

The Network Times
Handbook Series – Part II

LISP with VXLAN in Campus Fabric

A Practical Guide to Understand the Operation of Campus Fabric



Campus Fabric is usually provisioned and managed via SDN Controller. This book explains how the Campus Fabric with LISP Control-Plane and VXLAN Data-Plane works behind the scenes.

Toni Pasanen
CCIE#28158

LISP Control-Plane in Campus Fabric

A Practical Guide to Understand the Operation of Campus Fabric

Toni Pasanen, CCIE 28158

Copyright © Toni Pasanen, All rights reserved.

First edition - 17 February 2020

About the Author:

Toni Pasanen. CCIE No. 28158 (RS), Presales Engineer at Fujitsu Finland. Toni started his IT-career in 1998 at Tieto, where he worked as a Service Desk Specialist moving via the LAN team to the DataCenter team as a 3rd. Level Network Specialist. Toni joined Teleware (Cisco Learning partner) in 2004, where he spent two years teaching network technologies focusing on routing/switching and MPLS technologies. Toni joined Tieto again in 2006, where he spent the next six years as a Network Architect before joining Fujitsu. In his current role, Toni works closely with customers helping them in selecting the right network solutions not only from the technology perspective but also from the business perspective. He has also written the book *“Virtual Extensible LAN – VXLAN: The Practical Guide to Understand VXLAN Solution”*

Who should read the book

Before you implement Campus Fabric, and you most probably will do that by using Cisco DNA Center, you should understand what happens behind the scene. This book tries to give a clear picture of how LISP Control-Plane used in Campus Fabric network solution works. If you are considering implementing Campus Fabric, or if you are already done that, you might find this book valuable. Also, if you are preparing to network certification that includes LISP you might want to read this book.

Table of Contents

Chapter 1: LISP Introduction 1

EID-to-RLOC Registration Process 2

EID-to-RLOC Requesting Process 2

References 4

Chapter 2: Campus Fabric Configuration 5

Underlay Network Configuration - OSPF 5

OSPF optimization: prefix-suppression 6

LISP Ingress and Egress Tunnel Router Configuration 9

LISP Map-Server and Map-Resolver Configurations 10

LISP Customer Network Configuration 12

Customer Layer-2 Segment with SVI attached to VRF 12

Customer specific LISP Configuration on XTR 13

Customer specific LISP Configuration on Control Plane Node 15

References 16

Chapter 3: Control-Plane: EID-to-RLOC Registration 17

Registration Phase 1: TCP Probe 17

Registration Phase 2: UDP based Map-Register from ETR to MS 18

Registration Phase 3: UDP based Map-Notify from MS to ETR 22

Registration Phase 4: TCP based Registration-Refresh from MS to ETR 26

Registration Phase 5: TCP based Map-Register from ETR to MS 29

Registration Phase 6: TCP based Registration ACK from MS to ETR 33

Registration Phase 7: Reliable TCP ACK from ETR to MS 36

Registration Phases 1-6 Summary 38

EID-to-RLOC Registration Verification 39

References 43

Chapter 4: Control-Plane: EID-to-RLOC Request 45

Request Phase 1: Local Proxy-ARP by ITR 45

Request Phase 2: Encapsulated Control Message (ECM) by ITR 46

Request Phase 3: Map-Reply by MR 50

Request Phase 4: Map-Request probe by ITR 53

Request Phase 5: Map-Reply Probe by RLOC 57

Map-Request Process Summary: Phases 1-5 60

References 61

Chapter 5: Data Plane: VXLAN Encapsulation 63

VXLAN Data-Plane operation 63

SGT Information in Data-Plane 68

References 69

Chapter 6: LISP Control-Plane Security 70

The Basic LISP Control-Plane Security 70

Map-Register validation process 71
Map-Reply validation process 74
LISP-SEC Control-Plane Security 76
ITR Map-Request Process 76
Map-Server Map-Reply process 80
ITR Authentication Process 84
Configuration 85
References 86

Chapter 7: Campus Fabric External Connections 87

Introduction 87
EID-Prefix advertisement from Campus Fabric to non-LISP Site 87
IP-Prefix advertisement from non-LISP Site to Campus Fabric. 89
Data-Plane operation 90
Configuration 96
BGP Peering Configuration 96
Control Plane Node Configuration 97
Border-PxTR Configuration (egress direction) 98
Border-PxTR Configuration (ingress direction) 99
Control-Plane Operation: Campus Fabric to non-LISP site 100
Control-Plane Operation: non-LISP site to Campus Fabric 105
Control and Data-Plane Operation: Internet 112

Chapter 8: Campus Fabric Inter-VN Connections 115

Introduction 115
Inter-VN EID-Prefix advertisement model 115
Configuration 117
Control Plane Node configuration: VRF 100_White 118
Control Plane Node configuration: VRF 200_Grey 120
PxTR configuration: VRF 100_Grey and VRF 200_Grey 121
C-P Egress Operation: LISP Registration 123
C-P Egress Operation: BGP-Updates: PxTR 127
C-P Egress Operation: BGP-Updates: non-LISP router 128
C-P Ingress Operation: BGP-Updates: PxTR 128
C-P Ingress Operation: BGP-Updates: Border-PxTR 130
Data-Plane Operation 132
Summary 137
References 138

Chapter 9: Campus Fabric Redundancy 139

Introduction 139
CP Node Redundancy: Anycast-IP RLOC 140
CP Node Redundancy: Core switch out of service 143
CP-Node Redundancy: Primary CP Node Out of Service 145
PxTR Redundancy: Priority 150
PxTR Redundancy: Priority Configuration 151
PxTR Redundancy: Priority Verification 152
PxTR Redundancy: PxTR Switch-over - Max. Priority 156

Chapter 10: Inter-Site and Internet connections 160

LISP Registration Process by iTR 161
LISP-to-BGP redistribution by Map-Server 163
BGP Update from PxTR to the non-LISP router 165
BGP-to-LISP redistribution by Transit-PxTR23 166
LISP Map-Register from Transit-PxTR23 to Map-Server CP21. 168
Map-Request and Map-Reply process 169
VXLAN encapsulation 170
Non-LIPS router routing process 171
Map-Request caused by incoming IP packet from non-LISP site 172
Verification 173
Internet Connection: EID to non-LISP Router 174
Internet Connection: non-LISP Router network to LISP Site 175
Internet Connection verification 177
References 178

Chapter 11: The Underlay Network-SPT Algorithm 179

Introduction 179
OSPF 180
Link-State Database (LSDB) optimization 181
Shortest-Path First (SPF)/Dijkstra Algorithm 187
SPF Run – Phase I: Building a Shortest-Path Tree 188
First iteration round 189
Second iteration round 190
Third iteration round 191
Fourth iteration round 192
Fifth iteration round 193
Sixth iteration round 194
Seventh iteration round 195
SPF Run – Phase II: Adding Leafs to Shortest-Path Tree 196
References: 198

Chapter 12: Underlay Network – Comparison of OSPF and IS-IS 199

Scenario-1: Interface loopback 50 down on Leaf-101 (IS-IS) 202
Scenario-2: Interface loopback 50 down on Leaf-101 (OSPF) 204
Scenario-3: OSPF Incremental SPF – L50 Down on Leaf-101 (Stub) 206
Scenario-4: OSPF Incremental SPF – Interface g0/3 Down on Spine-12 (transit link does not participate in SPT) 207
Scenario-5: IS-IS SPF – Interface g0/3 Down on Spine-12 (Full SPF computation) 208
Scenario-6: IS-IS Incremental SPF – Interface g0/3 Down on Spine-12 (transit link does not participate in SPT) 209
Conclusion 210
References: 211

Appendix A. Chapter 4 Device Configurations 212

- A.1 Edge-1 Configuration 212*
- A.2 Edge-2 Configuration 216*
- A.3 ControlPlane Configuration 219*
- A.4 ISP-Router Configuration 220*

Appendix B. Chapter 7-8 Device Configurations 221

- B.1 Border-PxTR Configurations 221*
- B.2 Control Plane Configurations 224*
- B.3 Edge-1 Configurations 227*
- B.4 Edge-2 Configurations 230*

