment



When you plan the deployment of a LISP virtualized network environment, you must plan for virtualization at both the device level and the path level.

For path level virtualization: LISP binds virtual routing and forwarding (VRFs) to instance IDs (IIDs). These IIDs are included in the LISP header to provide data plane (traffic flow) separation.

For device level virtualization: Both the EID and the RLOC namespaces can be virtualized. The EID can be virtualized by binding a LISP instance ID to an EID VRF; the RLOC by tying locator addresses and associated mapping services to the specific VRF within which they are reachable.

Prerequisites for LISP Instance-ID Support

• Allow the use of instance-id 0's within a virtual routing and forwarding (VRF) instance.

Guidelines and Limitations for LISP Instance-ID Support

The LISP Instance-ID Support feature has the following configuration guidelines and restrictions:

• If you enable LISP, nondisruptive upgrade (ISSU) and nondisruptive downgrade (ISSD) paths are not supported. Disable LISP prior to any upgrade. This restriction applies only to releases before 8.3(1), not to 8.3(1) or subsequent LISP releases.

Device Level Virtualization

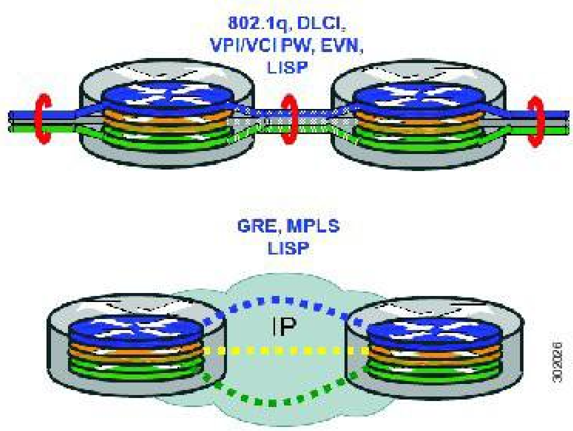
Virtualization at the device level uses virtual routing and forwarding (VRF) to create multiple instances of Layer 3 routing tables, as shown in the figure below. VRFs provide segmentation across IP addresses, allowing for overlapped address space and traffic separation. Separate routing, quality of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

Figure 5 : Device Level Virtualization

Path Level Virtualization

VRF table separation is maintained across network paths, as shown in the following figure. Single-hop path segmentation (hop by hop) is typically accomplished by using 802.1q VLANs, virtual path identifier/virtual circuit identifier password (VPI/VCI PW), or easy virtual network (EVN). You can also use the Locator ID Separation Protocol (LISP) in multihop mechanisms that include Multiprotocol Label Switching (MPLS) and generic routing encapsulation (GRE) tunnels. LISP binds VRF instances 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 6 : Path Level Virtualization



LISP Virtualization at the Device Level

LISP implements Locator ID separation and thereby creates two namespaces; endpoint ID (EID) and routing locator (RLOC). Either or both of these can be virtualized.

• EID virtualization—Enabled by binding a LISP instance ID to an EID virtual routing and forwarding (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.

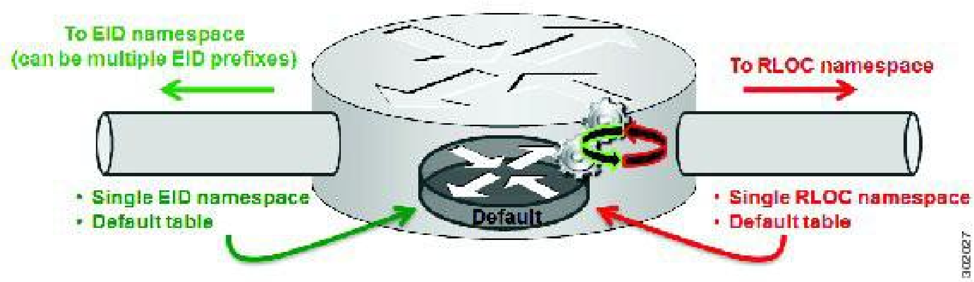
• Routing locator (RLOC) virtualization—Tying locator addresses and associated mapping services to the specific VRF within which they are reachable enables RLOC virtualization.

Because LISP can virtualize either or both of these namespaces, two models of operation are defined: the shared model and the parallel model. To understand how these models differ from the non-virtualized model of LISP, review information about the default (non-virtualized) model of LISP before reading about the shared model and the parallel model.

Default (Non-Virtualized) LISP Model

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

Figure 7 : Default (Nonvirtualized) LISP Model

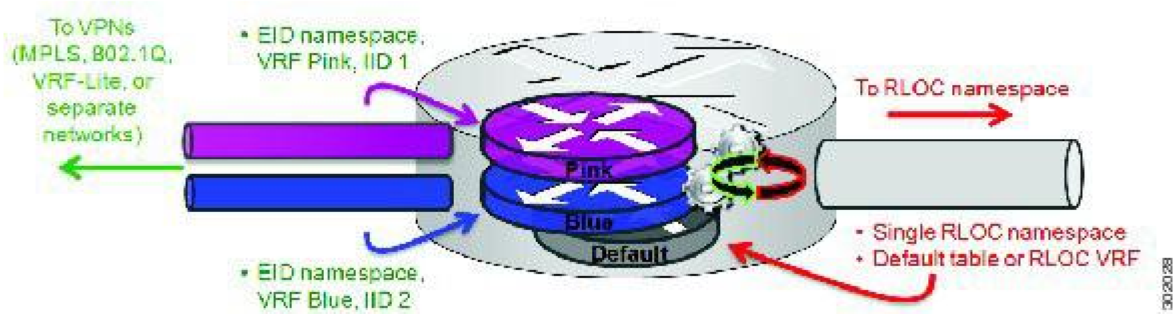


The mapping system must also be reachable through the default table. This default model can be thought of as a single instantiation of the parallel model of LISP virtualization where EID and RLOC addresses are within the same namespace.

3.1.2 LISP Shared Model Virtualization

A LISP shared model virtualized EID space is created when you bind VRFs associated with an EID space to Instance IDs. A common, shared locator space is used by all virtualized EIDs.

Figure 8 : LISP Shared Model Virtualization resolves EIDs within VRFs tied to Instance IDs. The default (global) routing table is the shared space.

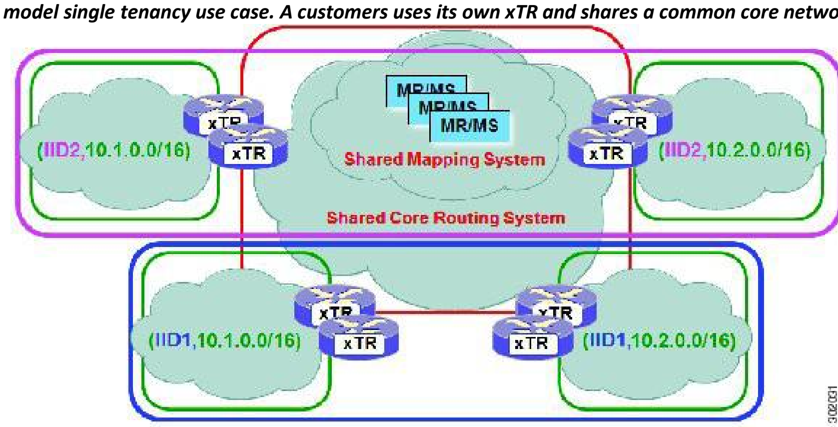


As shown in the figure, EID space is virtualized through its association with VRFs, and these VRFs are tied to LISP Instance IDs to segment the control plane and data plane in LISP. A common, shared locator space, the default (global) table, is used to resolve RLOC addresses for all virtualized EIDs.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

3.1.3 LISP Shared Model Virtualization Architecture

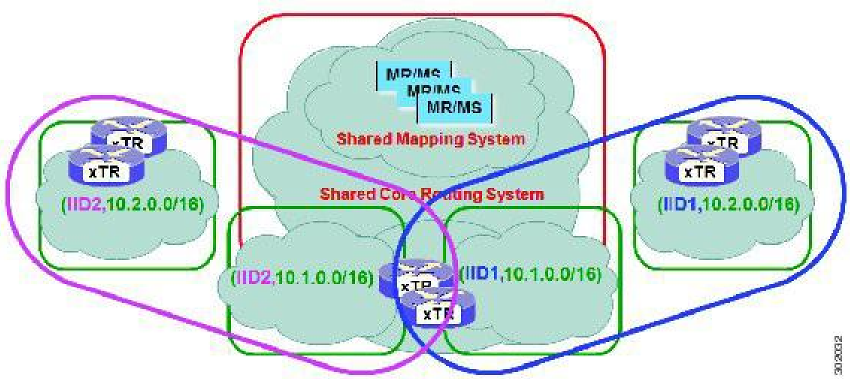
You can deploy the LISP shared model virtualization in single or multitenancy configurations. In the shared model single tenancy case, ingress and egress tunnel routers (xTRs) are dedicated to a customer but share infrastructure with other customers. Each customer and all sites associated with an xTR use the same instance ID and are part of a VPN using their own EID namespace. LISP instance IDs segment the LISP data plane and control plane. See the following figure.

Figure 9 : LISP shared model single tenancy use case. A customers uses its own xTR and shares a common core network and mapping system.



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. LISP instance IDs segment the LISP data plane and control plane. See the following figure.

Figure 10: LISP shared model multitenancy use case. Customer's use shared xTRs and share a common core network and mapping system.



3.1.4 LISP Shared Model Virtualization Implementation Considerations and Caveats

When you use the LISP Shared Model, instance IDs must be unique to an EID VRF.

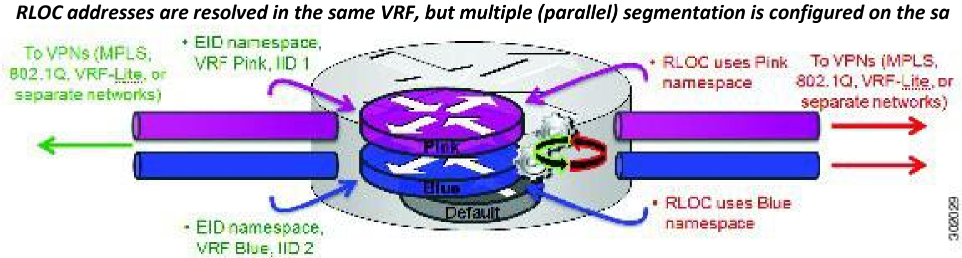
```
xTR-1# configure terminal
xTR-1(config)# vrf context alpha
xTR-1(config-vrf)# lisp instance-id 101
xTR-1(config-vrf)# exit
xTR-1(config)# vrf context beta
xTR-1(config-vrf)# lisp instance-id 101
Instance-ID 101 is already assigned to VRF context alpha
```

In the example, two EID VRFs are created: alpha and beta. In global configuration mode, a VRF named alpha is specified and associated with the instance ID 101. Next, a VRF named beta is specified and also associated with the instance ID 101. This configuration is not permissible because instance ID 101 is already associated with the VRF context named alpha. That is, you cannot connect the same instance ID to more than one EID VRF.

3.1.5 LISP Parallel Model Virtualization

The LISP parallel model virtualization ties the virtualized EID space associated with VRFs to RLOCs that are associated with the same or different VRFs (see the following figure).

Figure 11: LISP parallel model virtualization resolves an EID and associated RLOCs within the same or a different VRF. In this example, both EID and RLOC addresses are resolved in the same VRF, but multiple (parallel) segmentation is configured on the sa

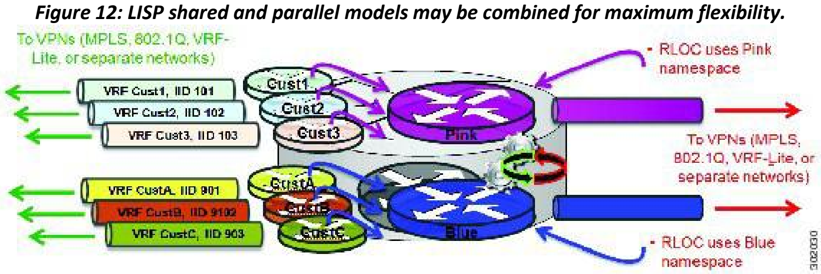


EID space is virtualized through its association with VRFs, and these VRFs are tied to LISP Instance IDs to segment the control plane and data plane in LISP. A common, "shared" locator space, the default (global) table is used to resolve RLOC addresses for all virtualized EIDs. The mapping system must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

In the figure, virtualized EID space is associated with a VRF (and bound to an Instance ID) that is tied to locator space associated with the same VRF, in this case - Pink/Pink and Blue/Blue. However, this is not required; the EID VRF does not need to match the RLOC VRF. In any case, a mapping system must be reachable through the associated locator space. Multiple parallel instantiations can be defined.

A shared model and parallel model can be combined such that multiple EID VRFs share a common RLOC VRF, and multiple instantiations of this architecture are implemented on the same platform, as shown in the following figure.

Figure 12: LISP shared and parallel models may be combined for maximum flexibility.



3.1.6 LISP Parallel Model Virtualization Architecture

You can deploy LISP parallel model virtualization in single or multitenancy configurations. In the parallel model multitenancy case, a set of xTRs is shared (virtualized) among multiple customers, and each customer uses their own private (segmented) core infrastructure and mapping system. All sites associated with the customer use the same instance ID and are part of a VPN using their own EID namespace, as shown in the following figure.

Figure 13: LISP parallel model multitenancy case. Shared xTRs use virtualized core networks and mapping systems. LISP instance IDs segment the LISP data plane and control plane.

3.1.7 LISPParallel Model Virtualization Implementation Considerationsand Caveats

When you use LISP parallel model virtualization, each vrfvrf vrf-name instantiation is considered by a separate process. Instance IDs must be unique only within a vrf instantiation.

```
xTR-1# configure terminal
xTR-1(config)# vrf context alpha
xTR-1(config-vrf)# address-family ipv4 unicast
xTR-1(config-vrf-af-ipv4)# exit
xTR-1(config)# vrf context beta
xTR-1(config-vrf)# address-family ipv4 unicast
xTR-1(config-vrf-af-ipv4)# exit
xTR-1(config-vrf)# exit
xTR-1(config)# vrf context gamma
xTR-1(config-vrf)# address-family ipv4 unicast
xTR-1(config-vrf-af-ipv4)# exit
xTR-1(config-vrf)# exit
xTR-1(config)# vrf context delta
xTR-1(config-vrf)# address-family ipv4 unicast
xTR-1(config-vrf-af-ipv4)# exit
xTR-1(config-vrf)# exit
xTR-1(config)# vrf context alpha
xTR-1(config-vrf)# lisp instance-id 101
xTR-1(config-vrf)# exit
xTR-1(config)# vrf context gamma
xTR-1(config-vrf)# lisp instance-id 101
xTR-1(config-vrf)# exit
xTR-1(config)# vrf context beta
xTR-1(config-vrf)# lisp instance-id 201
The vrf beta table is not available for use as an EID table (in use by switch
lisp 1 EID instance 101 VRF)
```

In the above example, four VRFs are created: alpha, beta, gamma, and delta, as follows:

- The vrf instantiation device lisp 1 is created and associated with the VRF named alpha.
- The EID VRF named beta is specified and associated with instance ID 101.

• A new vrf instantiation, device lisp 3, is created and associated with the locator-table VRF named gamma.

• The EID table VRF named delta is specified and also associated with instance ID 101.

These two instance IDs are unrelated to each other; one is relevant only within device lisp 1, and the other is relevant only within device lisp 2.

In the example, note that under device lisp 2, the code requests a VRF instance named beta. Note that the device is unable to use this VRF instance because it (beta) is already associated with a vrf command within the device lisp 1 instantiation.

You can reuse an instance ID. The EID VRF into which it is decapsulated depends on the vrf instantiation with which it is associated. However, you cannot connect the same EID VRF to more than one VRF.

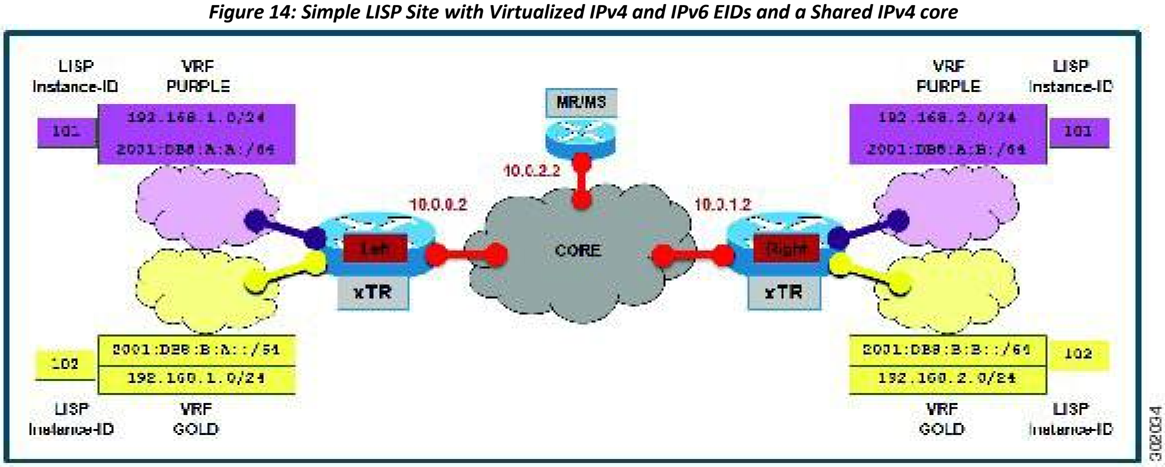
3.2 How to Configure LISP Instance-ID Support

3.2.1 Configuring Simple LISP Shared Model Virtualization

You can perform this task to enable and configure LISP ingress tunnel router/egress tunnel router (ITR/ETR) functionality (also known as xTR) with the LISP map server and map resolver, and thereby implement LISP shared model virtualization. This LISP shared model reference configuration is for a very simple two-site LISP topology, including xTRs and an map server/map resolver (MS/MR).

The following figure shows a basic LISP shared model virtualization solution. 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 VRF instances.

Figure 14: Simple LISP Site with Virtualized IPv4 and IPv6 EIDs and a Shared IPv4 core



In this figure, each LISP site uses a single edge switch that is 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) switch that is located in the network core within the shared RLOC address space.

Note

All IPv4 or IPv6 EID-sourced packets destined for 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
- a static route to Null0
- 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. Adding an IPv6 default route to Null0 ensures that all IPv6 packets are handled by LISP processing. (The use of the static route to Null0 is not strictly required, but is a LISP best practice.)

The components in the figure above are as follows:

LISP site

- The CPE functions as a LISP ITR and ETR (xTR).
- Both LISP xTRs have two VRFs: GOLD and PURPLE. Each VRF contains both IPv4 and IPv6
- EID-prefixes. A LISP instance ID is used to maintain separation between two VRFs. In this example, the share key is configured "per-site" and not "per-VRF." (Another configuration could configure the shared key 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 and is 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 following procedure (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

Before you begin, create the VRF instances by using the vrf definition command.

Before you begin

Create the VRFs using the **vrf definition** command.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**vrf context** *vrf-name* **Example:** `switch(config)# vrf context vrf1`	Enters VRF configuration submode.

Step 3	**ip lisp database-mapping** *EID-prefix/prefix-length locator* **priority** *priority* **weight** *weight* **Example:** `switch(config-vrf)# ip lisp database-mapping 192.168.1.0/24 10.0.0.2 priority 1 weight 100`	Configures an IPv4 EID-to-RLOC mapping relationship and its associated traffic policy for this LISP site. **Note** In this example, a single IPv4 EID prefix, 192.168.1.0/24, is being associated with the single IPv4 RLOC 10.0.0.2.
Step 4	Repeat Step 3 until all EID-to-RLOC mappings for the LISP site are configured. **Example:** `switch(config-vrf)# ipv6 lisp database-mapping 2001:db8:b:a::/64 10.0.0.2 priority 1 weight 100`	Configures an IPv6 EID-to-RLOC mapping relationship and its associated traffic policy for this LISP site.
Step 5	**ip lisp itr** **Example:** `switch(config-vrf)# ip lisp itr`	Enables LISP ITR functionality for the IPv4 address family.
Step 6	**ip lisp etr** **Example:** `switch(config-vrf)# ip lisp etr`	Enables LISP ETR functionality for the IPv4 address family.
Step 7	**ip lisp itr map-resolver** *map-resolver-address* **Example:** `switch(config-vrf)# ip lisp itr map-resolver 10.0.2.2`	Configures a locator address for the LISP map resolver to which this switch 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. **Note** You can configure up to two map resolvers if multiple map resolvers are available.

Step 8	**ip lisp etr map-server** *map-server-address* **key** *key-type authentication-key* **Example:** `switch(config-vrf)# ip lisp 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 switch, acting as an IPv4 LISP ETR, will use to register with the LISP mapping system. You must configure the map serve 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.
Step 9	**ipv6 lisp itr** **Example:** `switch(config-vrf)# ipv6 lisp itr`	Enables LISP ITR functionality for the IPv6 address family.
Step 10	**ipv6 lisp etr** **Example:** `switch(config-vrf)# ipv6 lisp etr`	Enables LISP ETR functionality for the IPv6 address family.
Step 11	**ipv6 lisp itr map-resolver** *map-resolver-address* **Example:** `switch(config-vrf)# ipv6 lisp itr map-resolver 10.0.2.2`	Configures a locator address for the LISP map resolver to which this switch 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 addresses. **Note** You can configure up to two map resolvers if multiple map resolvers are available.
Step 12	**ipv6 lisp etr map-server** *map-server-address* **key** *key-type authentication-key*	Configures a locator address for the LISP map-server and an authentication key that this

	Example: `switch(config-vrf)# ipv6 lisp etr map-server 10.0.2.2 key 0 Left-key`	switch, 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.
Step 13	**ip lisp locator-vrf default** **Example:** `switch(config-vrf)# ip lisp locator-vrf BLUE`	Configures a nondefault VRF table to be referenced by any IPv4 locators addresses.
Step 14	**ipv6 lisp locator-vrf default** **Example:** `switch(config-vrf)# ipv6 lisp locator-vrf default`	Configures a nondefault VRF table to be referenced by any IPv6 locator addresses.
Step 15	**exit** **Example:** `switch(config-vrf)# exit`	Exits VRF configuration mode and returns to global configuration mode.
Step 16	**ip lisp itr** **Example:** `switch(config)# ip lisp itr`	Enables LISP ITR functionality for the IPv4 address family.
Step 17	**ip lisp etr** **Example:** `switch(config)# ip lisp etr`	Enables LISP ETR functionality for the IPv4 address family.
Step 18	**ipv6 lisp itr** **Example:** `switch(config)# ipv6 lisp itr`	Enables LISP ITR functionality for the IPv6 address family.

Step 19	**ipv6 lisp etr** **Example:** `switch(config)# ipv6 lisp etr`	Enables LISP ETR functionality for the IPv6 address family.
Step 20	**ip route** *ipv4-prefix next-hop* **Example:** `switch(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. 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 21	**ipv6 route** *ipv6-prefix next-hop* **Example:** `switch(config)# ipv6 route ::/0 Null0`	Configures a default route to the upstream next hop for all IPv6 destinations. In this configuration example, because the xTR has only IPv4 RLOC connectivity, adding an IPv6 default route to Null0 ensures that all IPv6 packets are handled by LISP processing. (Use of the static route to Null0 is not strictly required, but is recommended as a LISP best practice.) If the destination is another LISP site, packets are LISP-encapsulated (using IPv4 RLOCs) to the remote site. If the destination is non-LISP, all IPv6 EIDs are LISP-encapsulated to a PETR (assuming one is configured).
Step 22	(Optional) **show running-config lisp** **Example:** `switch(config)# show running-config lisp`	Displays the LISP configuration on the switch.
Step 23	(Optional) **show** [**ip** | **ipv6**] **lisp** **Example:** `switch(config)# show ip lisp vrf TRANS`	The **show ip lisp** and **show ipv6 lisp** commands quickly verify the operational status of LISP as configured on the switch, as applicable to the IPv4 and IPv6 address families, respectively.
Step 24	(Optional) **show** [**ip** | **ipv6**] **lisp map-cache** [**vrf** *vrf-name*] **Example:** `switch(config)# show ip lisp map-cache`	The **show ip lisp map-cache** and **show ipv6 lisp map-cache** commands quickly verify the operational status of the map cache on a switch configured as an ITR or PITR, as applicable to the IPv4 and IPv6 address families, respectively.
Step 25	(Optional) **show** [**ip** | **ipv6**] **lisp database** [**vrf** *vrf-name*]	The **show ip lisp database** and **show ipv6 lisp database** commands quickly verify the operational status of the database mapping on

	Example: The following example shows IPv6 mapping database information for the VRF named GOLD. `switch(config)# show ipv6 lisp database vrf GOLD`	a switch configured as an ETR, as applicable to the IPv4 and IPv6 address families, respectively.
Step 26	(Optional) **show lisp site** [**name** *site-name*] **Example:** `switch(config)# show lisp site`	Displays the operational status of LISP sites as configured on a map server. This command applies only to a switch configured as a map server.

Step 27	**clear** [**ip** | **ipv6**] **lisp map-cache** [**vrf** *vrf-name*]
	Example:
	The first command displays IPv4 mapping cache information for vrf1. The second clears the mapping cache for vrf1 and shows the information after clearing the cache.
	`switch(config)# show ip lisp map-cache vrf vrf1`
	`switch(config)# clear ip lisp map-cache vrf vrf1`
		This command removes all IPv4 or IPv6 dynamic LISP map-cache entries stored by the switch, and displays the operational status of the LISP control plane. This command applies to a LISP switch that maintains a map cache (for example, if configured as an ITR or PITR).

3.2.2 Configuring a Private LISP Mapping System for LISP Shared Model Virtualization

You can perform this task to configure and enable standalone LISP map server/map resolver functionality for LISP shared model virtualization. In this procedure, you configure a switch as a standalone map server/map resolver (MR/MS) for a private LISP mapping system. Because the MR/MS is configured as a standalone switch, 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.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**lisp site** *site-name* **Example:**	Specifies a LISP site named LEFT and enters LISP site configuration mode.

	Command or Action	Purpose
	`switch(config)# lisp site LEFT`	**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 3	**authentication-key** [*key-type*] *authentication-key* **Example:** `switch(config-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 4	**eid-prefix** *EID-prefix* **instance-id** *instance-id* **Example:** `switch(config-lisp-site)# eid-prefix 192.168.1.0/24 instance-id 102`	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. **Note** 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 5	(optional) **eid-prefix** *EID-prefix* **instance-id** *instance-id* **Example:** `switch(config-lisp-site)# eid-prefix 2001:db8:a:b::/64 instance-id 102`	(optional) 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. This step is repeated here to configure an additional EID prefix under this LISP site. **Note** In this example, the IPv6 EID prefix 2001:db8:a:b::/64 and instance ID 102 are associated together.
Step 6	**exit** **Example:** `switch(config-lisp-site)# exit`	Exits LISP site configuration mode and returns to global configuration mode.

	Command or Action	Purpose

Step 7	**ip lisp map-resolver ipv6 lisp map-resolver**
	Example:
	`switch(config)# ip lisp map-resolver`
	`switch(config)# ipv6 lisp map-resolver`
		Enables LISP map resolver functionality for EIDs in the IPv4 address family and in the IPv6 family..

Step 8	**ip lisp map-server ipv6 lisp map-server**
	Example:
	`switch(config)# ip lisp map-server`
	`switch(config)# ipv6 lisp map-server`
		Enables LISP map server functionality for EIDs in the IPv4 address family and in the IPv6 address family..

Step 9	(optional) **show running-config lisp** **Example:** `switch(config)# show running-config lisp`	Displays the LISP configuration on the switch.
Step 10	(optional) **show** [**ip** | **ipv6**] **lisp** **Example:** `switch(config)# show ip lisp vrf TRANS`	The **show ip lisp** and **show ipv6 lisp** commands display the operational status of LISP as configured on the switch, as applicable to the IPv4 and IPv6 address families respectively.
Step 11	(optional) **show** [**ip** | **ipv6**] **lisp map-cache** [**vrf** *vrf-name*] **Example:** `switch(config)# show ip lisp map-cache`	The **show ip lisp map-cache** and **show ipv6 lisp map-cache** commands display the operational status of the map cache on a switch configured as an ITR or PITR, as applicable to the IPv4 and IPv6 address families respectively.
Step 12	(optional) **show** [**ip** | **ipv6**] **lisp database** [**vrf** *vrf-name*] **Example:** The following example shows IPv6 mapping database information for the VRF named GOLD. `switch(config)# show ipv6 lisp database vrf GOLD`	The **show ip lisp database** and **show ipv6 lisp database** commands display the operational status of the database mapping on a switch configured as an ETR, as applicable to the IPv4 and IPv6 address families respectively.
Step 13	(optional) **show lisp site** [**name** *site-name*] **Example:** `switch(config)# show lisp site`	The **show lisp site** command displays the operational status of LISP sites, as configured on a map server. This command only applies to a switch configured as a map server.

	Command or Action	Purpose

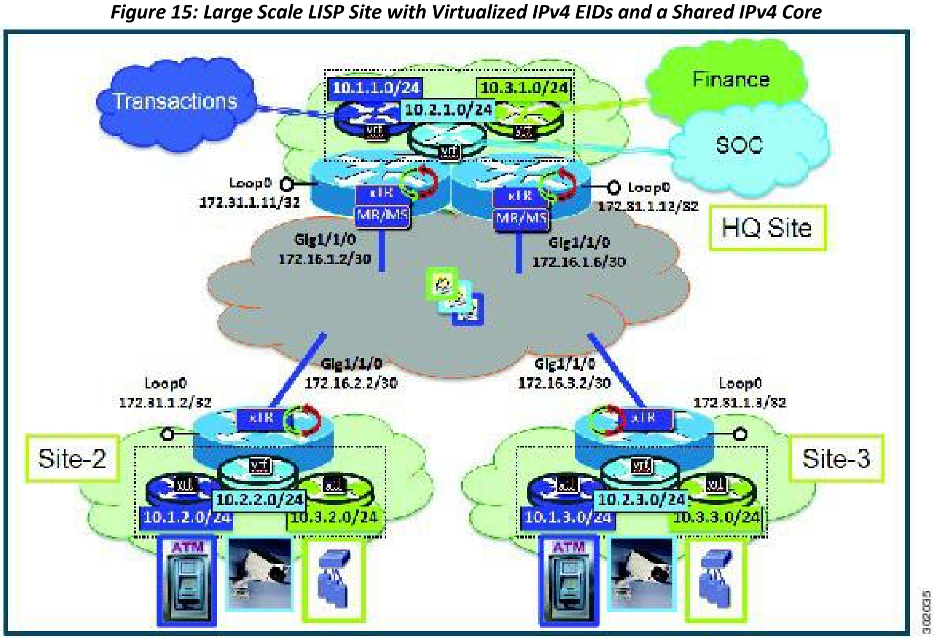
Step 14	**clear** [**ip** | **ipv6**] **lisp map-cache** [**vrf** *vrf-name*]
	Example:
	The first command displays IPv4 mapping cache information for vrf1. The second command clears the mapping cache for vrf1 and displays the updated status.
	`switch(config)# show ip lisp map-cache vrf vrf1`
	`switch(config)# clear ip lisp map-cache vrf vrf1`
		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 switch, respectively. They also show the operational status of the LISP control plane. This command applies to a LISP switch that maintains a map cache (for example, a switch configured as an ITR or PITR).

3.2.3 Configuring Large-Scale LISP Shared Model Virtualization

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

This procedure is for an enterprise that is deploying the LISP Shared Model where EID space is virtualized over a shared, common core network. A subset of the entire network is shown in the following figure. Three sites are shown: a multihomed "Headquarters" (HQ) site, and two remote office sites. The HQ site switches are deployed as xTRs and also as map resolver/map servers. The remote sites switches act as xTRs, and use the MS/MRs at the HQ site for LISP control plane support.

Figure 15: Large Scale LISP Site with Virtualized IPv4 EIDs and a Shared IPv4 Core



The components in the figure are as follows:

- **LISP site:**
  - Each customer premises equipment (CPE) switch 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. No overlapping prefixes are used; segmentation

between each VRF by LISP instance-ids makes this possible. Note that in this example, the separate authentication key is configured “per-vrf" and not “per-site", which affects both the xTR and MS configurations.

- The HQ LISP Site is multihomed 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 switch 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

Create the VRFs using the **vrf definition** command.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**lisp site** *site-name* **Example:** `switch(config)# lisp site TRANS`	Specifies a LISP site named TRANS and enters LISP site configuration mode. **Note** A LISP site name is significant to the local map server on which it is configured and has no relevance anywhere else. This site name serves solely as an administrative means of associating an EID-prefix or prefixes with an authentication key and other site-related mechanisms.
Step 3	**authentication-key** [*key-type*] *authentication-key* **Example:** `switch(config-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 4	**eid-prefix** *EID-prefix* / *prefix-length* **instance-id** *instance-id* **accept-more-specifics** **Example:** `switch(config-lisp-site)# eid-prefix 10.1.0.0/16 instance-id 1 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. The EID-prefix 10.1.0.0/16 is assumed to be an aggregate that covers all TRANS EID-prefixes at all LISP Sites. Use **accept-more-specifics** 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 5	**exit** **Example:** `switch(config-lisp-site)# exit`	Exits LISP site configuration mode and returns to LISP configuration mode.
Step 6	Repeat Steps 3 through 5 for each LISP site to be configured.	Repeat steps 3 through 5 for the site SOC and FIN as shown in the configuration example at the end of this procedure.
Step 7	**ip lisp map-resolver** **Example:** `switch(config)# ip lisp map-resolver`	Enables LISP map resolver functionality for EIDs in the IPv4 address family.
Step 8	**ip lisp map-server** **Example:** `switch(config)# ip lisp map-server`	Enables LISP map server functionality for EIDs in the IPv4 address family.
Step 9	**vrf context** *vrf-name* **Example:** `switch(config)# vrf context vrf1`	Enters VRF configuration submode.

Step 10	**database-mapping** *EID-prefix/prefix-length locator* **priority** *priority* **weight** *weight* **Example:** `switch(config-vrf)# 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. • 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. • Repeat Step 10 until all EID-to-RLOC mappings within this eid-table vrf and instance ID for the LISP site are configured.
Step 11	Repeat Step 10 until all EID-to-RLOC mappings within this EID table VRF and instance ID for the LISP site are configured.	
Step 12	**ip lisp etr map-server** *map-server-address* **key** *key-type authentication-key* **Example:** `switch(config-vrf)# ip lisp etr map-server 172.16.1.2 key 0 TRANS-key`	Configures a locator address for the LISP map server and an authentication key, which this switch, 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 in 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 can be an IPv4 or IPv6 address. Because each xTR has only IPv4 RLOC connectivity, the map server is reachable using its IPv4 locator addresses.

Step 13	**ip lisp itr map-resolver** *map-resolver-address* **Example:** `switch(config-vrf)# ip lisp itr map-resolver 172.16.1.2`	Configures a locator address for the LISP map resolver to which this switch will send map request messages for IPv4 EID-to-RLOC mapping resolutions. **Note** In this example, the map resolver is specified in switch lisp configuration mode and is 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 xTR is pointing to itself for mapping resolution (and to its neighbor xTR/MS/MR at the same site). **Note** 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. **Note** You can configure up to two map resolvers if multiple map resolvers are available.

Step 14	Repeat Step 13 to configure another locator address for the LISP map resolver **Example:** `switch(config-vrf)# ip lisp itr map-resolver 172.16.1.6`	Configures a locator address for the LISP map resolver to which this switch will send map request messages for IPv4 EID-to-RLOC mapping resolutions. **Note** 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). **Note** 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. **Note** You can configure up to two map resolvers if multiple map resolvers are available.
Step 15	**ip lisp itr** **Example:** `switch(config-vrf)# ip lisp itr`	Enables LISP ITR functionality for the IPv4 address family.
Step 16	**ip lisp etr** **Example:** `switch(config-vrf)# ip lisp etr`	Enables LISP ETR functionality for the IPv4 address family.
Step 17	**ip lisp locator-vrf default** **Example:** `switch(config-vrf)# ip lisp locator-vrf BLUE`	Configures a nondefault VRF table to be referenced by any IPv4 locators addresses.
Step 18	**ipv6 lisp locator-vrf default** **Example:** `switch(config-vrf)# ipv6 lisp locator-vrf default`	Configures a nondefault VRF table to be referenced by any IPv6 locator addresses.

Step 19	**exit** **Example:** `switch(config-vrf)# exit`	Exits VRF configuration mode and returns to global configuration mode.
Step 20	Repeat step 9 to 19 for all VRFs.	
Step 21	**ip route** *ipv4-prefix next-hop* **Example:** `switch(config)# ip route 0.0.0.00.0.0.0 172.16.1.1`	Configures a default route to the upstream next hop for all IPv4 destinations. **Note** 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 **Note** Packets are deemed to be a candidate for LISP encapsulation when they are sourced from a LISP EID and the destination is one of the following: • a current map-cache entry • a default route with a legitimate next-hop • a static route to Null0 • 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 22	(Optional) **show running-config lisp** **Example:** `switch(config)# show running-config lisp`	Displays the LISP configuration on the switch.
Step 23	(Optional) **show** [**ip** | **ipv6**] **lisp** **Example:** `switch(config)# show ip lisp vrf TRANS`	The **show ip lisp** and **show ipv6 lisp** commands are useful for quickly verifying the operational status of LISP as configured on the switch, as applicable to the IPv4 and IPv6 address families respectively.