Appendix C. Chapter 10 Device Configurations 236

- C.1 xTR14 Configurations 236*
- C.2 xTR24 Configurations 238*
- C.3 Co15 Configurations 240*
- C.4 Co25 Configurations 240*
- C.5 CP11 Configurations 241*
- C.6 CP21 Configurations 243*
- C.7 Transit-PxTR13 Configurations 244*
- C.8 Transit-PxTR23 Configurations 246*
- C.9 Inet-PxTR12 Configurations 248*
- C.10 Inet-PxTR22 Configurations 250*
- C.11 ISP-201 Configurations 252*
- C.12 ISP-201 Configurations 252*

About this book

Traditional Campus Networks are using some version of the Spanning-Tree Protocol (STP) as an L2 Control-Plane protocol. STP builds a loop-free switching path into a switching environment. However, unlike L3 Control Plane protocols, such as OSPF and BGP, the STP does not give any information about the location of hosts. The host location learning process is done by using a “flood and learn” mechanism, where switches learn the location of hosts from the source mac-address field in ingress Ethernet frames. Redundant paths are done by bundling two-to-eight physical Inter-Switch links into a logical Port-Channel. Gateway redundancy either relies on some First-Hop Redundancy Protocol (VRRP, HSRP or GLBP) or virtual switching system like Cisco VSS - Virtual Switch-System.

Several years ago, Datacenter networks were built in the same way as traditional Campus Networks. Nowadays the modern DC networks are based on an Underlay and Overlay network architecture. The Underlay network is used as a transport network between network devices while the Overlay network is used tenant-based L3/L2 virtualization. The Control-Plane protocol for Underlay Network is usually OSPF, IS-to-IS or BGP while the Overlay network Control Plane protocol is usually *BGP L2VPN EVPN*, which advertises mac-address among the other NLRIs. In Data-Plane, Ethernet Frames are encapsulated inside VXLAN headers (Outer-mac, outer-IP, UDP and VXLAN headers) when sent to a host located behind a remote VTEP switch. Redundant paths rely on Equal Cost Multi-Path (ECMP) while Gateway redundancy is implemented in distributed Anycast-Gateway. A common term used for this solution is *BGP BGP EVPN VXLAN Fabric*.

In 2018, Cisco launched a new *Cisco Digital Network Architecture (CDNA)* model where a Campus Network solution relies on the same kind of underlay/Overlay network architecture than what has already been using in modern Datacenter networks. While the common Overlay Network Control-Plane protocol choice for DC networks is *BGP L2VPN EVPN*, the CDNA solution uses the *Locator/ID Separation Protocol (LISP)*. Data-Plane encapsulation though uses the VXLAN encapsulation (with a VXLAN GPE extension). This network solution is commonly called *Campus Fabric/SD-Access*. The Campus Fabric network can be provisioned by using *DNA Center (DNAC)*. DNAC generates both the Control-Plane and Data-Plane configurations as well as IP addressing using the administrator-defined network addressing scheme. This means that to do the job, you do not necessarily have to fully understand how the solution works.

The purpose of this book is to explain how the LISP Control-Plane with VXLAN Data Plane works. All examples are done using EVE-NG with IOS-XE 16.06.07 Everest (CSR1000v). The reason for running CSR1000v instead of actual Catalyst 9300/9500 switches is that this way readers can build their own practices labs without having physical switches. Note that CSR1000v does not support Ethernet service or L2 VLAN access-port. This is the reason why Bridge-Domains are used instead of VLAN.

Disclaimers

The content of this book is based on the author's own experience and testing results. This book is meant to be neither design nor an implementation guide. After reading this book, readers should do their technology validation before using it in a production environment.

Chapter 1: LISP Introduction

The endpoint identity is usually based on its IP address while the subnet where the host IP address belongs to defines its location. This means that whenever the host moves from one segment to another, its identity changes. From the host perspective, changing the IP address affects how its data is treated in a network. First, policy rules on traffic filters are usually based on the IP address information. In order to get the same treatment, there have to be equal policy rules throughout the network. Second, traffic classification and prioritization (QoS) is also based on IP addresses or subnets. The IP address in this sense defines both the identity and location of an endpoint as well as defines how the data is treated. *Locator/ID Separation Protocol (LISP)* disjoints the location from the host IP address. Instead of using the IP address of a host as a locator, LISP uses Routing Locator (RLOC), which basically is the IP address of the router where the host is connected to.

The focus of this book is to describe the operation of a Campus Fabric with LISP Control-Plane. Figure 1-1 illustrates the basic LISP operation and its building block by explaining processes that are required in order to establish an IP connectivity between Host-A (172.16.100.101) and Host-B (172.16.100.102). This process is divided into two sections; the EID registration process and the EID-to-RLOC mapping request process.

EID-to-RLOC Registration Process

- Phase 1.** Host-B joins the network. It sends a Gratuitous ARP to verify that its IP address is unique and to inform the network devices that it has just joined the network.
- Phase 2.** The router E2-xTR learns the MAC and the IP address of Host B from the GARP message. The role of E2-xTR in the LISP process is *Egress Tunnel Router (ETR)*. This means that it is responsible for registering (publishing) the Host-B Endpoint Identifier (EID) and Routing Location (RLOC) of the last hop egress router (E2-xTR itself) to LISP Map-Server (MS). E2-xTR sends a Map-Register message to MS where it describes the EID-to-RLOC mapping. E2-xTR uses its loopback IP address 192.168.2.2 as RLOC.
- Phase 3.** When Map-Server receives the update, it stores the information into the EID-to-RLOC mapping database. This database is a centralized database that holds all the EID-to-RLOC information of the LISP domain.

EID-to-RLOC Requesting Process

- Phase A.** Host-A starts communication with Host-B. Because the destination IP address is in the same subnet than Host-A, it tries to resolve the MAC-IP address binding information by sending an ARP-Request message. The router E1-xTR receives the message and answers with ARP-Reply using its own MAC-address. The role of the E1-xTR in this example is the *Ingress Tunnel Router (ITR)*. Instead of forwarding ARP-request, E1-xTR as local Proxy-ARP replies to ARP using its own shared MAC-address.
- Phase B.** The ITR E1-xTR sends a Map-Request message to the Map-Resolver to find out the RLOC information of Host-B.
- Phase C.** The Map-Server consults the EID-to-RLOC database published by Map-Server. Note that both Map-Server and Map-Resolver co-exist in the ControlPlane node and they together form the *LISP Mapping System*.
- Phase D.** The Map-Resolver replies with Map-Reply message where it describes the EID-to-RLOC mapping information of Host-B (EID: 172.16.100.102/32 with RLOC 192.168.2.2).

Phase E. E1-xTR receives the Map-Reply from Map-Resolver. It stores the EID-to-RLOC information into *LISP Mapping Cache*. Now it knows that Host-B is located behind the IP address 192.168.2.2. It checks the next-hop-router for the destination 192.168.2.2 from its RIB. It then encapsulates the received, original packet sent by Host-A with a VXLAN tunnel header, where the destination IP address is 192.168.2.2. and the source IP address is 192.168.1.1 and forwards the encapsulated packet out of its interface Gi1.

The same operation happens in reversed order when Host-B replies to Host-A. In that sense, E1-xTR and E2-xTR are both ingress and egress tunnel routers (xTRs).

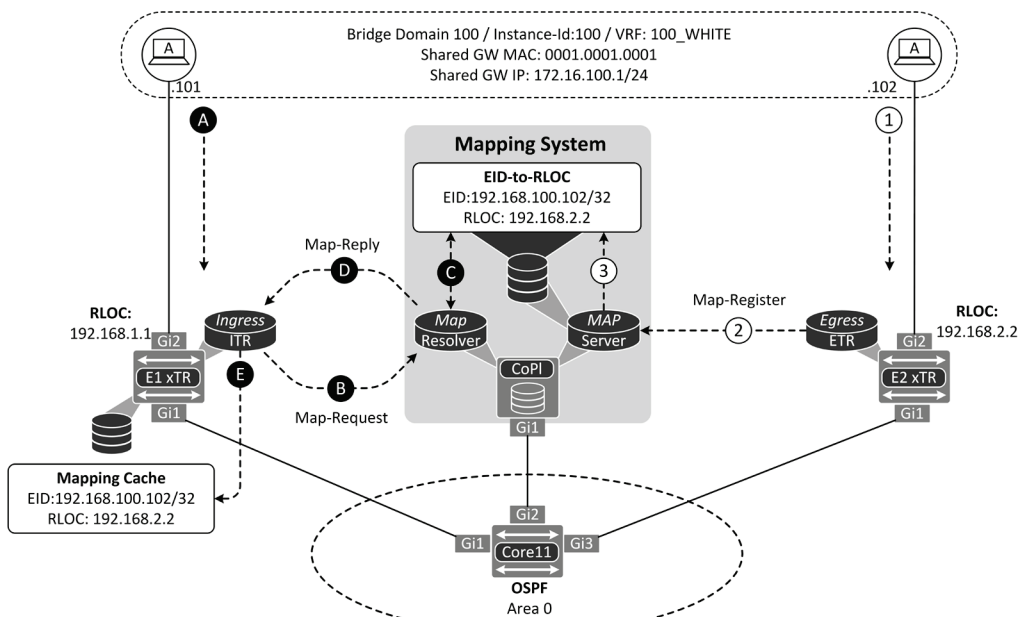


Figure 1-1: Map-Register and Map-Request process summary.

Note that the SD-Access (SDA) term for LISP ingress/egress Tunnel Routers (xTR) is an Edge-switch. In addition, in the SDA solution, both Map-Server and Map-Resolver co-exist in the same device called as Control Plane Node in SDA.

References

- [RFC 6830] D. Farinacci et al., "The Locator/ID Separation Protocol (LISP)", RFC 6830, January 2013.
- [RFC 6833] V. Fuller and D. Farinacci., "Locator/ID Separation Protocol (LISP) Map-Server Interface", RFC 6833, January 2013.
- [VXLAN-GPE] F. Maino et al., "Generic Protocol Extension for VXLAN", draft-ietf-nvo3-vxlan-gpe-09, December 2019.
- [LISP Control Plane] D. Farinacci et al., "The Locator/ID Separation Protocol (LISP) Control Plane", draft-ietf-lisp-rdc6833bis-25, June 16, 2019.
- [VXLAN-GPE] J. Lemon et al., "Croup Policy Encoding with VXLAN GPE", draft-ietf-vxlan-gpe-gbp-02, March 2, 2019.
- [LISP Rel-Tran] C. Cassar et al., "LISP Reliable Transport", draft-kouvelas-lisp-reliable-transport-02, March 26, 2015.

The LISP Network: Evolution of the Next Generation of Data Networks
ISBN-10: 1-58714-471-9 – Victor Moreno, and Dino Farinacci

LISP Network Deployment and Troubleshooting: The Complete Guide to LISP
Implementation on IOS-XE, IOS-XR, and NX-OS
ISBN-10: 1-58714-506-5 – Tarique Shakil, Vinit Jain, and Yves Louis

Chapter 2: Campus Fabric Configuration

The functionality of a LISP system is easier to explain and understand by using examples. This is why the configuration is explained before the theory part.

The configuration of Campus Fabric from scratch can roughly be divided into three phases; the Underlay Network configuration, the LISP Overlay Network configuration and customer network configuration. This section introduces how these phases are done using IOS-XE Command Line Interface (CLI). Also, this section shows some basic verification commands after each phase. The complete device configuration can be found in Appendix A. Chapter 11 discusses the operation of the Shortest Path First/Dijkstra algorithm in detail. Chapter 12 explains compares OSPF and IS-IS.

Underlay Network Configuration - OSPF

The common choice for routing protocol used in Underlay Network is either IS-IS or OSPF. Both of these protocols are Link-State protocol and they use Shortest-Path First (SPF) algorithm (also called Dijkstra) to find out the shortest path to each destination. There is no significant difference between these two routing protocols from the Campus Fabric Underlay Network perspective and either one can be used. We are using OSPF in this example. The basic OSPF configuration is straight forward, start the OSPF process and assigns OSPF RID. Then add all Inter-Switch links into the OSPF Area 0. In addition, advertise Interface Loopback 0 (used as an RLOC). To avoid unnecessary DR/BDR election in link the OSPF Network-Type for each OSPF interface is set to point-to-point

```
router ospf 1
  router-id 192.168.1.1
!
interface Loopback0
  ip address 192.168.1.1 255.255.255.255
  ip ospf network point-to-point
  ip ospf 1 area 0
!
interface GigabitEthernet1
  ip address 10.1.11.1 255.255.255.0
  ip lisp source-locator Loopback0
  ip ospf network point-to-point
  ip ospf 1 area 0
```

Example 2-1: *Campus Fabric Underlay Network Configuration on E1-xTR1.*

OSPF optimization: prefix-suppression

The default OSPF network type in the Ethernet link is Broadcast. This means that routers connected to link select the Designated Router (DR) and Backup-designated (BDR) router among themselves. Non-Designated routers send Router LSAs only to DR, which in turn floods them to other routers in the segment as a Network LSA (Type-2).

OSPF speakers advertise Type-1 Router LSAs which are used for Shortest-Path calculation. In addition, OSPF speakers advertise the Type-3 Router LSA for describing each subnet they are connected to. This includes the subnet used in Inter-Switch links which are considered as a Transit Networks, and are not used for data traffic destination. This means that they consume unnecessary hardware resources and affect convergence time. These unnecessary Transit Network LSAs can be hidden by using command **ip ospf prefix-suppression** under each Inter-Switch links. Example 2-2 shows the OSPF LSDB taken from E1xTR (prompt is Edge-1) before OSPF prefix-suppression. Their link count in Area 0 is 16.

Edge-1#sh ip ospf database beg ADV					
Link ID	ADV Router	Age	Seq#	Checksum	Link count
192.168.1.1	192.168.1.1	1163	0x80000002	0x00130F	3
192.168.2.2	192.168.2.2	1165	0x80000003	0x0054C3	3
192.168.11.11	192.168.11.11	1159	0x80000006	0x00DDCA	7
192.168.111.111	192.168.111.111	1159	0x80000003	0x00C07E	3

Example 2-2: “sh ip ospf database | beg ADV” before suppression.

Example 2-3 illustrates the OSPF LSDB taken from E1xTR after OSPF prefix-suppression. Their link count is reduced from 16 to 10.

Edge-1#sh ip ospf database beg ADV					
Link ID	ADV Router	Age	Seq#	Checksum	Link count
192.168.1.1	192.168.1.1	118	0x80000003	0x008FB8	2
192.168.2.2	192.168.2.2	110	0x80000004	0x00B08E	2
192.168.11.11	192.168.11.11	87	0x80000009	0x006129	4
192.168.111.111	192.168.111.111	102	0x80000004	0x000BC8	2

Example 2-3: “sh ip ospf database | beg ADV” after suppression.

Example 2-4 shows the routing table of E1xTR before OSPF prefix-suppression. Each Transit network is learned via OSPF.

```
Edge-1#sh ip route ospf | beg Gate
Gateway of last resort is not set

    10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
O       10.2.11.0/24 [110/2] via 10.1.11.11, 00:22:16, GigabitEthernet1
O       10.11.111.0/24 [110/2] via 10.1.11.11, 00:22:16, GigabitEthernet1

    192.168.2.0/32 is subnetted, 1 subnets
O       192.168.2.2 [110/3] via 10.1.11.11, 00:22:16, GigabitEthernet1
    192.168.11.0/32 is subnetted, 1 subnets
O       192.168.11.11 [110/2] via 10.1.11.11, 00:22:16, GigabitEthernet1
    192.168.111.0/32 is subnetted, 1 subnets
O       192.168.111.111 [110/3] via 10.1.11.11, 00:22:06, GigabitEthernet1
```

Example 2-4: “*sh ip route ospf | beg Gate*” before OSPF prefix-suppression.