Step 24	(Optional) **show** [**ip** | **ipv6**] **lisp map-cache** [**vrf** *vrf-name*] **Example:** `switch(config)# show ip lisp map-cache`	Displays the operational status of the map cache on a switch configured as an ITR or PITR, as applicable to the IPv4 and IPv6 address families.
Step 25	(Optional) **show** [**ip** | **ipv6**] **lisp database** [**vrf** *vrf-name*] **Example:** `switch(config)# show ipv6 lisp database vrf GOLD`	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 switch configured as an ETR, as applicable to the IPv4 and IPv6 address families. This example shows IPv6 mapping database information for a VRF named GOLD.
Step 26	(Optional) **show lisp site** [**name** *site-name*] **Example:** `switch(config)# show lisp site`	The **show lisp site** command verifies the operational status of LISP sites, as configured on a map server. This command only applies to a switch configured as a map server.

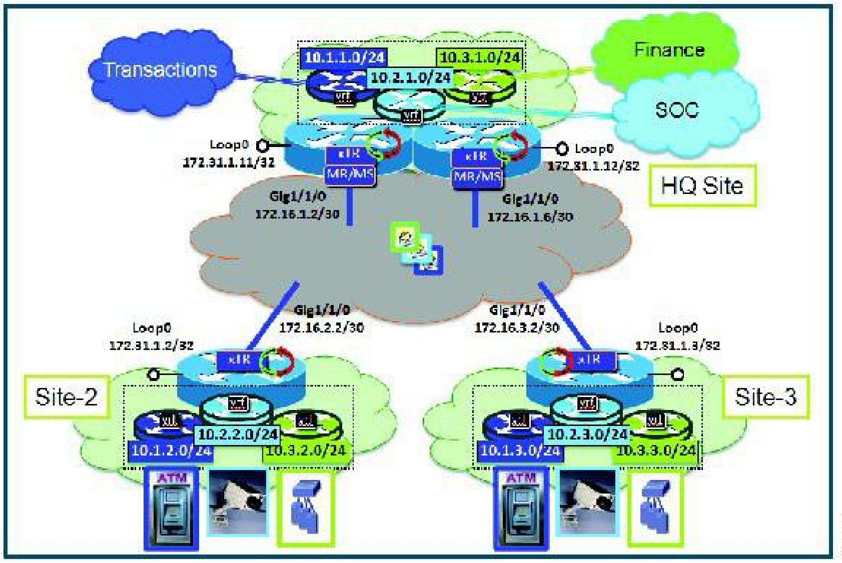
Step 27	(Optional) **clear** [**ip** | **ipv6**] **lisp map-cache** [**vrf** *vrf-name*]
	Example:
	`switch(config)# show ip lisp map-cache vrf vrf1`
	`switch(config)# clear ip lisp map-cache vrf vrf1`
		The **clear ip lisp map-cache** and **clear ipv6 lisp map-cache** commands remove all IPv4 or IPv6 dynamic LISP map-cache entries stored by the switch. They verify the operational status of the LISP control plane. The command applies to a LISP switch that maintains a map cache (for example, a switch configured as an ITR or PITR).
		The first command in the example displays IPv4 mapping cache information for vrf1. The second command clears the mapping cache for vrf1 and displays the status information after clearing the cache.

3.2.4 Configuring a Remote Site for Large-Scale LISP Shared Model Virtualization

You can 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.

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

Figure 16 : 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 customer premises equipment (CPE) switch at a remote site functions as a LISP ITR and ETR (xTR).
  - Each LISP xTR has the same three VRFs as the HQ Site: the TRANS (for transactions), the SOC (for security operations), and the FIN (for financials). 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

Create the VRFs using the **vrf definition** command and verify that the Configure a Large-Scale LISP Shared Model Virtualization task has been performed at one or more central (headquarters) sites.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 2	**vrf context***vrf-name* **Example:** `Switch(config)# vrf context vrf1`	Enters VRF configuration submode.

Step 3	**database-mapping** *EID-prefix/prefix-length locator* **priority** *priority* **weight** *weight* **Example:** `Switch(config-vrf)# 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 4	**ip lisp etr map-server** *map-server-address* **key** *key-type authentication-key* **Example:** `Switch(config-vrf)# ip lisp 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 switch, 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.
Step 5	Repeat Step 4 to configure another locator address for the same LISP map server. **Example:** `Switch(config-vrf)# ip lisp 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 switch, 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 6	**ip lisp itr map-resolver** *map-resolver-address* **Example:** `Switch(config-vrf)# ip lisp itr map-resolver 172.16.1.2`	Configures a locator address for the LISP map resolver to which this switch will send map request messages for IPv4 EID-to-RLOC mapping resolutions. • In this example, the map resolver is specified within switch 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. **Note** Up to two map resolvers may be configured if multiple map resolvers are available.
Step 7	Repeat Step 6 to configure another locator address for the LISP map resolver **Example:** `Switch(config-vrf)# ip lisp itr map-resolver 172.16.1.6`	Configures a locator address for the LISP map resolver to which this switch will send map request messages for IPv4 EID-to-RLOC mapping resolutions. **Note** 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.

	Command or Action	Purpose
		Note Up to two map resolvers may be configured if multiple map resolvers are available.
Step 8	**ip lisp itr** **Example:** `Switch(config-vrf)# ip lisp itr`	Enables LISP ITR functionality for the IPv4 address family.
Step 9	**ip lisp etr** **Example:** `Switch(config-vrf)# ip lisp etr`	Enables LISP ETR functionality for the IPv4 address family.
Step 10	**ip lisp locator-vrf default** **Example:** `Switch(config-vrf)# ip lisp locator-vrf BLUE`	Configures a non-default VRF table to be referenced by any IPv4 locators addresses.
Step 11	**ipv6 lisp locator-vrf default** **Example:** `Switch(config-vrf)# ipv6 lisp locator-vrf default`	Configures a non-default VRF table to be referenced by any IPv6 locator addresses.
Step 12	**exit** **Example:** `Switch(config-vrf)# exit`	Exits VRF configuration mode and returns to global configuration mode.
Step 13	Repeat Steps 2 to 12 for all VRFs.	

Step 14	**ip route** *ipv4-prefix next-hop*
	Example:
	`Switch(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
		• 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 • a static route to Null0 • 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 15	(Optional) **show running-config lisp** **Example:** `Switch(config)# show running-config lisp`	Verifies the LISP configuration on the switch.
Step 16	(Optional) **show** [**ip** | **ipv6**] **lisp** **Example:** `Switch(config)# show ip lisp vrf TRANS`	The **show ip lisp** and **show ipv6 lisp** commands verify the operational status of LISP as configured on the switch, as applicable to the IPv4 and IPv6 address families, respectively.
Step 17	(Optional) **show** [**ip** | **ipv6**] **lisp map-cache** [**vrf** *vrf-name*] **Example:** `Switch(config)# show ip lisp map-cache`	The **show ip lisp map-cache** and **show ipv6 lisp map-cache** commands verify the operational status of the map cache on a switch configured as an ITR or PITR, as applicable to the IPv4 and IPv6 address families, respectively.
Step 18	(Optional) **show** [**ip** | **ipv6**] **lisp database** [**vrf** *vrf-name*] **Example:** The following example shows IPv6 mapping database information for the VRF named GOLD. `Switch(config)# show ipv6 lisp database vrf GOLD`	The **show ip lisp database** and **show ipv6 lisp database** commands display the operational status of the database mapping on a switch configured as an ETR, as applicable to the IPv4 and IPv6 address families, respectively.
Step 19	(Optional) **show lisp site** [**name** *site-name*] **Example:** `Switch(config)# show lisp site`	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 switch configured as a map server.
Step 20	**clear** [**ip** | **ipv6**] **lisp map-cache** [**vrf** *vrf-name*] **Example:**	The **clear ip lisp map-cache** and **clear ipv6 lisp map-cache** commands remove all IPv4 or IPv6 dynamic LISP map-cache entries

	Command or Action	Purpose

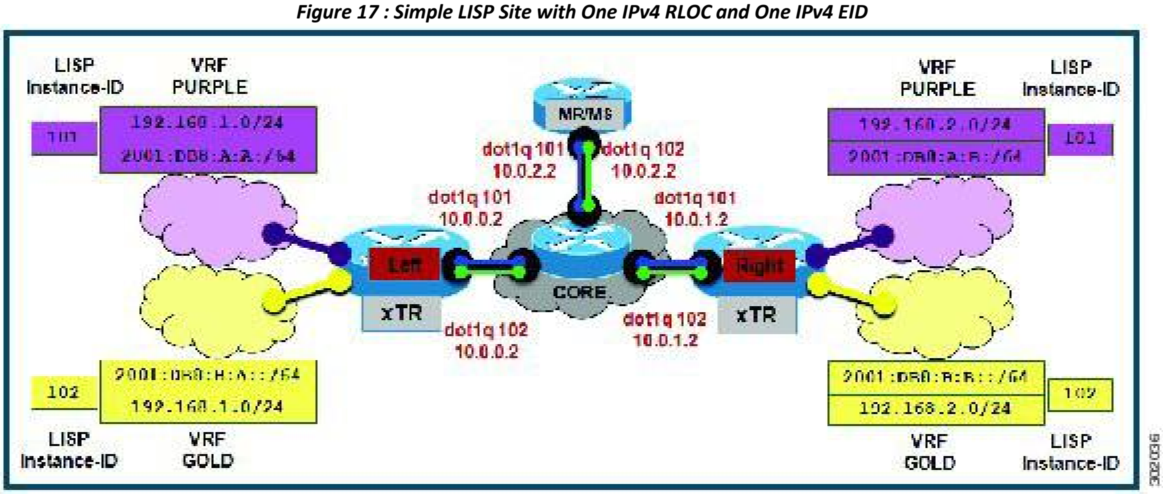
	The following commands display IPv4 mapping cache information for vrf1, and clear the mapping cache for vrf1. Clearing also displays the show information after it clears the cache.
	`Switch(config)# show ip lisp map-cache vrf vrf1`
	`Switch(config)# clear ip lisp map-cache vrf vrf1`
		stored by the switch. These verify the operational status of the LISP control plane. The command applies to a LISP switch that maintains a map cache (for example, if configured as an ITR or PITR).

3.2.5 Configuring Simple LISP Parallel Model Virtualization

You can perform these tasks to enable and configure LISP ITR/ETR (xTR) functionality and LISP map resolver and map server for LISP parallel model virtualization.

The configuration in the following figure below is for two LISP sites that are connected in parallel mode. Each LISP site uses a single edge switch configured as both an ITR and ETR (xTR), with a single connection to its upstream provider. Note that the upstream connection is VLAN-segmented to maintain RLOC space separation within the core. Two VRFs are defined here: BLUE and GREEN. The IPv4 RLOC space is used in each of these parallel networks. Both IPv4 and IPv6 EID address space is used. The LISP site registers to one map server/map resolver (MS/MR), which is segmented to maintain the parallel model architecture of the core network.

Figure 17 : Simple LISP Site with One IPv4 RLOC and One IPv4 EID



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

LISP site

• The customer premises equipment (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. The share key is configured “per-VPN."

• Each LISP xTR has a single RLOC connection to a parallel IPv4 core network.

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 (Left-xTR and Right-xTR).

Before you begin
Create the VRFs using the **vrf context** command.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**vrf context** *vrf-name* **Example:** `switch(config)# vrf context vrf1`	Enters VRF configuration submode. • In this example, the RLOC VRF named vrf1 is configured.
Step 3	**lisp instance-id** *instance-id* **Example:** `switch(config-vrf)# lisp instance-id 101`	Configures an association between a VRF and a LISP instance ID.
Step 4	**ip lisp database-mapping** *EID-prefix/prefix-length locator* **priority** *priority* **weight** *weight* **Example:** `switch(config-vrf)# ip lisp database-mapping 192.168.1.0/24 10.0.0.2 priority 1 weight 1`	Configures an EID-to-RLOC mapping relationship and its associated traffic policy for this LISP site. **Note** In this example, a single IPv4 EID prefix, 192.168.1.0/24, within instance ID 1 at this site is associated with the local IPv4 RLOC 10.0.0.2.
Step 5	**exit** **Example:** `switch(config-vrf)# exit`	Exits VRF configuration submode and returns to global mode.
Step 6	**ipv4 itr map-resolver** *map-resolver-address* **Example:** `switch(config)# ip lisp itrmap-resolver 10.0.2.2`	Configures a locator address for the LISP map resolver to which this switch will send map request messages for IPv4 EID-to-RLOC mapping resolutions.

	Command or Action	Purpose
		Note 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. **Note** Up to two map resolvers may be configured if multiple map resolvers are available.
Step 7	**ip lisp etr map-server** *map-server-address* **key** *key-type authentication-key* **Example:** `switch(config)# ip lisp etr map-server 10.0.2.2 key 0 PURPLE-key`	Configures a locator address for the LISP map server and an authentication key for which this switch, acting as an IPv4 LISP ETR, will use to register with the LISP mapping system. **Note** 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.
Step 8	**ip lisp itr** **Example:** `switch(config)# ip lisp itr`	Enables LISP ITR functionality for the IPv4 address family.
Step 9	**ip lisp etr** **Example:** `switch(config)# ip lisp etr`	Enables LISP ETR functionality for the IPv4 address family.
Step 10	**ipv6 lisp itr map-resolver** *map-resolver-address* **Example:** `switch(config)# ipv6 lisp itr map-resolver 10.0.2.2`	Configures a locator address for the LISP map resolver to which this switch will send map request messages for IPv6 EID-to-RLOC mapping resolutions.

	Command or Action	Purpose
		Note 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 addresses. **Note** Up to two map resolvers may be configured if multiple map resolvers are available.
Step 11	**ipv6 lisp etr map-server** *map-server-address* **key** *key-type authentication-key* **Example:** `switch(config)# ipv6 lisp etr map-server 10.0.2.2 key 0 PURPLE-key`	Configures a locator address for the LISP map-server and an authentication key that this switch, acting as an IPv6 LISP ETR, will use to register to the LISP mapping system. **Note** 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.
Step 12	**ipv6 itr** **Example:** `switch(config)# ipv6 itr`	Enables LISP ITR functionality for the IPv6 address family.
Step 13	**ipv6 etr** **Example:** `switch(config)# ipv6 etr`	Enables LISP ETR functionality for the IPv6 address family.

<table>
<tr><td>Step 14</td><td>ip route vrf rloc-vrf-name ipv4-prefix next-hop
Example:
<code>switch(config)# ip route vrf BLUE 0.0.0.0 0.0.0.0 10.0.0.1</code></td><td>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
a static route to Null0
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.</td></tr>
<tr><td>Step 15</td><td>ipv6 route vrf rloc-vrf-name ipv6-prefix next-hop
Example:
<code>switch(config)# ipv6 route vrf BLUE ::/0 Null0</code></td><td>Configures a default route to the upstream next hop for all IPv6 destinations, reachable within the specified RLOC VRF.

All IPv6 EID-sourced packets destined for both LISP and non-LISP sites require LISP support for forwarding in the following 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
a static route to Null0
no route at all</td></tr>
</table>

		In this configuration example, because the xTR has only IPv4 RLOC connectivity, adding an IPv6 default route to Null0 ensures that all IPv6 packets are handled by LISP processing. If the destination is another LISP site, packets are LISP-encapsulated (using IPv4 RLOCs) to the remote site. If the destination is non-LISP, all IPv6 EIDs are LISP-encapsulated to a Proxy ETR (PETR) –assuming one is configured.**Note** The use of the static route to Null0 is not required, but is considered a LISP best practice.
Step 16	(Optional) **show running-config lisp** **Example:** `switch(config)# show running-config lisp`	Shows the LISP configuration on the switch.
Step 17	(Optional) **show** [**ip** | **ipv6**] **lisp** **Example:** `switch(config)# show ip lisp vrf TRANS`	The **show ip lisp** and **show ipv6 lisp** commands verify the operational status of LISP as configured on the switch, as applicable to the IPv4 and IPv6 address families, respectively.
Step 18	(Optional) **show** [**ip** | **ipv6**] **lisp map-cache** [**vrf** *vrf-name*] **Example:** `switch(config)# show ip lisp map-cache`	The **show ip lisp map-cache** and **show ipv6 lisp map-cache** commands verify the operational status of the map cache on a switch configured as an ITR or Proxy ETR (PETR), as applicable to the IPv4 and IPv6 address families, respectively.
Step 19	(Optional) **show** [**ip** | **ipv6**] **lisp database** [**vrf** *vrf-name*] **Example:** The following example shows IPv6 mapping database information for the VRF named GOLD. `switch(config)# show ipv6 lisp database vrf GOLD`	The **show ip lisp database** and **show ipv6 lisp database** commands verify the operational status of the database mapping on a switch configured as an ETR, as applicable to the IPv4 and IPv6 address families, respectively.
Step 20	(Optional) **show lisp site** [**name** *site-name*] **Example:** `switch(config)# show lisp site`	The **show lisp site** command verifies the operational status of LISP sites, as configured on a map server. This command only applies to a switch configured as a map server.

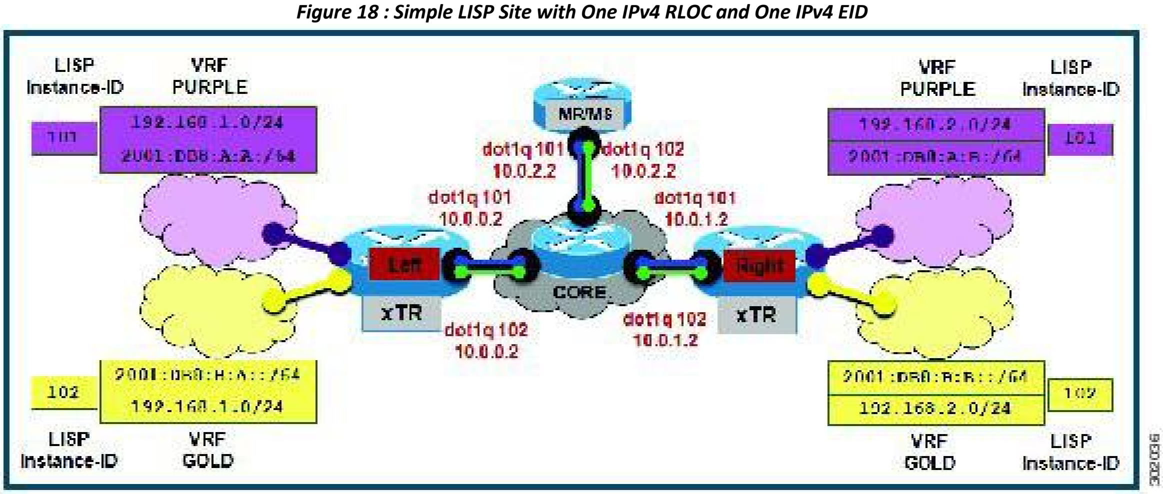
Step 21	**clear** [**ip** | **ipv6**] **lisp map-cache** [**vrf** *vrf-name*]
	Example:
	`switch(config)# show ip lisp map-cache vrf vrf1`
	`switch(config)# clear ip lisp map-cache vrf vrf1`
		The **clear ip lisp map-cache** and **clear ipv6 lisp map-cache** commands remove all IPv4 or IPv6 dynamic LISP map-cache entries stored by the switch. This verifies the operational status of the LISP control plane. This command applies to a LISP switch that maintains a map cache (for example, if configured as an ITR or PITR).
		The commands in the example display IPv4 mapping cache information for vrf1, and clear the mapping cache for vrf1 and show information after clearing the cache.

3.2.6 Configuring a Private LISP Mapping System for LISP Parallel Model Virtualization

Perform this task to configure and enable standalone LISP map server/map resolver functionality for LISP parallel model virtualization. In this task, a Inspur switch is configured as a standalone map resolver/map server (MR/MS) for a private LISP mapping system. Because the MR/MS is configured as a stand-alone switch, 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.

- **Mapping system:**

Figure 18 : Simple LISP Site with One IPv4 RLOC and One IPv4 EID



- One map resolver/map server (MS/MR) system is shown in the figure above and assumed available for the LISP xTR to register to within the proper parallel RLOC space. The MS/MR has an IPv4 RLOC address of 10.0.2.2, within each VLAN/VRF (Green and Blue) providing parallel model RLOX separation in the IPv4 core.
- The map server site configurations are virtualized using LISP instance IDs to maintain separation between the two VRFs, PURPLE and GOLD.

Repeat this task for all lisp instantiations and RLOC VRFs.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 2	**lisp site** *site-name* **Example:** `Switch(config)# lisp site PURPLE`	Specifies a LISP site named Purple and enters LISP site configuration mode. • In this example, the LISP site named Purple is configured.
Step 3	**authentication-key** [*key-type*] *authentication-key* **Example:** `Switch(config-lisp-site)# authentication-key 0 Purple-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 ETR must be configured with EID prefixes and instance IDs matching the one(s) configured on this map server, as well as an identical authentication key.
Step 4	**eid-prefix** *EID-prefix* **instance-id** *instance-id* **Example:** `Switch(config-lisp-site)# eid-prefix 192.168.1.0/24 instance-id 101`	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 IPv4 EID prefixes under this LISP site. • In this example, the IPv4 EID prefix 192.168.1.0/24 and instance ID 101 are associated together.
Step 5	**eid-prefix** *EID-prefix* **instance-id** *instance-id* **Example:** `Switch(config-lisp-site)# eid-prefix 2001:db8:a:b::/64 instance-id 101`	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 IPv6 EID prefixes under this LISP site. • In this example, the IPv6 EID prefix 2001:db8:a:a::/64 and instance ID 101 are associated together.

Step 6	**exit** **Example:** `Switch(config-lisp-site)# exit`	Exits LISP site configuration mode and returns to global configuration mode.
Step 7	**ip lisp map-resolver** **Example:** `Switch(config)# ip lisp map-resolver`	Enables LISP map resolver functionality for EIDs in the IPv4 address family.
Step 8	**ip lisp map-server** **Example:** `Switch(config)# ip lisp map-server`	Enables LISP map server functionality for EIDs in the IPv4 address family.
Step 9	**ipv6 lisp map-resolver** **Example:** `Switch(config)# ipv6 lisp map-resolver`	Enables LISP map resolver functionality for EIDs in the IPv6 address family.
Step 10	**ipv6 lisp map-server** **Example:** `Switch(config)# ipv6 lisp map-server`	Enables LISP map server functionality for EIDs in the IPv6 address family.
Step 11	**ip route vrf** *rloc-vrf-name ipv4-prefix next-hop* **Example:** `Switch(config)# ip route vrf BLUE 0.0.0.0 0.0.0.0 10.0.2.1`	Configures a default route to the upstream next hop for all IPv4 destinations, reachable within the specified RLOC VRF.
Step 12	**show running-config lisp** **Example:** `Switch(config)# show running-config lisp`	Verifies the LISP configuration on the switch.
Step 13	**show** [**ip** | **ipv6**] **lisp** **Example:** `Switch(config)# show ip lisp vrf TRANS`	The **show ip lisp** and **show ipv6 lisp** commands are useful for quickly verifying the operational status of LISP as configured on the switch, as applicable to the IPv4 and IPv6 address families respectively.

Step 14	**show** [**ip** | **ipv6**] **lisp map-cache** [**vrf** *vrf-name*] **Example:** `Switch(config)# show ip 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 switch configured as an ITR or PITR, as applicable to the IPv4 and IPv6 address families respectively.
Step 15	**show** [**ip** | **ipv6**] **lisp database** [**vrf** *vrf-name*] **Example:** The following example shows IPv6 mapping database information for the VRF named GOLD. `Switch(config)# show ipv6 lisp database vrf GOLD`	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 switch configured as an ETR, as applicable to the IPv4 and IPv6 address families respectively.
Step 16	**show lisp site** [**name** *site-name*] **Example:** `Switch(config)# show lisp site`	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 switch configured as a map server.

Step 17	**clear** [**ip** | **ipv6**] **lisp map-cache** [**vrf** *vrf-name*]
	Example:
	The following example displays IPv4 mapping cache information for vrf1, shows the command used to clear the mapping cache for vrf1, and displays the show information after clearing the cache.
	`Switch(config)# show ip lisp map-cache vrf vrf1`
	`Switch(config)# clear ip lisp map-cache vrf vrf1`
		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 switch. This can be useful for trying to quickly verify the operational status of the LISP control plane. This command applies to a LISP switch that maintains a map cache (for example, if configured as an ITR or PITR).

3.3 Configuration Examples for LISP Instance-ID Support

3.3.1 Example: Configuring Simple LISP Shared Model Virtualization

These examples show the complete configuration for the LISP topology. On the xTRs, the VRFs and EID prefixes are assumed to be attached to VLANs configured on the switches.

This example shows how to configure the left xTR:

```
vrf context GOLD
 ipv6 lisp itr
 ip lisp itr
 ipv6 lisp etr
 ip lisp etr
ipv6 lisp
 database-
 mapping
 2001:db8:b:a::/
 64 10.0.0.2
```

```
 priority 1
 weight 100
  ip lisp database-mapping 192.168.1.0/24 10.0.0.2 priority 1 weight 100
  lisp instance-id 102
  ipv6 lisp locator-
  vrf default ip
  lisp locator-vrf
  default
  ipv6 lisp itr map-
  resolver 10.0.2.2 ip
  lisp itr map-resolver
  10.0.2.2
  ipv6 lisp etr map-server 10.0.2.2 key Left-key


  ip lisp etr map-server 10.0.2.2 key

Left-key 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

vrf context
 PURPLE ipv6
 lisp itr
ip lisp itr ipv6
 lisp etr ip
 lisp etr
  ipv6 lisp database-mapping 2001:db8:a:a::/64 10.0.0.2 priority 1 weight 100
  ip lisp database-mapping 192.168.1.0/24 10.0.0.2 priority 1 weight 100
  lisp instance-id 101
  ipv6 lisp locator-vrf
  default ip lisp
  locator-vrf default
  ipv6 lisp itr map-resolver
  10.0.2.2 ip lisp itr map-
  resolver 10.0.2.2
  ipv6 lisp etr map-server 10.0.2.2 key
  Left-key ip lisp etr map-server
  10.0.2.2 key Left-key
```

This example shows how to configure the right xTR:

```
vrf context GOLD ipv6 lisp itr ip lisp
itr ipv6 lisp etr ip lisp etr
ipv6 lisp database-mapping 2001:db8:b:b::/64 10.0.1.2 priority 1 weight 100
ip lisp database-mapping 192.168.2.0/24 10.0.1.2 priority 1 weight 100
lisp instance-id 102
```

```
  ipv6 lisp locator-vrf
  default ip lisp
  locator-vrf default
  ipv6 lisp itr map-resolver
  10.0.2.2 ip lisp itr map-
  resolver 10.0.2.2
  ipv6 lisp etr map-server 10.0.2.2 key
  Right-key ip lisp etr map-server
  10.0.2.2 key Right-key

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


 encapsulatio
 n dot1q 102
 vrf
 forwarding
 GOLD
 ip address 192.168.2.1 255.255.255.0
 ipv6 address 2001:DB8:B:B::1/64

 vrf context
 PURPLE ipv6
 lisp itr
 ip lisp itr
 ipv6 lisp
 etr ip lisp
 etr
  ipv6 lisp database-mapping 2001:db8:a:b::/64 10.0.1.2 priority 1 weight 100
  ip lisp database-mapping 192.168.2.0/24 10.0.1.2 priority 1 weight 100
  lisp instance-id 101
  ipv6 lisp locator-
  vrf default ip
  lisp locator-vrf
  default
  ipv6 lisp itr map-
  resolver 10.0.2.2 ip
  lisp itr map-resolver
  10.0.2.2
  ipv6 lisp etr map-server 10.0.2.2
  key Right-key ip lisp etr map-server
  10.0.2.2 key Right-key
```

3.3.2 Example: Configuring a Private LISP Mapping System for LISP Shared Model Virtualization

This example shows how to configure the LISP 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-
    resolv
    er
    ipv6
    map-
    server
    ipv6
    map-
    resolv
    er
    exit
!
    ip route 0.0.0.0 0.0.0.0 10.0.2.1
```

3.3.3 Example: Configuring Large-Scale LISP Shared Model Virtualization

Example:

The examples 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 switches. Both HQ-RTR-1 and HQ-RTR-2 are provided to illustrate the proper method for configuring a LISP multihomed site.

This example shows how to configure HQ-RTR-1 with an xTR, a map server, and a map resolver.

```
feature lisp
interface
loopback 0
   ip address
172.31.1.11/32
interface
ethernet2/1
   ip address
172.16.1.6/30
```

```
interface Ethernet
2/2
   vrf member TRANS
   ip address
10.1.1.1/24
interface
Ethernet 2/3
    vrf member SOC
    ip address
10.2.1.1/24
interface Ethernet
2/4
    vrf member FIN
    ip address
10.3.1.1/24 ip
lisp itr
ip lisp etr
ip lisp map-
resolver ip
lisp map-
server
ip lisp database-mapping 172.31.1.11/32 172.16.1.2 priority 1 weight 50
ip lisp database-mapping 172.31.1.11/32 172.16.1.6 priority 1
weight 50 ip lisp itr map-resolver 172.16.1.2
ip lisp itr map-resolver 172.16.1.6
ip lisp etr map-server 172.16.1.2 key DEFAULT-key ip lisp etr map-server
172.16.1.6 key DEFAULT-key vrf context FIN
ip lisp itr ip lisp etr
  ip lisp database-mapping 10.3.1.0/24 172.16.1.2 priority 1 weight 50
  ip lisp database-mapping 10.3.1.0/24 172.16.1.6 priority 1 weight 50
  lisp instance-id 3
  ip lisp itr map-resolver
  172.16.1.2 ip lisp itr map-
  resolver 172.16.1.6
  ip lisp etr map-server 172.16.1.2 key
  FIN-key   ip   lisp   etr   map-server
  172.16.1.6   key   FIN-key   ip   lisp
  locator-vrf default
  vrf context SOC ip lisp itr ip lisp
  etr
  ip lisp database-mapping 10.2.1.0/24 172.16.1.2 priority 1 weight 50
  ip lisp database-mapping 10.2.1.0/24 172.16.1.6 priority 1 weight 50
  lisp instance-id 2
  ip lisp itr map-resolver
  172.16.1.2 ip lisp itr map-
  resolver 172.16.1.6
  ip lisp etr map-server 172.16.1.2 key
  SOC-key   ip   lisp   etr   map-server
  172.16.1.6   key   SOC-key   ip   lisp
  locator-vrf default
vrf
  context
  TRANS
  ip lisp
  itr
  ip lisp etr
  ip lisp database-mapping 10.1.1.0/24 172.16.1.2 priority 1 weight 50
  ip lisp database-mapping 10.1.1.0/24 172.16.1.6 priority 1 weight 50

  lisp instance-id 1
  ip lisp itr map-resolver
  172.16.1.2 ip lisp itr
  map-resolver 172.16.1.6
  ip  lisp  etr  map-server  172.16.1.2
  key  TRANS-key  ip  lisp  etr  map-
```

```
  server  172.16.1.6  key  TRANS-key  ip
  lisp locator-vrf default
lisp site DEFAULT
  eid-prefix 172.31.1.0/24 accept-
  more-specifics authentication-key
  DEFAULT-key
lisp site FIN
  eid-prefix 10.3.0.0/16 accept-
  more-specifics authentication-
  key FIN-key
lisp site SOC
  eid-prefix 10.2.0.0/16 instance-id 2 accept-
  more-specifics authentication-key SOC-key
lisp site TRANS
  eid-prefix 10.1.0.0/16 instance-id 1 accept-
  more-specifics authentication-key TRANS-key
```

This example shows how to configure HQ-RTR-2 with an xTR, a map server, and a map resolver.

```
feature
lisp
interface
loopback
0
   ip address
172.31.1.12/32
interface
ethernet2/1
   ip address
172.16.1.6/30
interface
Ethernet 2/2
   vrf member TRANS
   ip address
10.1.1.2/24
interface
Ethernet 2/3
    vrf member SOC
    ip address
10.2.1.2/24
interface
Ethernet 2/4
    vrf member FIN
    ip address 10.3.1.2/24

    ip lisp itr ip lisp etr
    ip lisp map-resolver ip lisp map-server
ip lisp database-mapping 172.31.1.12/32 172.16.1.2 priority 1 weight 50
ip lisp database-mapping 172.31.1.12/32 172.16.1.6 priority
1 weight 50 ip lisp itr map-resolver 172.16.1.2
ip lisp itr map-resolver 172.16.1.6
ip lisp etr map-server 172.16.1.2 key
DEFAULT-key  ip  lisp  etr  map-server
172.16.1.6    key    DEFAULT-key    vrf
context FIN
ip lisp itr ip lisp etr
  ip lisp database-mapping 10.3.1.0/24 172.16.1.2 priority 1 weight 50
  ip lisp database-mapping 10.3.1.0/24 172.16.1.6 priority 1 weight 50
  lisp instance-id 3
  ip lisp itr map-resolver
  172.16.1.2 ip lisp itr
  map-resolver 172.16.1.6
  ip lisp etr map-server 172.16.1.2
  key  FIN-key  ip  lisp  etr  map-
```

```
  server 172.16.1.6 key FIN-key ip
  lisp locator-vrf default
  vrf context SOC ip lisp itr ip
  lisp etr
  ip lisp database-mapping 10.2.1.0/24 172.16.1.2 priority 1 weight 50
  ip lisp database-mapping 10.2.1.0/24 172.16.1.6 priority 1 weight 50
  lisp instance-id 2



  ip lisp itr map-resolver
  172.16.1.2 ip lisp itr map-
  resolver 172.16.1.6
  ip lisp etr map-server 172.16.1.2 key
  SOC-key ip lisp etr map-server
  172.16.1.6 key SOC-key ip lisp
  locator-vrf default
vrf
  context
  TRANS
  ip lisp
  itr
  ip lisp etr
  ip lisp database-mapping 10.1.1.0/24 172.16.1.2 priority 1 weight 50
  ip lisp database-mapping 10.1.1.0/24 172.16.1.6 priority 1 weight 50
  lisp instance-id 1
  ip lisp itr map-resolver
  172.16.1.2 ip lisp itr map-
  resolver 172.16.1.6
  ip lisp etr map-server 172.16.1.2 key
  TRANS-key ip lisp etr map-server
  172.16.1.6 key TRANS-key ip lisp
  locator-vrf default
lisp site DEFAULT
  eid-prefix 172.31.1.0/24 accept-more-
  specifics authentication-key DEFAULT-
  key
lisp site FIN
  eid-prefix 10.3.0.0/16 accept-more-
  specifics authentication-key FIN-key
lisp site SOC
  eid-prefix 10.2.0.0/16 instance-id 2 accept-more-
  specifics authentication-key SOC-key
lisp site TRANS
  eid-prefix 10.1.0.0/16 instance-id 1 accept-more-
  specifics authentication-key TRANS-key
```

3.3.4 Example: Configuring a Remote Site for Large-Scale LISP Shared Model Virtualization

This example shows the complete configuration for the remote site switch. Only one remote site configuration is shown.

This example shows how to configure Site 2 with an xTR, using the map server and a map resolver from the HQ site.