Example 2-5 shows the routing table of E1xTR after prefix-suppression. There are no Transit networks anymore learned from OSPF peers.

```
Edge-1#sh ip route ospf | beg Gate
Gateway of last resort is not set

    192.168.2.0/32 is subnetted, 1 subnets
O       192.168.2.2 [110/3] via 10.1.11.11, 00:27:20, GigabitEthernet1
    192.168.11.0/32 is subnetted, 1 subnets
O       192.168.11.11 [110/2] via 10.1.11.11, 00:27:20, GigabitEthernet1
    192.168.111.0/32 is subnetted, 1 subnets
O       192.168.111.111 [110/3] via 10.1.11.11, 00:27:10, GigabitEthernet1
```

Example 2-5: “*sh ip route ospf | beg Gate*” after OSPF prefix-suppression.

Figure 2-1 summarizes the OSPF configuration. It also illustrates the IP addressing scheme.

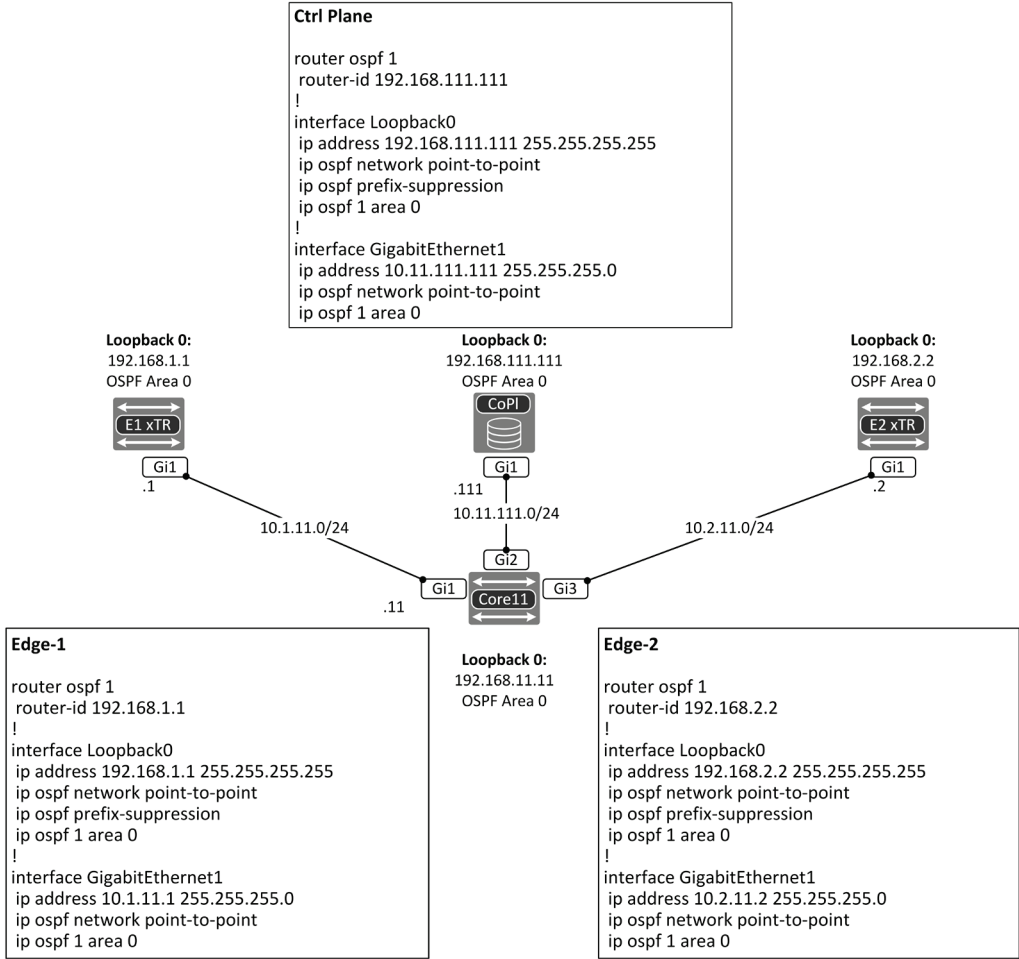


Figure 2-1: *The Underlay Network Configurations.*

LISP Ingress and Egress Tunnel Router Configuration

The command **router lisp** starts the LISP process. When sending EID-to-RLOC messages to LISP Mapping-Server, ETR uses the RLOC address as a source address for the message. The command **locator-table default** command specifies that the RLOC address used as an RLOC belongs to the default (global) routing table. This model, where multiple instances are mapped into different VRFs using the same RLOC address, is called the *Shared Model*. The command **locator-set RLOC-SET1** sets the name for the locator-set that defines the actual source IP address used in LISP Map-Register messages. Priority and weight values are used when there is two or more ETR advertising the same EIDs. These values are sent within a LISP Map-Register message to Mapping-Server by ETR (priority is explained in chapter 9). Mapping-Server then uses these values to select preferred RLOC. The lowest priority value is preferred. If priorities are equal, then traffic is load-balanced based on weight values. There is only one ETR in this example and these values are not effective. The command **locator default-set RLOC-SET1** sets the RLOC-SET1 as default.

The command **service ipv4** is the service that uses an IP address of an endpoint as an EID when sending LISP Map-Register messages. The commands **itr** and **etr** set both roles ITR and ETR to a device. The command **encapsulation vxlan** specifies the Data-Plane encapsulation type. The IP address 192.168.111.111 belongs to the Control-Plane node (Loopback 0) and it is configured as both Map-Resolver and Map-Server. The option **key nwktimes** is a shared-secret for LISP message authentication between xTRs and Map-Server. The command **proxy-reply** option sets the *Proxy Map-Reply* flag (P-bit) in the LISP Map-Register message to 1. By doing this, ETR authorizes the Map-Server to answer the LISP Map-Request messages on behalf of ETR itself. The Command **ip lisp source locator Loopback 0** sets the Loopback 0 IP address as a source for all LISP messages sent out of the interface Gi1. The command **service ethernet** is the service that uses a MAC address of an endpoint as an EID when sending Map-Register messages. Its configuration is the same as service IPv4. This book uses only IPv4 service due to the lack of Ethernet service support in CSR1000v.

```
router lisp
 locator-table default
 locator-set RLOC-SET1
  IPv4-interface Loopback0 priority 1 weight 1
 exit-locator-set
 !
 locator default-set RLOC-SET1
 service ipv4
  encapsulation vxlan
  itr map-resolver 192.168.111.111
  itr
  etr map-server 192.168.111.111 key nwktimes
  etr map-server 192.168.111.111 proxy-reply
  etr
 exit-service-ipv4
 !
 service ethernet
  itr map-resolver 192.168.111.111
  itr
  etr map-server 192.168.111.111 key nwktimes
  etr map-server 192.168.111.111 proxy-reply
  etr
 exit-service-ethernet
 !
 interface GigabitEthernet1
  ip lisp source-locator Loopback0
```

Example 2-6: Tunnel Router E1xTR (Edge-1) LISP configurations.

LISP Map-Server and Map-Resolver Configurations

Example 2-7 illustrates the Map-Server/-Resolver LISP configuration. The **locator-table default** command specifies that the IP address is used as an RLOC belongs to the default (global) routing table. The command **encapsulation vxlan** defines the Data-Plane encapsulation type. The command **map-server** and **map-resolver** under the IP and Ethernet services defines that the Control-Plane node is both Map-Server and Map-responder. The Command **ip lisp source locator-loopback 0** sets the Loopback 0 IP address as a source for all LISP messages sent out of the interface Gi1.

```

router lisp
 locator-table default
 service ipv4
   encapsulation vxlan
   map-server
   map-resolver
   exit-service-ipv4
 !
 service ethernet
   map-server
   map-resolver
   exit-service-ethernet
 !
 interface GigabitEthernet1
   ip lisp source-locator Loopback0

```

Example 2-7: Control Plane Node LISP configurations.

Figure 2-2 summarizes the common LISP configuration. The LISP configuration is the same in both xTRs Edge-1 and Edge-2, and that is why the configuration of Edge-2 is excluded from the picture. At this stage, the example network is ready for client networks to be implemented.

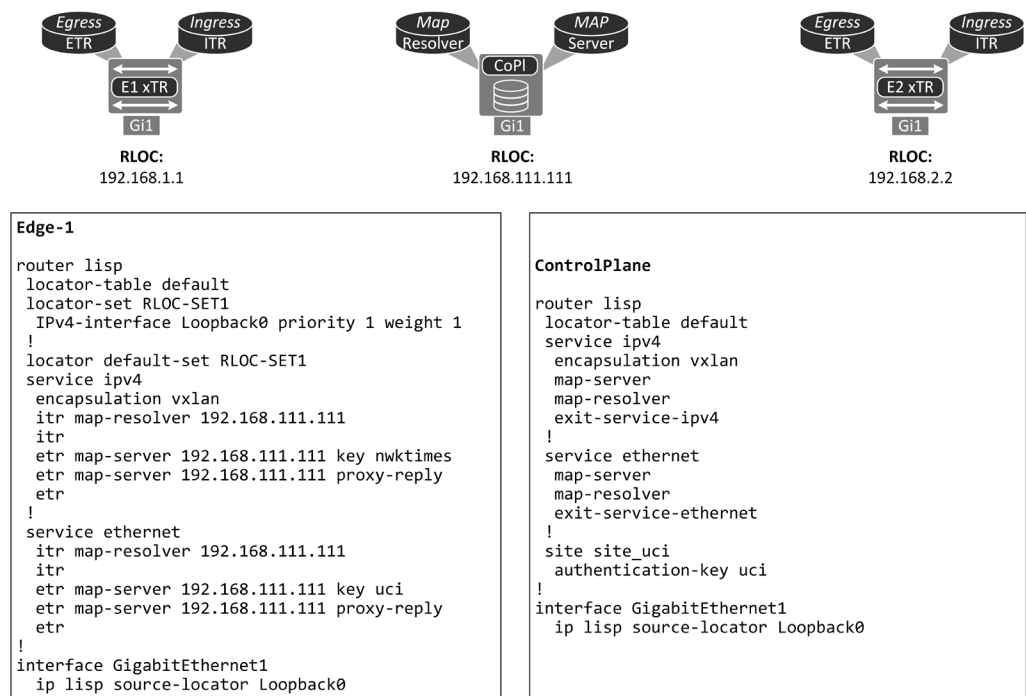


Figure 2-2: The Common LISP Configurations.

LISP Customer Network Configuration

Customer Layer-2 Segment with SVI attached to VRF

The command **vrf definition 100_White** creates the Virtual Routing Instance named 100_White. Route-Distinguisher 1:100 and And Route-Target 1:00 are assigned to it.

Note! The example network topology is built by using CSR1000v running IOS-XE Everest 16.06.07 code. This is because Nexus 9000v does not support LISP and there is no virtual image for Catalyst 9300/9500 switch available at the time of writing. CSR1000v does not support L2VLAN or L3VLAN interface configuration. However, it supports the Bridge Domain with SVI configuration, which is used in these examples.

The interface Gi2 is Host-101 facing interface. The command **service instance 100 ethernet** basically defines that this port offers an ethernet service to a connected host. The command **bridge-domain 100** defines the L2 identifier for the broadcast domain in the same way that the command **switch port access vlan 100** in traditional VLAN setup.

The command **interface BDI100** enables the Layer 3 routing interface for L2 BD100 just like the command **Interface vlan 100** in traditional VLAN based service. BDI100 is attached to vrf 100_White by using the command **vrf forwarding 100_White**. The L3 interface has the shared Anycast Mac-address 0001.0001.0001 and shared Anycast IP address 172.16.100.1/24. By using the BDI Interface sub-command **ip local proxy-arp** the router is authorized to respond to ARP-requests for addresses on the same subnets. As an example, when Host-A (172.16.100.101) tries to resolve the MAC address of Host-B (172.16.100.102), it sends an ARP-Request. When E1xTR receives the ARP-request, it sends an ARP-Reply message using its own mac-address 0001.0001.0001. This way there is no need to forward L2 BUM traffic over the Campus Fabric. The command **ip route-cache same-interface** enables fast-switch on an interface where egress and ingress interface are the same.

```
Edge-1(config)#vrf definition 100_White
Edge-1(config-vrf)# rd 1:100
Edge-1(config-vrf)# address-family ipv4
Edge-1(config-vrf-af)# route-target export 1:100
Edge-1(config-vrf-af)# route-target import 1:100
!
Edge-1(config)#interface GigabitEthernet2
Edge-1(config-if)# service instance 100 ethernet
Edge-1(config-if-srv)# encapsulation untagged
Edge-1(config-if-srv)# bridge-domain 100

Edge-1(config)#interface BDI100
Edge-1(config-if)# mac-address 0001.0001.0001
Edge-1(config-if)# vrf forwarding 100_White
Edge-1(config-if)# ip address 172.16.100.1 255.255.255.0
Edge-1(config-if)# ip local-proxy-arp
Edge-1(config-if)# ip route-cache same-interface
Edge-1(config-if)# no shut
```

Example 2-8: *Customer L2/L3 configuration on E1xTR (Edge-1).*

Customer specific LISP Configuration on XTR

The command **instance-id 100** creates the VPN identifier for VRF 100_White. This information is used as VPN-Id in both Control-Plane and Data-Plane. Instance-Id is sent to the Map-Server (Control Plane Node) within the Map-Register message by an ETR (Edge routers). By using Instance-Id, Map-Server is able to differentiate possible overlapping IP addresses used in different Virtual Networks (VN). The command **remote-rloc-probe on-route-change** triggers a probe to remote RLOC whenever there is a routing change concerning the IP address of RLOC in the routing table (RIB).

The command **dynamic-eid 172_16_100_0-100_White** creates a dynamic EID mapping-table where the network 172.16.100.0/24 is attached with the locator-set RLOC-SET1 by using the command **database-mapping 172.16.100.0/24 locator-set RLOC-SET1**. IPv4 service is attached to the instance by using the command **service ipv4**. This particular dynamic EID mapping table is then added into VRF_White by using the command **eid-table vrf 100_White**. The command **map-cache 0.0.0.0/0 map-request** under the service ipv4 defines that Ingress Tunnel Router (ITR) is allowed to send EID Map-Request to Map-Responder for any EID. The command **lisp mobility 172_16_100_0-100_White** under the Interface BDI100 is the last command for dynamic-EID policy configuration. At this stage when E1xTR (Edge-1) receives the first frame from Host-A towards Host-B, it checks if the source IP address belongs to the network defined under the dynamic-EID table 172_16_100_0-100_White. Because the check is a hit, it sends the Map-Request message to Map-Responder defined in LISP service ipv4 using its own RLOC IP address as a source address.

```
Edge-1(config)#router lisp
Edge-1(config-router-lisp)# instance-id 100
Edge-1(config-lisp-inst)# remote-rloc-probe on-route-change
Edge-1(config-lisp-inst)# dynamic-eid 172_16_100_0-100_White
Edge-1(config-lisp-inst-dyn-eid)# database-mapping 172.16.100.0/24
locator-set RLOC-SET1
Edge-1(config-lisp-inst)# service ipv4
Edge-1(config-lisp-inst-srv-ipv4)# eid-table vrf 100_White
Edge-1(config-lisp-inst-srv-ipv4)# map-cache 0.0.0.0/0 map-request
!
Edge-1(config)#interface BDI100
Edge-1(config-if)# lisp mobility 172_16_100_0-100_White
```

Example 2-9: *Customer Instance and IPv4 service configuration on E1xTR (Edge-1).*

Customer specific LISP Configuration on Control Plane Node

In order to Map-Server can accept LISP Map-Register messages from ETRs, there have to be an instance-specific EID-prefixes under the customer site. The command **site Network-Times** defines the name of the customer site. The command **authentication-key nwktimes** defines the authentication key. The command **eid-record instance-id 100 172.16.100.0/24 accept-more-specifics** defines that an EID registration within the EID-prefix 172.16.100.0/24 can be accepted and installed into the EID-to-RLOC database with Instance-id 100. Note that Instance-Id is included in Map-Register messages sent by ETR.

```
CtrlPlane(config)#router lisp
CtrlPlane(config-router-lisp)# site Network-Times
CtrlPlane(config-router-lisp-site)# authentication-key nwktimes
CtrlPlane(config-router-lisp-site)# eid-record instance-id 100
172.16.100.0/24 accept-more-specifics
```

Example 2-10: Customer Site/Instance configuration on ControlPlane node.

Figure 2-3 summarizes the customer-specific LISP configuration.