```
feature lisp
interface
loopback 0
   ip address
172.31.1.2/32
interface
ethernet2/1
   ip address
```

```
172.16.2.2/30
interface Ethernet
2/2
   vrf member TRANS
   ip address
10.1.2.1/24
interface
Ethernet 2/3
    vrf member SOC
    ip address
10.2.2.1/24
interface Ethernet
2/4
    vrf member FIN
    ip address
    10.3.2.1/24

    ip lisp itr ip
    lisp etr
ip lisp map-
resolver ip
lisp map-
server
ip lisp database-mapping 172.31.1.2/32 172.16.2.2 priority 1 weight 100

ip lisp itr map-resolver
172.16.1.2 ip lisp itr
map-resolver 172.16.1.6
ip lisp etr map-server
172.16.1.2 key DEFAULT-
key ip lisp etr map-
server 172.16.1.6 key
DEFAULT-key vrf context
FIN
ip lisp itr ip lisp etr
  ip lisp database-mapping 10.3.2.0/24 172.16.2.2 priority 1 weight 100
lisp instance-id 3
  ip lisp itr map-resolver
  172.16.1.2 ip lisp itr
  map-resolver 172.16.1.6
  ip lisp etr map-server 172.16.1.2
  key  FIN-key  ip  lisp  etr  map-
  server  172.16.1.6  key  FIN-key  ip
  lisp locator-vrf default
vrf context SOC ip lisp
itr ip lisp etr
  ip lisp database-mapping 10.2.2.0/24 172.16.2.2 priority 1 weight 100
  lisp instance-id 2
  ip lisp itr map-resolver
  172.16.1.2 ip lisp itr
  map-resolver 172.16.1.6
  ip lisp etr map-server 172.16.1.2
  key  SOC-key  ip  lisp  etr  map-
  server  172.16.1.6  key  SOC-key  ip
  lisp locator-vrf default
  vrf context TRANS ip lisp itr
  ip lisp etr
  ip lisp database-mapping 10.1.2.0/24 172.16.2.2 priority 1 weight 100
  lisp instance-id 1
  ip lisp itr map-resolver
  172.16.1.2 ip lisp itr
  map-resolver 172.16.1.6
  ip  lisp  etr  map-server  172.16.1.2
  key  TRANS-key  ip  lisp  etr  map-
  server  172.16.1.6  key  TRANS-key  ip
```

```
lisp locator-vrf default
```

3.3.5 Example: Configuring Simple LISP Parallel Model Virtualization

Example:

These examples show the complete configuration for the LISP topology. On the xTRs, the VRFs and EID prefixes are assumed to be attached to VLANs configured on the switches.

This example shows how to configure 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
ipv6 route ::/0 Null0
```

This example shows how to configure the 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
encapsulati
on     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
ipv6 route ::/0 Null0
```

3.3.6 Example: Configuring a Private LISP Mapping System for LISP Parallel Model Virtualization

This example shows how to configure the map server/map resolver:

```
hostname MSMR
!
vrf definition BLUE address-family ipv4 exit
!
vrf definition GREEN address-family ipv4 exit
!
ipv6 unicast-routing
!
interface Ethernet0/0.101 encapsulation dot1Q 101 vrf
forwarding BLUE
ip address 10.0.0.2 255.255.255.0
!
interface
Ethernet0/0.10
2
encapsulation
dot1Q 102 vrf
forwarding
GREEN
ip address 10.0.0.2 255.255.255.0

!
router lisp 1
locator-table
vrf BLUE site
Purple
authentication-key
PURPLE-key
eid-prefix instance-id 101
192.168.1.0/24 eid-prefix instance-
id 101 192.168.2.0/24 eid-prefix
instance-id 101 2001:DB8:A:A::/64
eid-prefix instance-id 101
2001:DB8:A:B::/64
!
ipv4 map-server ipv4 map-resolver
ipv6 map-server ipv6 map-resolver
!
router lisp 2
locator-table
vrf GREEN site
Gold
authentication-key
GOLD-key
eid-prefix instance-id 102
```

```
192.168.1.0/24 eid-prefix instance-
id 102 192.168.2.0/24 eid-prefix
instance-id 102 2001:DB8:B:A::/64
eid-prefix instance-id 102
2001:DB8:B:B::/64
!
ipv4 map-server ipv4 map-resolver
ipv6 map-server ipv6 map-resolver
!
ip route vrf GREEN 0.0.0.0 0.0.0.0 10.0.2.1
ip route vrf BLUE 0.0.0.0 0.0.0.0 10.0.2.1
```

3.4 Feature History for Configuring LISP Instance ID

This table lists the release history for this feature.

Table 4 : Feature History for Configuring LISP Instance ID

Feature Name	Releases	Feature Information
Locator/ID Separation Protocol (LISP) Instance ID	8.3(1)	This feature is introduced.

CHAPTER 4 Configuring LISP Delegate Database Tree (DDT)

This chapter contains the following sections:
·LISP Delegate Database Tree (DDT).
·Overview of DDT.
·Restrictions for LISP Delegate Database Tree (DDT).
·Configuring LISP Delegate Database Tree (DDT).
·Configuration Examples for LISP Delegate Database Tree (DDT).

4.1 LISP Delegate Database Tree (DDT)

4.2 Overview of DDT

LISP Delegated Database Tree (DDT) defines a large-scale distributed database of LISP Endpoint Identifier (EID) space using a DDT node. A DDT node is configured to be authoritative for some specified portion of an overall LISP EID space, as well as the set of more specific subprefixes that are delegated to other DDT nodes. It is also configured with the set of more-specific sub-prefixes that are further delegated to other DDT nodes. To delegate a sub-prefix, the "parent" DDT node is configured with the Routing Locators (RLOCs) of each child DDT node that is authoritative for the sub-prefix. Each RLOC either points to a map server (sometimes termed a "terminal DDT node") to which an egress tunnel routers (ETRs) registers that sub-prefix or points to another.

4.3 Restrictions for LISP Delegate Database Tree (DDT)

The following restriction applies to the LISP Delegate Database Tree (DDT) feature:
·If LISP is enabled, nondisruptive upgrade (ISSU) and nondisruptive downgrade (ISSD) paths are not supported. Disable LISP prior to any upgrade. This restriction only applies to releases before 8.3(1) but not to this release or to future LISP releases.

4.4 Configuring LISP Delegate Database Tree (DDT)

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 2	**lisp ddt** **Example:** `Switch(config)# lisp ddt`	Configures a switch to perform LISP DDT functionality.

Step 3	**lisp ddt root** *root-locator* [**public-key** *number*] **Example:** `Switch(config)# lisp ddt root 10.1.1.1`	Configures an IPv4 or IPv6 locator for a DDT root node within the delegation hierarchy on a DDT-enabled map resolver. • In this example, a DDT-enabled map resolver is configured to refer to the DDT root node locator: 2001:db8:1::1111.
Step 4	**lisp ddt map-server-peer***map-server-locator* {**eid-prefix** *eid-prefix* | **instance-id** *iid*} **[map-server]** *map-server-locator* **Example:** `Switch(config)# lisp ddt map-server-peer 10.1.1.1 eid-prefix 2001:db8:eeee::/48`	Configures a DDT-enabled map server, the locator and EID prefix (and/or instance ID) for a map server peer within the LISP DDT delegation hierarchy. • In this example, a LISP DDT map server is configured as authoritative for the IPv6 EID prefix 2001:db8:eeee::/48 for its own locator 10.1.1.1
Step 5	**lisp ddt authoritative-prefix** {**eid-prefix** *eid-prefix* | **instance-id** *iid* } **Example:** `Switch(config)# lisp ddt authoritative-prefix eid-prefix 172.16.0.0/16`	Configures a LISP DDT node to be authoritative for a specified EID prefix. • In this example, the LISP DDT node is configured to be authoritative for the IPv4 EID-prefix 172.16.0.0/16
Step 6	**exit** **Example:** `Switch(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 7	**show lisp ddt vrf** *vrf-name* **Example:** `Switch# show lisp ddt vrf vrf-1`	Displays the configured DDT root(s) and/or DDT delegation nodes on a switch enabled for LISP DDT. When **vrf** *vrf-name*is specified, information for VRF is displayed.
Step 8	**show lisp ddt queue** [*eid-address* | **instance-id** *iid* {*eid-address*} | **vrf** *vrf-name*] **Example:**	Displays the map-resolver's map-request queue. If *eid-address* is specified, then only the queue element for an EID being map-requested is displayed
Step 9	**show lisp ddt referral-cache** [*eid-address* | **instance-id** *iid* {*eid-address*} | **cache-entries** {**vrf** *vrf-name*} | **vrf** *vrf-name*] **Example:** `Switch# show lisp ddt referral-cache 10.1.1.1`	Displays the DDT referral cache stored in map-resolvers. When the *eid-address* variable is specified each cache entry that is less specific than the *eid-address* variable will be displayed.
Step 10	**end** **Example:** `Switch# end`	

4.5 Configuration Examples for LISP Delegate Database Tree (DDT)

4.5.1 Examples: LISP Delegate Database Tree (DDT)

The following is an example of parent and child DDT nodes, where the parent has all of 10.0.0.0/8 and delegates two sub-prefixes, 10.0.0.0/12 and 10.0.16.0/12 to two child DDT nodes. All of these prefixes are within the DDT sub-tree Key-ID=0, IID=223, and AFI=1 (IPv4).

```
Switch(config)# lisp ddt authoritative-prefix instance-id 223 10.0.0.0/8
Switch(config)# lisp ddt child 192.168.1.100 instance-id 223 eid-prefix
10.0.0.0/12 Switch(config)# lisp ddt child 192.168.1.200 instance-id 223
eid-prefix 10.16.0.0/12
```

The following example defines the delegation of the EID-prefix 10.0.0.0/12 to a DDT Map Server with RLOC 192.168.1.100 and delegation of the EID-prefix 10.16.0.0/12 to a DDT Map-Server with RLOC 192.168.1.200. The child DDT Map-Server for 10.16.0.0/12 is further configured to allow ETRs to register the sub-prefixes 10.18.0.0/16 and 10.17.0.0/16:

```
Switch(config)# lisp ddt authoritative-prefix instance-id 223 eid-prefix
10.16.0.0/12 Switch(config)# lisp site site-1
Switch(config)#  eid-prefix 10.18.0.0/16 instance-id 223
Switch(config)# lisp site site-2
Switch(config)# eid-prefix 10.17.0.0/16 instance-id 223
```

4.6 Feature History for Delegate Database Tree

Table 5 : Feature History for LISP Delegate Database Tree

Feature Name	Releases	Feature Information
Locator/ID Separation Protocol (LISP) Delegate Database Tree (DDT)	8.3(1)	This feature is introduced.

CHAPTER 5 Configuring LISP Multicast

This chapter contains the following sections:

• LISP Multicast.
• Finding Feature Information.
• Restrictions for LISP Multicast.
• Configuration Example for LISP Multicast.

5.1 LISP Multicast

This chapter describes how to configure the Multicast functionality in Locator/ID Separation Protocol (LISP) architecture where the Multicast source and Multicast receivers can reside in separate LISP sites.

LISP introduced a mapping function from a site's Endpoint ID (EID) prefix to its associated Routing Locator (RLOC). Unicast packets require the mapping of both the source and destination address. Multicast only requires the source address to be mapped as the destination group address is not topology-dependent.

The implementation of Multicast LISP includes the following features:

·Building the multicast distribution tree across LISP sites.

·Forwarding multicast data packets from sources to receivers across LISP sites.

·Supporting different service models, including ASM (Any Source Multicast), and SSM (Source Specific Multicast).

·Supporting different combinations of LISP and non-LISP capable source and receiver sites.

·When the Multicast LISP feature is enabled, a new tunnel interface type called GLT (Generic Lisp Tunnel) is created. The GLT is supported by Oracle Identity Manager APIs and only one GLT per Virtual Device Context (VDC) is created.

Attention

The LISP Multicast feature is not supported on the F3 series module.

5.2 Finding Feature Information

Your software release may not support all the features documented in this module. 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.

5.3 Restrictions for LISP Multicast

The following restrictions apply to the LISP Multicast feature:

·Only IPv4 Multicast LISP is supported over the Unicast core.

·Only Any Source Multicast (ASM) and Single Source Multicast (SSM) modes are supported.

·Only static Rendezvous Point (RP) is supported.

5.3.1 Configuring LISP Multicast

Perform this task to configure a device to support Locator/ID Separation Protocol (LISP) Multicast functionality.

In this task, a LISP site an edge router configured as an xTR (performs as both an ITR and an ETR) and includes a single IPv4 connection to an upstream provider. Both the RLOC and the EID are IPv4. Additionally, this LISP site registers to one map resolver/map server (MR/MS) device in the network core.

Mapping system:

·One map resolver/map server (MR/MS) system is assumed to be available for the LISP xTR to configure. The MR/MS have IPv4 RLOC 11.0.0.2.

·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.

The steps in this task enable and configure LISP Multicast ITR and ETR (xTR) functionality when using a LISP map server and map resolver for mapping services.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**vrf context** *name* **Example:** `Device(config)# vrf context management`	Creates a virtual routing and forwarding instance (VRF) and enters VRF configuration mode.
Step 3	**ip pim rp-address** *rp-address access-list* **Example:** `Device(config-vrf)# ip pim rp-address 10.0.0.1 group-list 224.0.0.0/8`	Configures the address of a Protocol Independent Multicast (PIM) rendezvous point (RP) for a particular group.
Step 4	**ip pim ssm range** *access-list* **Example:** `Device(config-vrf)# ip pim ssm range 232.0.0.0/8`	Defines the Source Specific Multicast (SSM) range of IP multicast addresses.
Step 5	**ip lisp itr-etr** **Example:** `Device(config-vrf)# ip lisp itr-etr`	Configures the Inspur INOS device to act as both an IPv4 LISP Ingress Tunnel Router (ITR) and Egress Tunnel Router (ETR).

Step 6	**ip lisp database-mapping** *EID-prefix/prefix-length locator* **priority** *priority* **weight** *weight* **Example:** `Device(config-vrf)# ip lisp database-mapping 10.0.0.0/24 10.0.0.1 priority 1 weight 100`	Configures an IPv4 endpoint identifier to Routing Locator (EID-to-RLOC) mapping relationship and its associated traffic policy.
Step 7	**lisp instance-id** *id* **Example:** `Device(config-vrf)# lisp instance-id 1`	Configures an instance ID to be associated with endpoint identifier (EID)-prefixes for a Locator/ID Separation Protocol (LISP) xTR .
Step 8	**ip lisp locator-vrf default** **Example:** `Device(config-vrf)# ip lisp locator-vrf default`	Configures a nondefault virtual routing and forwarding (VRF) table to be referenced by any IPv4 locators.
Step 9	**ip lisp itr map-resolver** *map-resolver-address* **Example:** `Device(config-vrf)# ip lisp itr map-resolver 10.0.0.2`	Configures the IPv4 locator address of the Locator/ID Separation Protocol (LISP) Map-Resolver to be used by the ingress tunnel router (ITR) ITR or Proxy ITR (PITR) when sending Map-Requests for IPv4 EID-to-RLOC mapping resolution. **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	**ip lisp etr map-server** *map-server-address* **key** *key-type authentication-key* **Example:** `Device(config-vrf)# ip lisp etr map-server 10.0.0.2 key 3 5b0f2bd760fe4ce3`	Configures the IPv4 locator address of the Locator/ID Separation Protocol (LISP) Map-Server to be used by the egress tunnel router (ETR) when registering for IPv4 EIDs. **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	**ip lisp multicast** **Example:** `Device(config-vrf)# ip lisp multicast`	Configures the device to support Locator/ID Separation Protocol (LISP) Multicast functionality.

Step 12	**exit** **Example:** `Device(config-vrf)# exit`	Exits vrf configuration mode.
Step 13	**show ipmroutedetail** **Example:** `Device# show ip mroute detail`	(Optional) Displays information about the LISP multicast encapsulation for the IPv4 multicast routes.
Step 14	**show ippimlisp encap** **Example:** `Router# show ip pim lisp encap`	(Optional) Displays information about the LISP encapsulation indices stored by PIM.
Step 15	**show forwardingdistributionmulticast route** *group-addr* **Example:** `Router# show forwarding distribution multicast route group 226.1.1.1`	(Optional) Displays information about the multicast Forwarding Information Base (FIB) distribution routes.

5.4 Configuration Example for LISP Multicast

5.4.1 Example: Configuring LISP Multicast

The following example shows how to configure Locator/ID Separation Protocol (LISP) Multicast on either the Egress Tunnel Router (ETR) or the Ingress Tunnel Router (ITR):