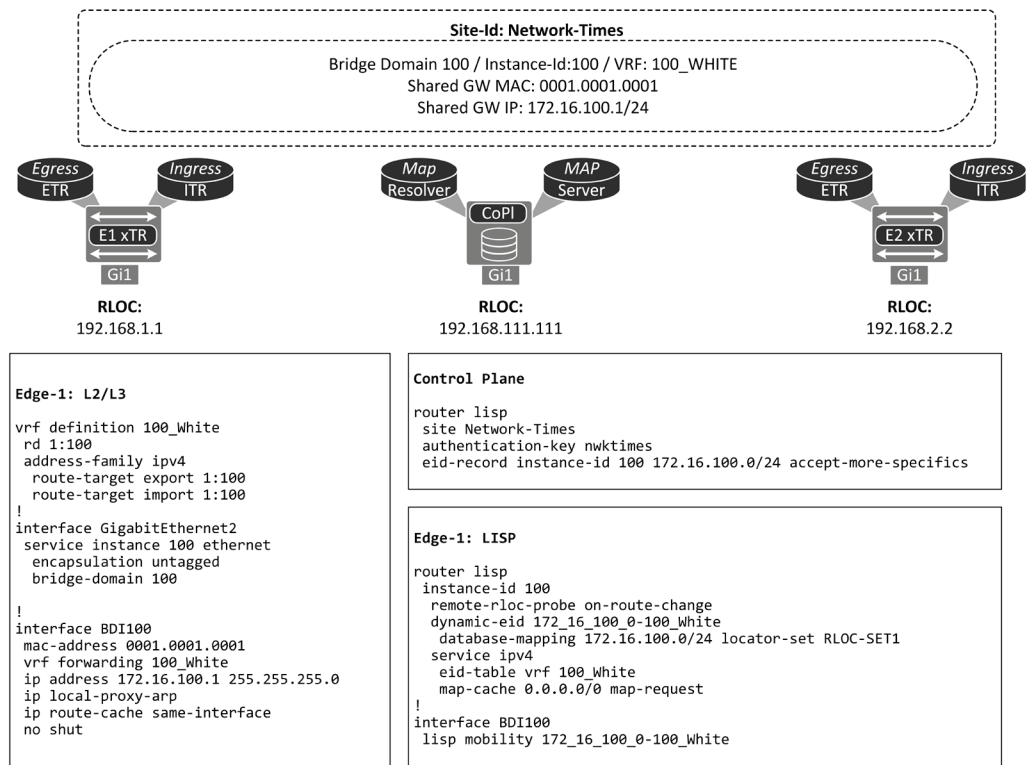


Figure 2-3: Configuring a Customer segment into the Campus Fabric summary.

References

LISP Network Deployment and Troubleshooting: The Complete Guide to LISP Implementation on IOS-XE, IOS-XR, and NX-OS

ISBN-10: 1-58714-506-5 – Tarique Shakil, Vinit Jain, and Yves Louis

https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst9500/software/release/16-6/configuration_guide/cf/b_166_cf_9500_cg/b_166_cf_9500_cg_chapter_00.pdf

Chapter 3: Control-Plane: EID-to-RLOC Registration

The LISP Control-Plane operation can be split into two-part. First, registering EID-to-RLOC information to the Map-Server by Egress Tunnel Routers (ETR). Second, finding the location of an endpoint by requesting EID-to-RLOC information from the Map-Server by Ingress Tunnel Router (ITR).

Registration Phase 1: TCP Probe

In figure 3-1, Host-A joins the network. It generates a Gratuitous-ARP message to verify that its IP address is unique and inform the first-hop-router E1xTR that it has joined to the network. When E1xTR receives the GARP-messages, it starts the EID-to-RLOC registration process. Before sending a Map-Register message, the E1xTR verifies the IP connectivity to Map-Server by sending a TCP-message with an SYN-bit and destination TCP port 4342. As a response to the TCP SYN message, Map-Server replies with TCP-message where both ACK and RST bits are set. The RST-bit set because Edge-1 uses TCP SYN message as a probe, not as a TCP-connection opening request, and there is no need to establish a TCP connection at this phase.

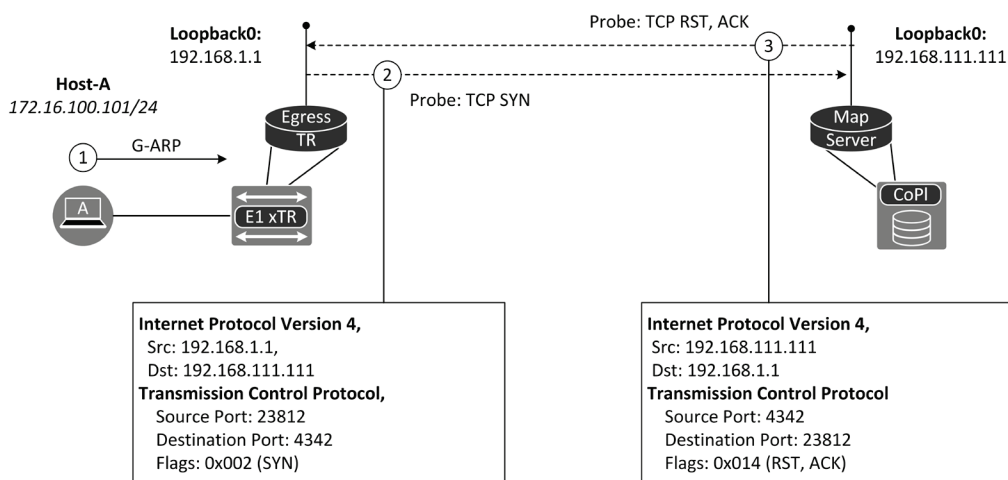


Figure 3-1: EID-to-RLOC Registration - TCP Probe.

Registration Phase 2: UDP based Map-Register from ETR to MS

After verifying the IP connectivity to Map-Server, E1xTR sends a *Map-Register* message to Map-Server. Figure 3-2 illustrates the partial packet captured from the core link of E1xTR, the complete capture is shown in Capture 3-1. The message is sent to Map-Server over an unreliable UDP connection by using the RLOC IP address of E1xTR address as a source address. The type-field value 3 defines that this is a Map-Register message.

The *P-bit (Proxy-Map-reply)*: When set states that a Map-Server can respond on behalf of E1xTR to Map-Request concerning requested EID 172.16.100.101/32.

The *I-Bit (xTR-ID Present)*: When set, it means that this message carries a randomly generated 128-bits XTR-ID identifier of E1xTR and 64-bit site id (not defined).

The *M-bit (Want-Map-Notify)*: When set, tells to receiving Map-Server that it should acknowledge the received Map-Register message by replying with a *Map-Notify* message.

The eight octets “*Nonce*” field is used for protecting against *Map-Register reply attacks*. When E1xTR receives the *Map-Notify* message used as an acknowledgment to Map-Register message, it verifies that the “nonce” value matches to “nonce” value in the original Map-Register message. In addition, E1xTR keeps track of the “nonce” values that it has used, and every time it sends a new Map-Register message, it increments the value. The Map-Server stores the “nonce” value received via the E1xTR Map-Register message into the EID-to-RLOC database along with xTR-ID. When E1xTR, identified with xTR-ID sends the next Map-Register message (or any other LISP messages), the “nonce” value should be bigger than the last received and stored “nonce” value. If this is not true, the Map-Server concerns the message as a “*reply-attacks*” and it drops the Map-Register message.

The Map-Register message describes the EID record and RLOC record. The *EID-record* describes the Instance-Id where the EID 172.16.100.101/32 belongs to and it is used as a VPN identifier. Note that we are not using a VRF aware Map-Server. Time-to-Live value describes how long this EID mapping information is valid. The default value is 1440 seconds (24 hours). An *Authoritative-bit*, when set, defines that sending ETR is the actual last-hop-router for this EID. The *Locator-record* describes the RLOC specific information. It describes the RLOC IP address mapped to EID, priority value used for setting the primary RLOC in a case where there is more than one ETR, and RLOC weight used for load-balancing if multiple RLOC has the same priority value. In addition, the RLOC record carries information about xTR-Id and Site-Id.