```
vrf context vrf1
  ip pim rp-address 35.0.0.1 group-list
  224.0.0.0/4 ip pim ssm range 232.0.0.0/8
  ip lisp itr-etr <<< this router acts as a Lisp xTR gateway
  ip lisp database-mapping 20.0.0.0/24 11.0.0.1 priority 1 weight 100
  lisp instance-id 1
  ip lisp locator-vrf
  default
  ip lisp itr map-resolver 11.0.0.2
  ip lisp etr map-server 11.0.0.2 key 3
  5b0f2bd760fe4ce3  ip lisp multicast <<< this
  router supports Lisp Multicast
```

5.5 Feature History for LISP Multicast

Table 6 : Feature History for LISP Multicast

Feature Name	Releases
Locator/ID Separation Protocol (LISP) Multicast	8.3(1) e

CHAPTER 6 LISP Support for Disjointed RLOC Domains

This chapter contains the following sections:
·LISP Support for Disjointed RLOC Domains.

6.1 LISP Support for Disjointed RLOC Domains

6.1.1 Overview of LISP Support for Disjointed RLOC Domains

Locator/ID Separation Protocol (LISP) implements a level of indirection that enables a new IP routing architecture. LISP separates IP addresses into two address spaces, Endpoint Identifiers (EIDs), which are assigned to end hosts, and Routing Locators (RLOCs), which are assigned to devices that make up the global routing system.

This feature enables communication between LISP sites that are connected to different RLOC spaces and have no connectivity to each other.

6.1.2 Prerequisites for LISP Support for Disjointed RLOC Domains

·You understand how LISP works, including infrastructure, workflow, roles and functions.

6.1.3 Information About LISP Support for Disjointed RLOC Domains

The fundamental principle of any network is that routing and reachability should exist between all devices that make up the total network system. There are many network systems, public and private, for which internetwork connectivity is not directly available.

·A Multiprotocol Label Switching (MPLS) IPv4 VPN from service provider A and an MPLS IPv4 VPN from service provider B, with different scopes, 10.1.0.0/16 and 10.2.0.0/16.

·An MPLS IPv4 VPN from service provider A and IPv4 internet.

When some sites within a network connect to one routing domain and other sites connect to another routing domain, a gateway function must be provided to facilitate connectivity between these disjointed routing domains. In traditional routing architectures, providing connectivity between disjointed routing domains can be quite complex.

The inherent property of LISP, which separates IP addresses into two address spaces, gives it the ability to connect disjointed RLOC domains through simplified configuration mechanisms. The key components are new control plane configuration options on the LISP Map-Server, and the Re-encapsulating Tunnel Router (RTR) function, which provides data plane connectivity between disjointed locator spaces. The components and the workflow are explained.

LISP Map-Server

LISP RTR

When a LISP site registers with the Map-Server, it provides RLOC information. Ensure that all relevant RLOCs are registered with the Map-Server. Map-Server configurations are required to enable connectivity across RLOC spaces.

An RTR provides data plane communications support for LISP to LISP traffic between LISP sites that do not share common locator space. Functionally, an RTR takes in LISP encapsulated packets from an Ingress Tunnel Router (ITR) in one locator scope, decapsulates them, checks the map-cache, and then re-encapsulates them to an Egress Tunnel Router (ETR) in another locator scope. The following are important considerations for an RTR:

·RTR should have RLOCs in all locator scopes that are being joined.

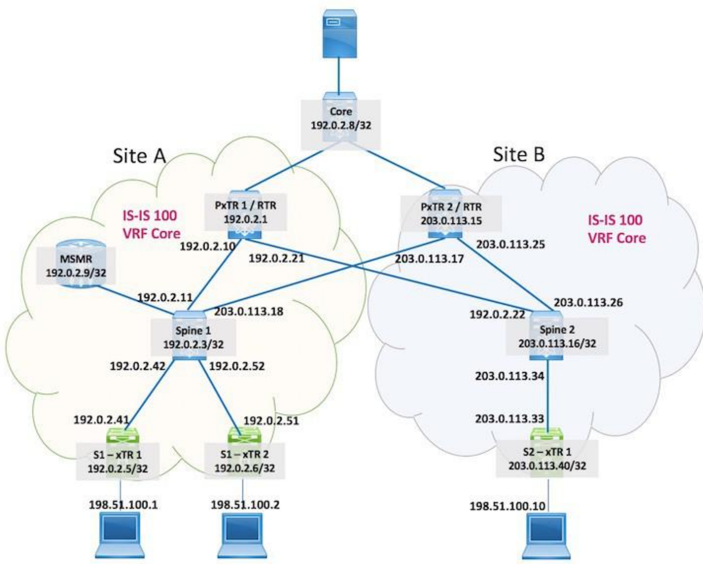
·An RTR sends Map-Request messages to populate its map-cache. As a Map-Request message contains an ITR RLOC field that is populated with one or more entries corresponding to the locators of the device sending the Map-

Request message, locator set configuration is required on the RTR to define its locators. This enables the Map-Server to correctly receive Map-Request messages from the RTR to assess locator scope connectivity.

·Since an RTR performs functions similar to a Proxy Ingress Tunnel Router (PITR) and Proxy Egress Tunnel Router (PETR), the PITR and PETR features must be enabled on the RTR.

Workflow of LISP Support for Disjointed RLOC Domains

Figure 19Figure 19: Sample topology—LISP Support for Disjointed RLOC Domains



For connecting disjointed RLOC domains in topology:

·Specified prefixes form the EID space in site A and site B.

·Ingress and Egress tunnel routers (referred as xTRs) represent the LISP site routers. xTR 1 and xTR 2 in site A have RLOC connectivity to one locator space, and the xTR in site B has RLOC connectivity to a different locator space.

·The RTR (PxTR 1, PxTR 2) is the LISP data plane device that enables communication between end hosts in the two sites, across locator spaces.

·Two virtual routing and forwarding (VRF) instances are created on the RTRs, one for the underlay (VRF *core*), and one for the overlay (VRF vrf5000).

An end host connected to xTR 1 in site A sends traffic to an end host attached to the xTR in site B. Since the source and destination RLOCs are from different RLOC spaces, PxTR 1 performs the role of RTR to transport traffic across the RLOC spaces. The detailed workflow:

1. xTR 1 (acting as an ITR) receives traffic from an attached end host, and sends a Map-Request for the destination EID (198.51.100.10), to the Map-Server (denoted by the IP address 192.0.2.9/32).
2. The Map-Server responds with a proxy reply containing the configured RTR locators (with IP addresses 192.0.2.1 and 203.0.113.15). The Map-Server does because the ITR-RLOC in the Map-Request from xTR 1 contains the RLOC from site A.
3. xTR 1 populates its map-cache with locator information (that is, PxTR 1 and PxTR 2 RLOCs) for the RTRs.
4. xTR 1 encapsulates LISP traffic and forwards it to the RTR in the data plane.
5. The RTR decapsulates the ingress LISP traffic and sends a Map-Request to the Map-Server for the destination EID, for the first packet.
6. The ITR-RLOC of the Map-Request comprises the locators configured under the locator set. The locators are 192.0.2.10 and 192.0.2.21.

A Map-Request is sent because the static map-cache is configured with the **map-request** command.

1. The Map-Server forwards the Map-Request to the ETR. The Map-Server does because the ITR-RLOC in the Map-Request from the RTR contains RLOCs from site A and site B.
2. The ETR replies to the RTR with the ETR locator information.
3. The RTR populates its map-cache with the ETR locator information.
4. The RTR re-encapsulates LISP traffic forwards the ETR.
5. The ETR receives and sends traffic to the destination end host.

PxTR 1 or RTR Configuration

Step 1 Configure LISP

```
(config)# feature lisp
```

Step 2 Create two VRF instances on the RTR, one for the underlay (VRF core), and one for the overlay (VRF *vrf5000*).

Configure LISP parameters for the core VRF

```
(config)# vrf context
core
            ip lisp
            itr-etr
            ip lisp itr map-resolver 192.0.2.9/32
            ip lisp etr map-server 192.0.2.9/32 key 3
            a97b0defe7b8ff70 ip lisp multicast
            lisp encapsulation vxlan
```

·After configuring the LISP ITR and ETR functions on PxTR 1, the LISP Map-Resolver (used by the ITR to send Map-Requests) and Map-Server (used by the ETR to register EIDs) locator addresses are configured.

·Also, LISP multicast transport and LISP Virtual Extensible LAN (VXLAN) encapsulation functions are enabled.

Configure LISP parameters for the vrf5000 VRF

```
 (config)# vrf context
vrf5000
            ip lisp proxy-itr
            192.0.2.1 ip lisp
            proxy-etr
            lisp instance-id 5000
```

The following configuration chunk is specific to connecting disjointed RLOC spaces.

```
lisp locator-set set5000
    192.0.2.10 priority 1 weight 10
    192.0.2.21 priority 2
    weight 20 exit
lisp map-request itr-rlocs
set5000 ip lisp locator-
vrf core
ip  lisp  map-cache  198.51.100.1/24
map-request   ip   lisp   map-cache
198.51.100.2/24  map-request  ip  lisp
multicast
lisp encapsulation vxlan
```

·The **lisp locator-set** command specifies a locator set for RTR RLOCs. 192.0.2.10 and 192.0.2.21 are the RLOCs connecting the RTR to each IPv4 locator space.

·The **lisp map-request itr-rlocs** command defines RTR RLOCs used in the Map-Request messages generated by the RTR. You can enable multiple locator sets, but only one of them can be active at a point in time, and that is determined by including the name in the **lisp map-request itr-rlocs** option.

·Since Map-Resolver and Map-Server addresses are enabled in VRF core, VRF core is referenced within VRF vrf5000, in the **locator-vrf core** command.

Step 3 Configure an IP address for routing in the underlay

```
(config)# interface
loopback0
             vrf
             member
             core
             ip address
             192.0.2.1/32 isis
             circuit-type level-
             1-2 ip router isis
             100
             ip pim sparse-mode
```

The configured loopback interface IP address is used for IS-IS communication within the LISP site, and is added to VRF core.

Step 4 The configurations are relevant for RTR or PxTR 1. Similarly, configure the RTR or PxTR 2 device too.

PxTR 1 or RTR Configuration—RTR Locator-Set Inheritance

An RTR locator set can be defined in the underlay VRF and can then be referenced in an overlay VRF.

Step 1 Configure LISP

```
(config)# feature lisp
```

Step 2 Create two VRF instances on the RTR, one for the underlay (VRF core), and one for the overlay (VRF *vrf5000*).

Configure LISP parameters for the core VRF

```
(config)# vrf context core
            ip lisp itr-etr
            ip lisp itr map-resolver 192.0.2.9/32
            ip lisp etr map-server 192.0.2.9/32 key 3
            a97b0defe7b8ff70 lisp locator-set setCore
               192.0.2.10 priority 1 weight 10
               192.0.2.21 priority
               2 weight 20 exit
            ip lisp multicast
            lisp encapsulation vxlan
```

Configure LISP parameters for the vrf5000 VRF

```
(config)# vrf context vrf5000
            ip lisp proxy-
            itr 192.0.2.1 ip
            lisp proxy-etr
            lisp instance-id 5000
            lisp map-request itr-
            rlocs setCore ip lisp
            locator-vrf core
            ip  lisp  map-cache  198.51.100.1/24
            map-request   ip   lisp   map-cache
            198.51.100.2/24   map-request   ip
            lisp multicast
            lisp encapsulation vxlan
```

Step 3 Configure an IP address for routing in the underlay

```
(config)# interface loopback0
            vrf member core
            ip address
            192.0.2.1/32
            isis circuit-
            type level-1-2
            ip router isis
            100
            ip pim sparse-mode
```

The configured loopback interface IP address is used for IS-IS communication within the LISP site, and is added to VRF core.

Step 4 The configurations are relevant for RTR or PxTR 1. Similarly, configure the RTR or PxTR 2 device too.

Map-Server configuration on a device with IOS XE software (not Inspur CN12700 Series device):

```
(config)# router lisp
            locator-table
            vrf core
            locator-set
            SITEAB
               192.0.2.1 priority 1
               weight 50 203.0.113.15
               priority 1 weight 50 exit
            locator-scope site-B
               rtr-locator-set SITEAB
               rloc-prefix
               203.0.113.40/32
               rloc-prefix
               192.0.2.21/32 rloc-
               prefix
               203.0.113.25/32
               exit
            locator-scope site-A
               rtr-locator-set
               SITEAB rloc-
               prefix
               192.0.2.5/32
               rloc-prefix
               192.0.2.6/32
               rloc-prefix
               203.0.113.17/32
               rloc-prefix
               192.0.2.10/32
```

6.1.4 Verifying LISP Support for Disjointed RLOC Domains

Testing Reachability from xTR 1 in Site A to the xTR in Site B

In the following example, locator information for both sites (192.0.2.1 in site A and 203.0.113.15 in site B) are displayed. xTR 1 in site A is connected to the xTR in site B.

```
siteA-xTR1# lig 198.51.100.10 instance-id 5000

Mapping information for EID 198.51.100.10 from 192.0.2.9/32 with RTT 2
msecs 198.51.100.10/32, uptime: 00:07:06, expires: 00:14:59, via map-
reply, complete
```

```
Locator        Uptime    State      Pri/Wgt

192.0.2.1      00:07:06  up           1/50
203.0.113.15   00:07:06  up           1/50
```

Testing Reachability from PxTR 1 to the xTR in Site B

In the following example, Map-Request, Map-Reply, and map-cache information is displayed. Also, locator information for the xTR in site B is displayed. This signifies that PxTR 1 is connected to the xTR in site B.

```
PxTR1# lig 198.51.100.10 vrf vrf5000

Send map-request to 192.0.2.9 for [5000]
100.3.3.10 ... Received map-reply from
203.0.113.40 with rtt 0.003248 secs Map-cache
entry for [5000] EID 198.51.100.10 is:
198.51.100.10/32, uptime: 05:05:47, expires: 23:59:58, via map-reply, auth

Locator       Uptime   State  Priority/Data Control  MTU  Weight
              in/out   in/out 203.0.113.40 05:05:47  up
              10/10    0/0*   2/0  1500
```

EID Space Details in the Map-Server/Map-Resolver (MSMR)

In the following example, you can see that the client with the specified EID, attached to the xTR in site B, is registered with the MSMR. The specified EID, Instance ID and corresponding locator is displayed.

```
MSMR# show lisp site 198.51.100.10 instance-id 5000

LISP Site Registration Information
Site name: site-AAllowed configured locators:
anyRequested EID-prefix: EID-prefix:
198.51.100.10/32 instance-id 5000
            First registered:     00:19:46
            Last registered:      00:19:46 Routing table tag: 0
            Origin:               Dynamic, more specific of 203.0.0.0/16
            Merge active:         No
            Proxy reply:          No
            TTL:                  1d00h
            State:                complete
            Registration
            errors:
            Authentication
            failures:                  0
            Allowed locators mismatch: 0

ETR 203.0.113.40, last registered 00:19:46, no proxy-reply, map-notify
            TTL 1d00h, no merge, hash-function sha1, nonce 0x4CC82237-
            0x6DCB0FC5

            state complete, no security-capability
            xTR-ID 0x90FA8033-0x867FE73F-0x5F32076D-
            0xE92E8945
            site-ID unspecified
            sourced by reliable transport

Locator       Local  State
              Pri/Wgt  Scope
203.0.113.40  yes    up
              10/10  site-B
```

In the following example, corresponding LISP site information for the MSMR is displayed. The information includes, EID, IID, and locator information.

```
MSMR# show lisp site detail

EID-prefix: 198.51.100.10/32
instance-id 5000 First registered:
                      08:12:10
Last registered:      08:12:10 Routing table tag:     0
Origin:               Dynamic, more specific of
203.0.0.0/16


Merge active:

No
Proxy reply:

No
TTL:

1d00h
State:                complete

Registration errors:
Authentication
failures:                  0 Allowed locators mismatch: 0

ETR 203.0.113.40, last registered 08:12:10, no proxy-reply, map-notify
                   TTL 1d00h, no merge, hash-function sha1, nonce 0x4CC82237-
                   0x6DCB0FC5

                   state complete, no security-capability
                   xTR-ID 0x90FA8033-0x867FE73F-0x5F32076D-0xE92E8945
               site-ID unspecified
                   sourced by reliable transport

Locator       Local  State Pri/Wgt
Scope 203.0.113.40  yes    up       10/10      site-B
```

Verify LISP map-cache Details on PxTR 1

In the following example, map-cache details corresponding to PxTR 1 for the specified EID are displayed. The information includes locator information.

```
PxTR1# show ip lisp map-cache 198.51.100.1 vrf vrf5001

LISP IP Mapping Cache for VRF "vrf5001" (iid 5001), 16 entries
* = Locator data counters are cumulative across all EID-prefixes

198.51.100.1/32, uptime: 1d03h, expires: 20:01:07, via map-
  reply, auth Last activity: 03:58:42
  State: complete, last modified: 1d03h, map-source:
  192.0.2.5 Pending hw update: FALSE
  Locator          Uptime    State       Priority/  Data      Control       MTU
                                         Weight     in/out    in/out
  192.0.2.5        1d03h     up          10/10      0/0*      2/0          1500

  Last up/down state change:           1d03h, state change
  count: 0 Last data packet in/out:    never/1d03h
  Last control packet in/out:
                                       03:58:
  52/never Last priority/weight change:
```

never/
never

6.1.5 Feature History for LISP Support for Disjointed RLOC Domains

This table lists the release history for this feature.

Table 7 : Feature History for LISP Support for Disjointed RLOC Domains

Feature Name	**Release**	**Feature Information**
Connecting LISP Disjointed RLOC Domains	8.3(1)	This feature was introduced.

CHAPTER 7 Configuring LISP Extranets

This chapter contains the following sections:
·Finding Feature Information.
·Feature History for LISP Extranets.
·Information About LISP Extranets.
·Licensing Requirements for LISP.
·Guidelines and Limitations for LISP Extranets.
·Configuring LISP Extranets.

7.1 Finding Feature Information

Your software release might not support all the features documented in this module. 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 "New and Changed Information"chapter or the Feature History table in this chapter.

7.2 Feature History for LISP Extranets

Table 8: Feature History for LISP Extranets

Feature Name	Releases	Feature Information
Locator ID Separation Protocol (LISP) Extranets	8.3(1)	This feature is supported on Inspur CN12700 Series switches.

7.3 Information About LISP Extranets

Starting from Inspur INOS 8.3(1), LISP Extranets support is added to the Locator ID Separator Protocol (LISP) in Inspur INOS.

Campus fabric architecture for enterprise network uses LISP as its overlay control protocol. LISP based deployments use the LISP Virtualization solution to provide segmentation, isolation, and security among the network elements. A network that uses LISP virtualization binds VRFs to instance IDs (IIDs) and these IIDs are used to support traffic flow segmentation across the overlay network. LISP learned mappings are kept within the same instance ID (IID) or VRF context and are not shared across IIDs or VRFs, which means that a host/resource can only talk to hosts/resources in VRFs with same IID.

With the LISP Extranets feature, users can specify policies that allows host and resources residing in one VRF (IID) domain to communicate with hosts in a separate VRF (IID) domain.

With LISP Extranets policies are specified in the Mapping System and the xTRs (Ingress Tunnel Router + Egress Tunnel Router) discover the leaked routes on demand, as part of the regular route discovery process.

The implementation of LISP Extranets on LISP includes the following features:

·A Map Server (MS) device running Inspur IOS XE Everest 16.9.1 release or later, where the user can establish LISP Extranet policies.

·A VRF with valid LISP instance-ID configuration that can be configured to handle leaked map-caches in LISP. This support is automatically provided on LISP from Inspur INOS Release 8.3(1) and later.

7.3.1 Use Case for LISP Extranets

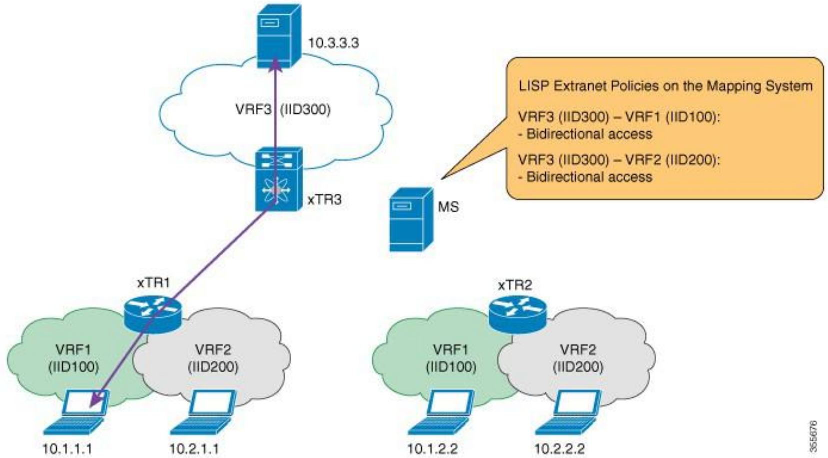


Figure 20 : Inspur INOS LISP Extranets Topology

The above figure illustrates the use of LISP Extranet policies. In the figure, there is a LISP shared virtualization environment where routers (xTR1 and xTR2) extend to two VRF domains (VRF1 and VRF 2) using LISP virtualization over an overlay fabric. A third router, xTR3, provides access to a different VRF domain (VRF3). Each VRF domain is given a unique instance-ID (IID) value. According to LISP virtualization, hosts in a IID domain can access resources within their same IID domain. For example, host 10.1.1.1 can only communicate with host 10.1.2.2 because they are connected to the same IID domain (IID 100), but cannot communicate with host 10.3.3.3 as it is connected to a different IID domain (IID 300).

Using the LISP Extranets feature, a user can establish a leaking policy, wherein the traffic can cross the IID boundaries. For example, if host 10.1.1.1 needs to communicate/share policies with host 10.3.3.3, network administrators can configure an extranet policy on the LISP Mapping System, xTR3 will dynamically discover the policy and allow host 10.1.1.1 and host 10.3.3.3 to talk to each other across IID boundaries.

Packet Flow in LISP Extranets Through Map Server Policies

In the following diagram, H1 (10.1.1.1) wants to send packet to H2 (10.3.3.3). Both hosts reside in different IID domains, but a LISP extranet policy setup in the Mapping System (MS) allows the traffic to cross the IID boundaries.

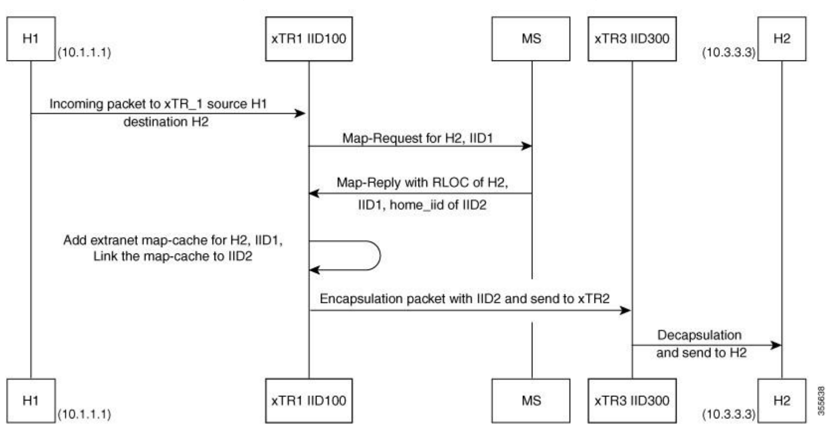


Figure 21 : Packet Flow in LISP VRF Leaking

1. H1 (10.1.1.1) on IID 100 sends a packet to H2 (10.3.3.3) on IID 300.
2. The packet first arrives on IID 100 residing on xTR1.IID 100 on xTR1 determines it is a cache miss and sends a

map-request to the Mapping System.

3. Map server receives the map request from xTR1 for mapping details on H2.

• The MS uses its extranet policy configuration to determine if communication is allowed between H1 on IID 100 and H2 on IID 300.

• The MS permits communication between these two IIDs. MS sends a map-reply back to IID 100 on xTR1, with an additional parameter called Home IID. In this example the Home IID is IID 300.

4. xTR1 receives the map-reply from the MS. It sees that it contains a valid Home IID within the map reply.

·The Home IID is different from the IID (IID 100) that was used to send the map request.

·xTR1 learns this mapping as an extranet route.

5. Creates a map-cache for the H2 EID prefix into IID 100 on xTR1 with the property that the traffic needs to be encapsulated using IID 300.

6. Traffic from H1 to H2 is encapsulated using IID 300 based on the new map-cache.

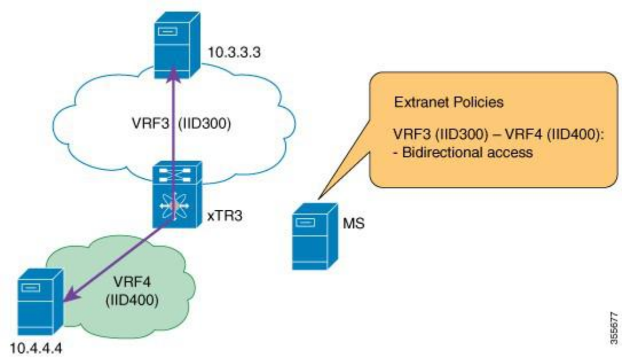
7. xTR3 receives traffic with IID 300, decapsulates the traffic and forwards the packet to H2.

7.3.2 Use Case for LISP Local Extranet Policies

Another benefit of the LISP Extranets feature is that a user can provision dynamic local VRF Leaking support. A user can specify an Extranet policy on the Mapping System (MS), which the LISP xTRs will apply locally. But this is provided only on demand basis, following traffic requests.

In the below illustration, LISP router (xTR3) provides routing access to two separate VRF domains (VRF3 and VRF4) that are locally connected to the router. If a user configures a LISP Extranet policy on the MS that allows IID 300 and IID 400 to communicate with each other, xTR3 dynamically discovers the policy and allows host 10.3.3.3 and host 10.4.4.4 to talk to each other across VRF boundaries.

Figure 22 : LISP Local Extranets



7.4 Licensing Requirements for LISP

The following table shows the LISP licensing requirements:

Product	License Requirement
Inspur INOS	This feature requires the LAN_ENTERPRISE_SERVICES_PKG license. For a complete explanation of the Inspur INOS licensing scheme, see the *Inspur INOS Licensing Guide*.

7.5 Guidelines and Limitations for LISP Extranets

LISP has the following configuration guidelines and limitations for the LISP Extranets feature:

• Only one provider IID is supported per policy configuration on Inspur IOS XE Everest 16.9.1 release.

7.6 Configuring LISP Extranets

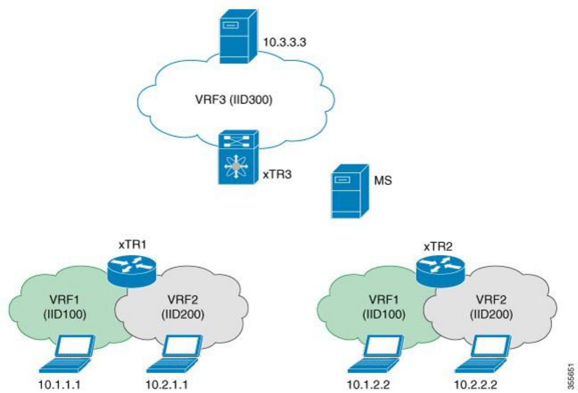
This section includes the following topics:

7.6.1 Configuring LISP Map Server with Extranet Policies

The LISP Extranet feature is configured through the extranet policies. Users can configure these policies as part of the Map Server (MS) configuration and the xTR routers will dynamically learn the policies.

The LISP Extranet feature supports both IPv4 and IPv6 address families.

Figure 23 : LISP Extranet Topology



In this example, the following devices are used:

- Map Server (MS) device with Inspur IOS XE Everest 16.9.1 release and later.
- xTR device with Inspur INOS Release 8.3(1) and later.
- The xTR1 and xTR2 have a mask length of /24, and the xTR3 has mask length of /16.

Before you begin

Create VRFs using the vrf context command. Enable feature LISP.

Procedure

Step 1 Configure the MS for IPv4 and IPv6 services.

```
MS# configure terminal
MS(config)# router lisp
MS(config-router-lisp)#
service ipv4 MS(config-
router-lisp)# map-server
MS(config-router-lisp)# map-
resolver MS(config-router-
lisp)# exit-service-ipv4
 !
MS(config-router-lisp)#
service ipv6 MS(config-
router-lisp)# map-server
MS(config-router-lisp)# map-
resolver MS(config-router-
lisp)# exit-service-ipv6
```

Step 2 Configure the LISP sites.

In the following configuration, there are three instance-IDs (100, 200, and 300).

```
MS# configure terminal
```

```
MS(config)# site SITE_ALL_v4
MS(config-lisp-site)# authentication-key Inspur
MS(config-lisp-site)# eid-record instance-id 100 10.1.0.0/16 accept-
more-specifics MS(config-lisp-site)# eid-record instance-id 200
10.2.0.0/16 accept-more-specifics MS(config-lisp-site)# eid-record
instance-id 300 10.3.0.0/16 accept-more-specifics

! Configure
another site!
MS(config)# site
SITE_ALL_v6
MS(config-lisp-site)# authentication-key Inspur
MS(config-lisp-site)# eid-record instance-id 100 2001:DB8:1::/48 accept-
more-specifics MS(config-lisp-site)# eid-record instance-id 200
2001:DB8:2::/48 accept-more-specifics MS(config-lisp-site)# eid-record
instance-id 300 2001:DB8:3::/48 accept-more-specifics
```

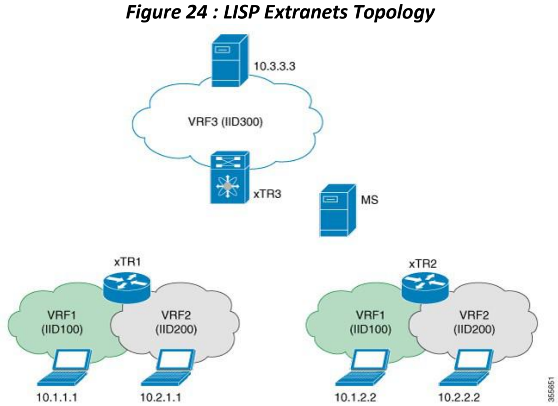
Step 3 Configure the LISP Extranet policies on MS. The **eid-record-provider instance-id** and **eid-record-subscriber instance-id** commands define the extranet policy for the provider and subscriber instances.

```
MS(config-router-lisp)# extranet ext_policy_1
MS(config-router-lisp-extranet)# eid-record-provider instance-id 300
MS(config-router-lisp-extranet)# ip-any
MS(config-router-lisp-extranet)# exit-eid-record-
provider

!
MS(config-router-lisp-extranet)# eid-record-subscriber instance-id 100
MS(config-router-lisp-extranet)# ip-any
MS(config-router-lisp-extranet)# exit-eid-record-
subscriber

!
MS(config-router-lisp-extranet)# eid-record-subscriber instance-id 200
MS(config-router-lisp-extranet)# ip-any
MS(config-router-lisp-extranet)# exit-eid-record-
subscriber
```

7.6.2 Configuring LISP xTR functionality for Extranet Policies

The LISP Extranet support is enabled once the Map Server (MS) device is configured with the Inspur IOS Everest 16.9.1 release and later. The xTRs in a LISP network dynamically learn the policies, and allow hosts from one VRF IID to talk to hosts from other VRF IIDs.

Figure 24 : LISP Extranets Topology



Procedure

Configure the xTRs for LISP. xTR3 configurations

```
!xTR3
Configurations!
switch# configure terminal
switch(config)# vrf context VRF3
!switch(config-vrf)# ip lisp itr-etr
!switch(config-vrf)# ipv6 lisp itr-etr
switch(config-vrf)# ip lisp database-mapping 10.3.0.0/16 10.10.10.3 priority 1
weight 100
switch(config-vrf)# ipv6 lisp database-mapping 2001:DB8:3::/48 10.10.10.3 priority
1 weight 100
switch(config-vrf)# lisp instance-id 300
switch(config-vrf)# ip lisp locator-vrf underlay
switch(config-vrf)# ipv6 lisp locator-vrf underlay
```

xTR1 configurations

```
!xTR1 Configurations!
switch# configure terminal

switch(config)#vrf context VRF1
switch(config-vrf)# ip lisp itr-etr
switch(config-vrf)# ipv6 lisp itr-etr
switch(config-vrf)# ip lisp database-mapping 10.1.1.0/24 10.10.10.1 priority
1 weight 100

switch(config-vrf)# ipv6 lisp database-mapping 2001:DB8:1:1::/64 10.10.10.1
priority 1 weight 100
switch(config-vrf)# lisp instance-id 100

switch(config-vrf)# ip lisp locator-vrf underlay

switch(config-vrf)# ipv6 lisp locator-vrf underlay
switch(config-vrf)# exit

switch(config)# vrf context VRF2
switch(config-vrf)# ip lisp itr-etr
switch(config-vrf)# ipv6 lisp itr-etr
switch(config-vrf)# ip lisp database-mapping 10.2.1.0/24 10.10.10.1 priority
1 weight 100
switch(config-vrf)# ipv6 lisp database-mapping 2001:DB8:2:1::/64 10.10.10.1
priority 1 weight 100
switch(config-vrf)# lisp instance-id 200
switch(config-vrf)# ip lisp locator-vrf underlay
switch(config-vrf)# ipv6 lisp locator-vrf underlay
switch(config-vrf)# exit
```

7.6.3 Verifying LISP Extranets

Use the following show commands to verify the LISP Extranet policy configurations.

Procedure

show ip lisp map-cache vrf *vrf-name*

Displays the current dynamic and static IPv4 endpoint identifier-to-routing locator (EID-to-RLOC) map-cache entries for a VRF.

xTR3 verifications:

```
switch# show ip lisp map-cache vrf VRF3

LISP IP Mapping Cache for VRF ”VRF3" (iid 300), 2
entries

10.1.1.0/24, uptime: 00:00:16, expires: 23:59:43, via map-reply, non-auth Producer
  Set: 4000 map-reply

Encap-IID  Locator     Uptime    State       Priority/  Data      Control       MTU
                                             Weight     in/out    in/out
           10.10.10.1  00:00:16  up          1/100      0/4       0/0          1500 100

10.2.1.0/24, uptime: 00:00:16, expires: 23:59:43, via map-reply, non-auth Producer
  Set: 4000 map-reply

Encap-IID  Locator     Uptime    State       Priority/  Data      Control       MTU
                                             Weight     in/out    in/out
           10.10.10.1  00:00:16  up          1/100      0/2       0/0          1500 200
```

The map-cache shown above is the source map-cache. In this example, the source map-cache lives on the VRF called VRF3 with an IID of 300. The source map-cache contains an additional field called Encap-IID. The traffic flowing through this map-cache will be encapsulated using its packets using the Encap-IID 100 and Encap-IID 200.

```
switch# show ip route vrf VRF3

IP Route Table for

VRF ”VRF3” 10.1.1.0/24,

ubest/mbest: 1/0 time
    *via 10.10.10.1%vrf-underlay, nve1, [240/2], 00:39:27, lisp, eid, segid: 100,
(Remote VNI) tunnelid: 0xa0a0a01 encap: VXLAN

10.2.1.0/24, ubest/mbest: 1/0 time
    *via 10.10.10.1%vrf-underlay, nve1, [240/2], 00:39:27, lisp, eid, segid: 200,
(Remote VNI) tunnelid: 0xa0a0a01 encap: VXLAN
```

xTR1 verifications

```
switch# show ip lisp map-cache vrf VRF1

LISP IP Mapping Cache for VRF ”VRF1" (iid 100), 1 entries

10.3.0.0/16, uptime: 00:00:20, expires: 23:49:43, via map-reply, non-auth Producer
  Set: 4000 map-reply

Encap-IID  Locator     Uptime    State       Priority/  Data      Control       MTU
                                             Weight     in/out    in/out
           10.10.10.3  00:00:20  up          1/100      0/0       0/0          1500 300

switch# show ip route vrf VRF1

IP Route Table for

VRF ”VRF1” 10.3.0.0/16,

ubest/mbest: 1/0 time
    *via 10.10.10.3%vrf-underlay, nve1, [240/2], 00:49:27, lisp, eid, segid: 300,
(Remote VNI) tunnelid: 0xa0a0a03 encap: VXLAN

switch# show ip lisp map-cache vrf VRF2
```

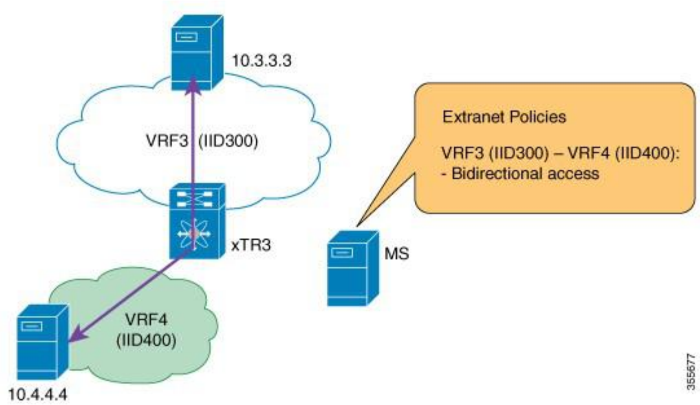
```
LISP IP Mapping Cache for VRF ”VRF2" (iid 200), 1 entries

10.3.0.0/16, uptime: 00:00:20, expires: 23:49:43, via map-reply, non-auth Producer
  Set: 4000 map-reply

Encap-IID      Locator     Uptime    State       Priority/  Data      Control      MTU
                                                 Weight     in/out    in/out
               10.10.10.3  00:00:20  up          1/100      0/0       0/0          1500
```

```
switch# show ip route vrf VRF2

IP Route Table for VRF ”VRF2”

300

10.3.0.0/16, ubest/mbest: 1/0 time
    *via 10.10.10.3%vrf-underlay, nve1, [240/2], 00:49:27, lisp, eid, segid: 300,
(Remote VNI) tunnelid: 0xa0a0a03 encap: VXLAN
```

7.6.4 Configuring Local LISP Extranets

Users can share resources across VRFs on the same device using LISP Extranets. All the configurations are done on the Map Server (MS). The xTRs in a LISP network dynamically learn the policies and share the policies across VRFs.

Figure 25 : Local LISP Extranets



Before you begin

Enable feature LISP.

Procedure

Configure the xTR device.