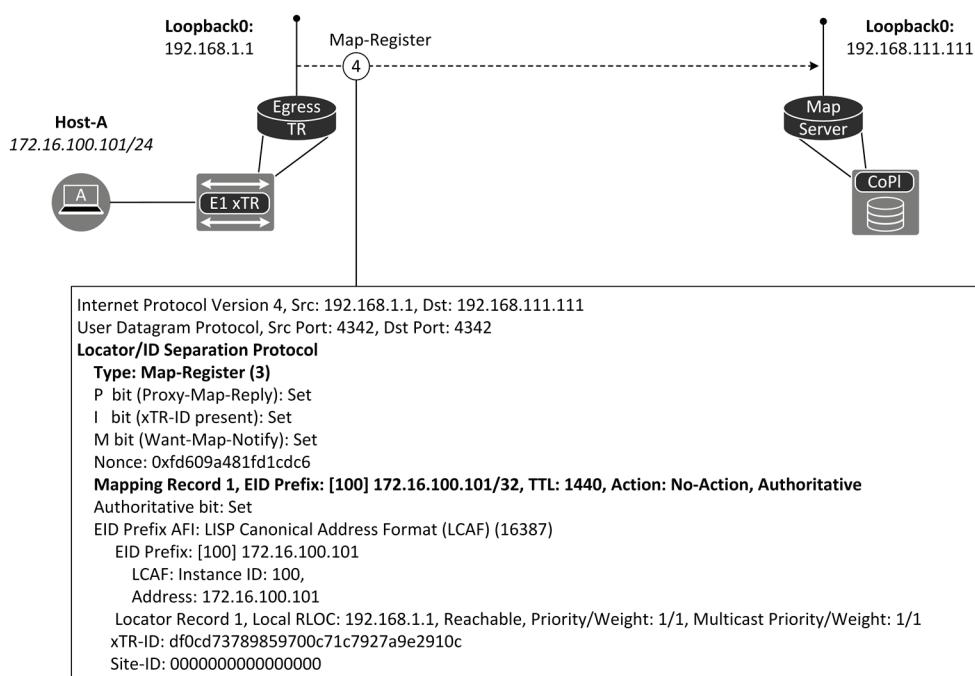


Figure 3-2: Map-Register message from ETR Edge-1.

Capture 3-1 on the next page shows the complete packet capture.

```
Ethernet II, Src: 50:00:00:01:00:02, Dst: 50:00:00:04:00:00
Internet Protocol Version 4, Src: 192.168.1.1, Dst: 192.168.111.111
User Datagram Protocol, Src Port: 4342, Dst Port: 4342
Locator/ID Separation Protocol
  0011 .... = Type: Map-Register (3)
  .... 1... = P bit (Proxy-Map-Reply): Set
  .... 0... = S bit (LISP-SEC capable): Not set
  .... 1... = I bit (xTR-ID present): Set
  .... 0... = R bit (Built for an RTR): Not set
  .... 0000 0000 0000 000. = Reserved bits: 0x0000
  .... 1 = M bit (Want-Map-Notify): Set
Record Count: 1
Nonce: 0xfd609a481fd1cdc6
Key ID: 0x0001
Authentication Data Length: 20
Authentication Data: 22c8e27db54073234d0685fe328f0ca865444daa
Mapping Record 1, EID Prefix: [100] 172.16.100.101/32, TTL: 1440, Action:
No-Action, Authoritative
  Record TTL: 1440
  Locator Count: 1
  EID Mask Length: 32
  000. .... = Action: No-Action (0)
  ...1 .... = Authoritative bit: Set
  .... 0000 0000 0000 = Reserved: 0x000
  0000 .... = Reserved: 0x0
  .... 0000 0000 0000 = Mapping Version: 0
  EID Prefix AFI: LISP Canonical Address Format (LCAF) (16387)
  EID Prefix: [100] 172.16.100.101
    LCAF: Instance ID: 100, Address: 172.16.100.101
      LCAF Header: 00000220000a
        Reserved bits: 0x00
        Flags: 0x00
        Type: Instance ID (2)
        Reserved bits: 0x20
        Length: 10
        Instance ID: 100
        Address AFI: IPv4 (1)
        Address: 172.16.100.101
Locator Record 1, Local RLOC: 192.168.1.1, Reachable, Priority/Weight:
1/1, Multicast Priority/Weight: 1/1
  Priority: 1
  Weight: 1
  Multicast Priority: 1
  Multicast Weight: 1
  Flags: 0x0005
    0000 0000 0000 0... = Reserved: 0x0000
    .... 1... = Local: Set
    .... 0... = Probe: Not set
    .... 1... = Reachable: Set
  AFI: IPv4 (1)
  Locator: 192.168.1.1
  xTR-ID: df0cd73789859700c71c7927a9e2910c
```

Capture 3-1: Map-Register message from E1xTR.

Debug 3-1 shows the partial Map-Register message process taken from the E1xTR when it received the GARP message from Host-A. Note that the debug is taken from a different time than the previous file capture 3-1. That is why the nonce values are different in capture 3-1 and debug 3-1.

The LISP process starts when E1xTR receives the GARP message from Host-A. Because the ingress interface is participating in dynamic-EID table 172.16.100.0/24-White that includes EID-prefix 172.16.100.0/24, E1xTR verifies if the source IP address of the received packet matches the list. Because the check is hit, E1xTR adds the IP address 172.16.100.101/32 into the dynamic EID table of Insurance-100 (L2 VRF 100_White). Next, E1xTR generates the nonce value, which is used as a message integrity check by receiving Map-Server and later by E1xTR itself when receiving Map-Notify message. Before sending an update to Map-Server, E1xTR verifies the reachability of the EID.

```
LISP: Processing dyn-EID detection for BDI100 EID prefix 172.16.100.101/32
LISP-0: Local dynEID 172_16_100_0-100_White IID 100 prefix 172.16.100.101/32
RLOC 192.168.1.1 pri/wei=1/1, Created (IPv4 intf RLOC Loopback0) (state:
active, rl.
LISP: RIBtable [IPv4:100_White:172.16.100.101/32] created.
LISP: RIBtable [IPv4:100_White:172.16.100.101/32] add source dyn-EIDi.
LISP-0: DynEID IID 100 172.16.100.101 [172_16_100_0-100_White:BDI100] Created.
LISP-0: IPv4 Map Server IID 100 192.168.111.111, reg_group_ctx src=0.0.0.0
dst=0.0.0.0 nonce=0xB6001A19-0x690B1D93, reg_group No RLOCs no merge, Starting
period.
LISP-0: Local dynEID 172_16_100_0-100_White IID 100 prefix 172.16.100.101/32,
found RIB route 172.16.100.101/32 to EID prefix (state: active, rlocs: 1/1,
source.
LISP: RIB Watch Group 100_White 172.16.100.0/24 (no-default), scheduling RIB
update.
LISP: RIB Watch Group 100_White 172.16.100.101/32 (no-default), scheduling RIB
update.
LISP: RIB Watch Group 100_White 172.16.100.0/24 (no-default), installing in
RIB.
LISP: RIB Next-hop 100_White BDI100 172.16.100.0 lcl 172.16.100.1, deleting.
LISP: RIB Watch Group 100_White 172.16.100.101/32 (no-default), installing in
RIB.
LISP-0: IID 100 Update registration group links for 1 prefixes.
LISP: RIB Next-hop 100_White BDI100 172.16.100.101 lcl 172.16.100.1, notified.
```

```
LISP-0: Local dynEID 172.16.100.0-100_White IID 100 prefix 172.16.100.101/32,
found RIB route 172.16.100.101/32 to EID prefix (state: active, rlocs: 1/1,
source.

LISP-0: Ping [dynamic-eid] 0x7FC23A6675B0 liveness 100_White IPv4 UNSPEC ->
172.16.100.101 Notify client 255% -> 100%.

LISP-0: IPv4 Map Server IID 100 192.168.111.111, Sending map-register (src_rloc
192.168.1.1) nonce 0xB6001A19-0x690B1D93.
```

Debug 3-1: Edge-1 generates the UDP based Map-Register message to Map-Server.