```
!xTR3
Configurations!
switch#
configure
terminal
switch(config)# vrf
context VRF3
switch(config-vrf)# ip
lisp itr-etr
switch(config-vrf)# ipv6
lisp itr-etr
```

```
switch(config-vrf)# ip lisp database-mapping 10.3.0.0/16 10.10.10.3 priority 1
weight 100
switch(config-vrf)# ipv6 lisp database-mapping 2001:DB8:3::/48 10.10.10.3
 priority 1 weight 100
switch(config-vrf)# lisp instance-id 300

switch(config-vrf)# ip lisp locator-vrf underlay

switch(config-vrf)# ipv6 lisp locator-vrf underlay

switch# configure terminal
switch(config)#vrf context VRF4
switch(config-vrf)# ip lisp itr-etr
switch(config-vrf)# ipv6 lisp itr-etr
switch(config-vrf)# ip lisp database-mapping 10.4.0.0/16 10.10.10.3 priority 1
weight 100
switch(config-vrf)# ipv6 lisp database-mapping 2001:DB8:4::/48 10.10.10.3
 priority 1 weight 100
switch(config-vrf)# lisp instance-id 400
switch(config-vrf)# ip lisp locator-vrf underlay
switch(config-vrf)# ipv6 lisp locator-vrf underlay
switch(config-vrf)# exit
```

7.6.5 Verifying LISP Local Extranets

Use the following show command to verify the LISP local Extranet policy configurations.

Procedure

show ip lisp database vrf *vrf-name*

Displays Locator ID Separation Protocol (LISP) Egress Tunnel Router (ETR) configured local IPv4 EID prefixes and associated locator sets for a VRF.

xTR3 verifications:

```
switch# show ip lisp database vrf VRF3

LISP ETR IP Mapping Database for VRF ”VRF3" (iid 300), global LSBs:

0x00000001 Local Database: 2
EID-prefix: 10.3.0.0/16, instance-id: 300, LSBs:
 0x00000001 Producer: static , locator_set: Reserved-
 0, uptime: 19:11:25 Locator: 10.10.10.3, priority: 1,
 weight: 100
          Uptime: 19:11:25, state: up, local
EID-prefix: 10.4.0.0/16, instance-id: 400, LSBs:
 0x00000001 Producer: leaked , locator_set: ,
 uptime: 19:11:25


switch# show ip lisp database vrf VRF4

LISP ETR IP Mapping Database for VRF ”VRF4" (iid 400), global LSBs:

0x00000001 Local Database: 2
EID-prefix: 10.3.0.0/16, instance-id: 300, LSBs:
 0x00000001 Producer: leaked , locator_set: ,
 uptime: 19:11:25
EID-prefix: 10.4.0.0/16, instance-id: 400, LSBs:
 0x00000001 Producer: static , locator_set: Reserved-
```

```
 0, uptime: 19:11:25 Locator: 10.10.10.3, priority: 1,
 weight: 100
           Uptime: 19:11:25, state:

up, local switch# show ip route vrf

VRF3

IP Route Table for VRF ”VRF3”
10.3.0.0/16, ubest/mbest: 1/0 time, attached
    *via 10.3.0.1, Vlan300, [0/0], 00:49:27,
direct 10.4.0.0/16, ubest/mbest: 1/0 time
    *via 10.4.0.1%VRF4, Vlan400, [10/1], 00:49:27,

lisp, eid switch# show ip route vrf VRF4

IP Route Table for VRF ”VRF4”

10.3.0.0/16, ubest/mbest: 1/0 time
    *via 10.3.0.1%VRF3, Vlan300, [10/1], 00:49:27, lisp, eid

10.4.1.1/16, ubest/mbest: 1/0 time, attached
    *via 10.4.0.1, Vlan400, [0/0], 00:49:27, direct
```

CHAPTER 8 Redistribution of RIB Routes into LISP

This chapter contains the following sections:

- Finding Feature Information.
- Feature History for Redistribution of RIB Routes into LISP.
- Information About Redistribution of RIB Routes into LISP.
- Configuring Database Application for Redistribution of RIB Routes into LISP.
- Configuring Map-cache Application for Redistribution of RIB Routes into LISP.
- Example: Redistribution of RIB Routes in LISP.

8.1 Finding Feature Information

Your software release might not support all the features documented in this module. 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 "New and Changed Information"chapter or the Feature History table in this chapter.

8.2 Feature History for Redistribution of RIB Routes into LISP

Table 9 : Feature History for Redistribution of RIB Routes into LISP

Feature Name	Releases	Feature Information
Redistribution of RIB Routes into LISP	8.3(1)	This feature is supported on Inspur CN12700 Series switches.

8.3 Information About Redistribution of RIB Routes into LISP

Starting with Inspur INOS 8.3(1), the Locator ID Separation Protocol (LISP) supports the redistribution of RIB routes into LISP feature. This feature allows LISP to import Layer 3 RIB routes in use for internal applications. Importing information from the RIBs allows for proactive learning of LISP prefixes in the control plane. This eliminates the need to statically specify prefixes to be used for map-caches or databases in LISP.

The redistribution of RIB routes into LISP is enabled under the VRF Context and supports both IPV4 and IPV6 address families.

The following RIB sources are supported for LISP redistribution:

Table 10 : Supported RIB Sources for Route Import

Source	Description
bgp	Border Gateway Protocol (BGP)
direct	Connected
eigrp	Enhanced Interior Gateway Routing Protocol (EIGRP)
isis	ISO IS-IS
ospf	Open Shortest Path First (OSPF)
ospfv3	OSPFv3

rip	Routing Information Protocol (RIP)
static	Static routes

8.4 Configuring Database Application for Redistribution of RIB Routes into LISP

Locator ID Separation Protocol (LISP) egress tunnel routers (ETR) import prefixes with a local RIB route into the LISP EID database and register it with the Mapping System. Perform the following steps on a xTR device in a LISP topology.

Before you begin

- Enable the LISP feature.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**vrf context** *vrf-name* **Example:** `switch(config)# vrf context VRF1`	Creates a new VRF and enters VRF configuration mode. The value of the *vrf-name* is any case-sensitive, alphanumeric string of up to 32 characters.
Step 3	{**ip** | **ipv6**} **lisp etr** **Example:** `switch(config-vrf)# ip lisp etr`	Configures LISP ETR functionality for the VRF.
Step 4	**lisp instance-id** *iid* **Example:** `switch(config-vrf)# lisp instance-id 100`	Configures an instance ID to be associated with endpoint identifier (EID)-prefixes for LISP. The range is from 1 to 16777215.
Step 5	ip lisp locator-vrf {*locator-vrf* | **default**} **Example:** `switch(config-vrf)# ip lisp locator-vrf default`	Configures a non-default VRF table to be referenced by any IPv4 locators.
Step 6	{**ip** | **ipv6**} **lisp etr map-server** *map-server-address* **key** *authentication-key* **Example:** `switch(config-vrf)# ip lisp etr map-server 10.10.10.10 key 3 1c27564ab12121212`	Configures the IPv4 or IPv6 locator address of the LISP map server to be used by the egress tunnel router (ETR) when registering for IPv4 endpoint identifier (EIDs).

Step 7	{**ip** | **ipv6**} **lisp route-import database** *protocol autonomous-system-number* [**route-map** *map-name*] [**locator-set** *set*] **Example:** `switch(config-vrf)# ip lisp route-import database ospf 100 route-map RM_OSPF_to_LISP locator-set RLOCSET`	Configures the import of routes from the RIB to define endpoint identifier EID space on an ETR. The **route-map** keyword specifies that the imported IPv4 prefixes should be filtered according to the specified route-map name.
Step 8	(Optional) **ip lisp route-import database maximum-prefix** *prefix-number*	Configures the maximum number of prefixes that can be imported. The valid range is from 1 to 1000. The default value is 1000.
Step 9	(Optional) **show** [**ip** | **ipv6**] **lisp route-import database**	Displays the EID instance address family for route-import database.
Step 10	(Optional) **show** [**ip** | **ipv6**] **lisp database vrf** *vrf-name*	Displays LISP ETR configured local IPv4 EID prefixes and associated locator sets.

8.5 Configuring Map-cache Application for Redistribution of RIB Routes into LISP

The LISP Ingress Tunnel Routers (ITRs) import the remote EID map caches and program them into the platform.

```
switch# show ip lisp map-cache vrf  VRF1

LISP IP Mapping Cache for VRF ”VRF1" (iid 100), 1
entries
* = Locator data counters are cumulative across all EID-
prefixes

192.168.1.1/24, uptime: 00:02:48, expires: 0.000000, via route-import, self
  Producer Set: 0004 route-import
  Negative cache entry, action: send-map-request
```

Before you begin

- Enable the LISP feature.

Procedure

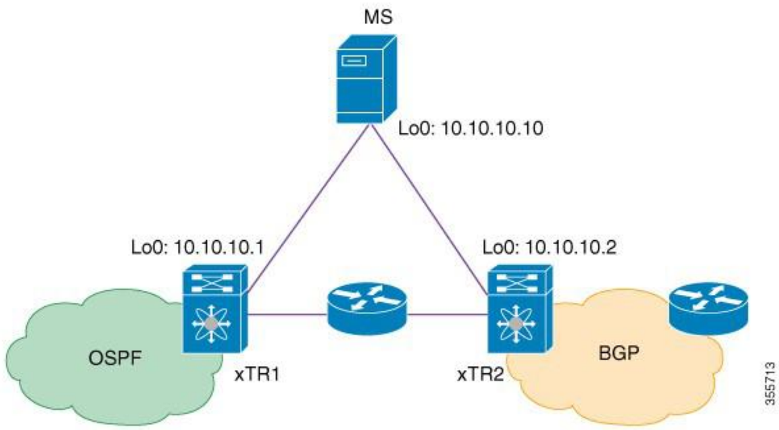
	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**vrf context** *vrf-name* **Example:** `switch(config)# vrf context VRF1`	Creates a new VRF and enters VRF configuration mode. The value of the **vrf-name** is any case-sensitive, alphanumeric string of up to 32 characters.

Step 3	Use one of the following commands: • {**ip** | **ipv6**} **lisp itr** • {**ip** | **ipv6**} **lisp proxy-itr** *locator-address* **Example:** `switch(config-vrf)# ip lisp itr`	Configures LISP ITR functionality for the VRF. Configures LISP Proxy-ITR functionality on the device. The *locator* address is used as a source address for encapsulating data packets or Map-Request messages. Optionally, you can provide an address for the other address family (for example, IPv6 for the **ip proxy-itr** command.
Step 4	**lisp instance-id** *iid* **Example:** `switch(config-vrf)# lisp instance-id 100`	Configures an instance ID to be associated with endpoint identifier (EID)-prefixes for LISP. The range is from 1 to 16777215.
Step 5	ip lisp locator-vrf {*locator-vrf* | **default**} **Example:** `switch(config-vrf)# ip lisp locator-vrf default`	Configures a non-default VRF table to be referenced by any IPv4 locators.
Step 6	{**ip** | **ipv6**} **lisp itr map-resolver** *map-resolver-address* **Example:** `switch(config-vrf)# ip lisp itr map-resolver 10.10.10.2`	Configures the locator address of the map-resolver to which the router sends map-request messages for IPv4 or IPv6 EIDs.
Step 7	{**ip** | **ipv6**} **lisp route-import map-cache** *protocol autonomous-system-number* [**route-map** *map-name*] **Example:** `switch(config-vrf)# ip lisp route-import map-cache bgp 65536 route-map RM_BGP_to_LISP`	Configures the import of routes from the RIB to define endpoint identifier EID space on an ITR. The **route-map** keyword specifies that the imported IPv4 prefixes should be filtered according to the specified route-map name.
Step 8	(Optional) **ip lisp route-import map-cache maximum-prefix** *prefix-number*	Configures the maximum number of prefixes that can be imported. The valid range is from 1 to 1000. The default value is 1000.
Step 9	(Optional) **show** [**ip** | **ipv6**] **lisp route-import map-cache**	Display the EID instance address family for route-import map-cache.
Step 10	(Optional) **show** [**ip** | **ipv6**] **lisp map-cache VRF** *vrf-name*	Displays LISP ITR configured local IPv4 EID prefixes.

8.6 Example: Redistribution of RIB Routes in LISP

The following example shows the redistribution of RIB routes in database and map-cache applications in a LISP topology.

Figure 26 : Redistribution of RIB Routes in a LISP Topology



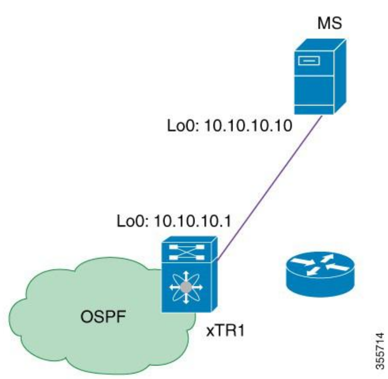
In the above image, xTR1 is provides global access to a local OSPF stub network and xTR2 connects to an external BGP cloud.

The following example shows a combined use of the two available route-import applications (database and map-cache) in a LISP overlay fabric network. It first describes how prefixes from the OSPF cloud are imported in xTR1 as database-mappings, and registered with the LISP mapping system, and then shows how the same prefixes are imported as map-caches on xTR2 so that data traffic follows the optimized LISP overlay path.

Configuring Route Import with the Database Application

This section describes the database application configurations and commands to verify the configurations. In this case, xTR1 redistributes prefixes from the OSPF network as database-mappings that are then registered with the Mapping System. The figure Redistribute RIB Routes into Database Topology shows the devices for configuring the LISP route import feature for the database application.

Figure 27 : Redistribute RIB Routes into Database Topology



xTR1 Configuration for RIB Route Redistribution into Database

The following example shows how to configure the xTR1 in the sample topology for the LISP route redistribution database application:

```
switch# configure terminal
switch(config)# vrf context VRF1
switch(config-vrf)# ip lisp itr
switch(config-vrf)# ip lisp etr
switch(config-vrf)# instance-id 100
```

```
switch(config-vrf)# ip lisp locator-vrf default
switch(config-vrf)# ip lisp etr map-server 10.10.10.10 key lisp
switch(config-vrf)# ip lisp itr map-resolver 10.10.10.10

! Register database mappings imported from OSPF with the MS/MR with these

locators! switch(config-vrf)# lisp locator-set RLOCSET
switch(config-vrf-lisp)# 10.10.10.1 priority 1 weight 100

! Import fabric prefixes into LISP database based from OSPF!
switch(config-vrf)# ip lisp route-import database ospf 100 route-map
RM_OSPF_to_LISP locator-set RLOCSET
```

Verifying Database-mappings with RIB Route Redistribution

The following example displays the sample show command outputs for the database-mapping configurations: Verify that the OSPF has programmed the target prefix in the routing table.

```
xTR1# show ip route 192.168.1.1 vrf VRF1

IP Route Table for VRF
"VRF1" '*' denotes best
ucast next-hop '**'
denotes best mcast next-
hop


'[x/y]' denotes
[preference/metric]
192.168.1.0/24,
ubest/mbest: 1/0
*via 172.27.1.1, Eth1/27, [110/10], 05:10:39,

ospf-100, intra
```

Verify that LISP has included the prefix in the database-mapping table.

```
xTR1# show ip lisp database vrf VRF1

LISP ETR IP Mapping Database for VRF "VRF1" (iid 100), global LSBs:
0x000000ff Local Database: 12
EID-prefix: 192.168.1.0/24, instance-id: 5001, LSBs:
0x000000ff  Producer:  route_import  ,  locator_set:
RLOCSET,   uptime:   00:19:48   Locator:   10.10.10.1,
priority: 1, weight: 100
```

Verify that the database-mapping is registered with the mapping system.

```
MS# show lisp site

LISP Site Registration Information
* = Some locators are down or unreachable
# = Some registrations are sourced by reliable transport
Site Name      Last      Up     Who Last             Inst     EID Prefix Register          Reg
SITE_ALL_v4    never     no     --                   100     0.0.0.0/0 00:00:05  yes       10.
```

Verify the EID instance address family configurations for the route redistribution database application.

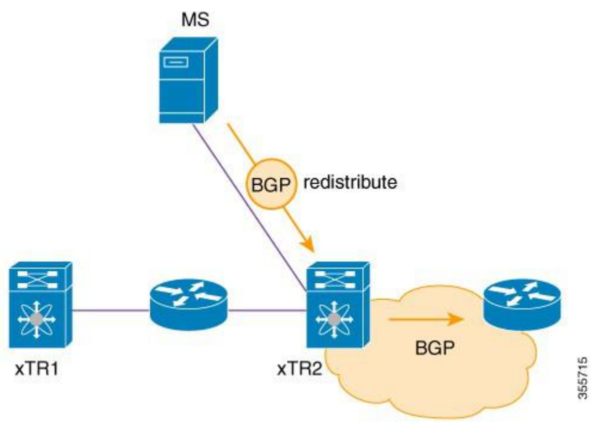
```
switch# show ip lisp route-import vrf VRF1 database

IP LISP Route Import for

 VRF "VRF1" DATABASE

   Specificatons     : 1

   Maximum
   Import            : 1000 Threshold pct      : 75%
```

```
Warn only         : F
Withdraw        : F

Routes
Imported :
0
Rejected
by limit :
0 Warned          : 0

  protocol        : ospf-100
  policy          : RM_OSPF_to_LISP
  bind_pending    : F
   type            : Route
   Import Policy locator_set :
   RLOCSET
```

Configuring RIB Route Redistribution with the Map-cache Application

This section describes the map-cache application configurations and commands to verify the configurations. In this section, the Map Server redistributes the registration table to BGP that propagates the prefixes as routes to xTR2, and finally to external networks. On xTR2, prefixes coming from the Map Server are imported into LISP as map-caches that can be resolved using LISP to optimize the path to destination device. The figure *Redistribute RIB Routes into Map-cache Topology* shows the devices configured for the LISP RIB route redistribution for the map-cache application.

Figure 28 : Redistribute RIB Routes into Map-cache Topology



xTR2 Configuration for RIB Route Redistribution into Map-cache

The following example shows how to configure the xTR2 in the sample topology for the LISP route map-cache application:

```
switch# configure terminal
switch(config)# vrf context
VRF1 switch(config-vrf)# ip
lisp itr switch(config-
vrf)# ip lisp etr
switch(config-vrf)#
instance-id 100
switch(config-vrf)# ip lisp locator-vrf default
switch(config-vrf)# ip lisp etr map-server 10.10.10.10 key lisp
switch(config-vrf)# ip lisp itr map-resolver 10.10.10.10

! Import fabric prefixes into lisp map-cache from BGP!

switch(config-vrf)# ip lisp route-import map-cache bgp 65536 route-map
RM_BGP_to_LISP
```

Verifying Map-cache with RIB Route Redistribution

The following example displays the sample show command outputs for the map-cache route import configurations:

```
xTR2# show ip lisp map-cache vrf VRF1

LISP IP Mapping Cache for VRF ”VRF1" (iid 100), 1 entries
* = Locator data counters are cumulative across all EID-
prefixes 192.168.1.0/24, uptime: 00:09:42, expires: 0.000000,
via route-import Producer Set: 0004 route_import
Negative cache entry, action: send-map-request
```

Once a prefix is imported as a map-cache, the routing table shows how LISP takes over the prefix to ensure an optimized path through the LISP overlay to the destination device.

```
xTR2# show ip route 192.168.2.2 vrf VRF1

IP Route Table for
VRF ”VRF1" '*' denotes
best ucast next-hop




'**' denotes best mcast
next-hop '[x/y]' denotes
[preference/metric]
'%<string>' in via output denotes VRF <string>

192.168.1.1/24, ubest/mbest: 2/0 time
   **via Null0, [10/1], 02:08:42, lisp, eid
    via  10.10.10.3%default,  [200/0],  01:06:55,  bgp-65536,  internal,  tag  65536
    (mpls-vpn)
```

Verify the EID instance address family configurations for the route-import map-cache application.

```
switch# show ip lisp route-import vrf VRF1 map-cache

IP LISP Route Import

 for VRF “VRF1" MAP-

 CACHE

   Specificatons     : 1

   Maximum
   Import            : 1000 Threshold pct      : 75%
   Warn only         : F
   Withdraw        : F

   Routes
   Imported :
   0
   Rejected
   by limit :
   0 Warned          : 0

     protocol        : bgp-65536
     policy          : RM_BGP_to_LISP
     bind_pending    : F
      type            : Route Import Policy
```

CHAPTER 9 Configuration Limits for LISP

This chapter contains the following sections:
·Configuration Limits for LISP.

9.1 Configuration Limits for LISP

The configuration limits are documented in the *Inspur CN12700 Series INOS Verified Scalability Guide*.