Registration Phase 3: UDP based Map-Notify from MS to ETR

When a Map-Server receives a Map-Register message from the E1xTR, it verifies that the EID-prefix and the Instance-Id match to its configuration. This means that there has to be an EID-prefix 172.16.100.0/24 in Instance -100 under the site Network-Times. In addition, the “nonce” value has to be bigger than what is received with the previous Map-Register message from the same ETR. When the verification process is passed successfully, the Map-Server copies the original “nonce” value, the EID-record and the RLOC-record information to the Map-Notify message which it sends to the ETR Edge-1.

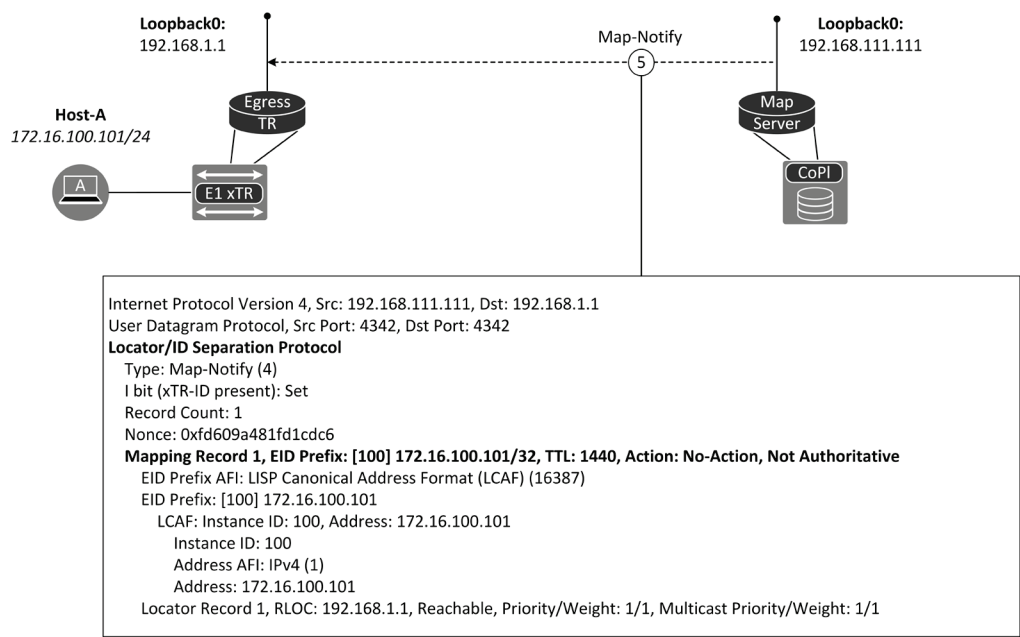


Figure 3-3: Map-Notify from Map-Server Node.

Capture 3-2 shows the complete packet capture.

Internet Protocol Version 4, Src: 192.168.111.111, Dst: 192.168.1.1
User Datagram Protocol, Src Port: 4342, Dst Port: 4342
Locator/ID Separation Protocol
0100 = Type: Map-Notify (4)
.... 1... = I bit (xTR-ID present): Set
.... .0.. = R bit (Built for an RTR): Not set
.... ..00 0000 0000 0000 = Reserved bits: 0x00000
Record Count: 1
Nonce: 0x609a481fd1cdc6
Key ID: 0x0001
Authentication Data Length: 20
Authentication Data: c3ab7a993de62c5cb53ca97267817847733d9bac
Mapping Record 1, EID Prefix: [100] 172.16.100.101/32, TTL: 1440, Action: No-Action, Not Authoritative
Record TTL: 1440
Locator Count: 1
EID Mask Length: 32
000. = Action: No-Action (0)
...0 = Authoritative bit: Not set
.... .000 0000 0000 = Reserved: 0x000
0000 = Reserved: 0x0
.... 0000 0000 0000 = Mapping Version: 0
EID Prefix AFI: LISP Canonical Address Format (LCAF) (16387)
EID Prefix: [100] 172.16.100.101
LCAF: Instance ID: 100, Address: 172.16.100.101
LCAF Header: 00000220000a
Instance ID: 100
Address AFI: IPv4 (1)
Address: 172.16.100.101
Locator Record 1, RLOC: 192.168.1.1, Reachable, Priority/Weight: 1/1, Multicast Priority/Weight: 1/1
Priority: 1
Weight: 1
Multicast Priority: 1
Multicast Weight: 1
Flags: 0x0001
0000 0000 0000 0... = Reserved: 0x0000
....0.. = Local: Not set
....0. = Probe: Not set
....1 = Reachable: Set
AFI: IPv4 (1)
Locator: 192.168.1.1
xTR-ID: df0cd73789859700c71c7927a9e2910c
Site-ID: 0000000000000000

Capture 3-2: Map-Notify from Map-Server Node.

Debug 3-2 illustrates the LISP process of Map-Server when it receives the UDP based unreliable Map-Register message from E1xTR. It creates a mapping record into the EID-to-RLOC mapping database. The EID-to-RLOC mapping entry is shown in figure 3-11 (page 40). The entry is attached to the site Network-Times with an IID 100 and the TTL 24 hours (1440sec.) and with the RLOC 192.168.1.1 (which reachability is verified from the RIB), and the registration is marked as Authoritative (received from EID first-hop-router - RLOC). After creating an EID-to-RLOC database entry, Map-Server sends a Map-Notify message to E1xTR. From the notify message, we can see the E1xTR Identifier and the nonce value used with this particular update.

```
LISP: Processing received Map-Register(3) message on GigabitEthernet1 from
192.168.1.1:4342 to 192.168.111.111:4342

LISP: Processing Map-Register mapping record for IID 100 172.16.100.101/32 LCAF
2, ttl 1440, action none, authoritative, 1 locator 192.168.1.1 pri/wei=1/1 LpR

LISP-0: IID 100 Parsing SVC_IP_IAF_IPv4 registration for prefix
172.16.100.101/32.

LISP: Processing IP Map-Register mapping record for IID 100 172.16.100.101/32
LCAF 2, ttl 1440, action none, authoritative, 1 locator 192.168.1.1 pri/wei=1/1
LpR

LISP-0: IID 100 Processing SVC_IP_IAF_IPv4 registration for prefix
172.16.100.101/32.

LISP-0: MS EID IID 100 prefix 172.16.100.101/32 SVC_IP_IAF_IPv4 site *,
Created.

LISP-0: MS Site EID IID 100 prefix 172.16.100.101/32 SVC_IP_IAF_IPv4 site
Network-Times, Created dynamic site EID prefix entry.

LISP-0: MS registration IID 100 prefix 172.16.100.101/32 192.168.1.1
SVC_IP_IAF_IPv4 site Network-Times, Created new registration.

LISP-0: MS registration IID 100 prefix 172.16.100.101/32 192.168.1.1
SVC_IP_IAF_IPv4 site Network-Times, Adding locator 192.168.1.1.

LISP: RIB Watch Group default 192.168.1.1/32 , created.

LISP: RIB Watch Group default 192.168.1.1/32 , scheduling RIB update.

LISP-0: MS inst member IID 100 192.168.1.1 Became valid.

LISP: Session VRF default, Local UNSPEC, Peer 192.168.1.1, Role: Passive,
State: Unknown, Created (RX 0, TX 0).
```



```

LISP-0: MS EID IID 100 prefix 172.16.100.101/32 SVC_IP_IAF_IPv4 site Network-
Times, Scheduling map notifications for prefix 172.16.100.101/32.

LISP-0: MS EID IID 100 prefix 172.16.100.101/32 SVC_IP_IAF_IPv4 site Network-
Times, Scheduling unreliable map notification for prefix 172.16.100.101/32 to
ETR 192.168.1.1.

LISP: RIB Watch Group default 192.168.1.1/32 , installing in RIB.

LISP-0: MS EID IID 100 prefix 172.16.100.101/32 SVC_IP_IAF_IPv4 site Network-
Times, ALT route update/create.

LISP-0: ALTroute IID 100 prefix 172.16.100.101/32 <-> created.

LISP-0: ALTroute IID 100 prefix 172.16.100.101/32 <-> add source MS-EID.

LISP: Session VRF default, Local UNSPEC, Peer 192.168.1.1, Role: Passive,
State: Unknown, Destroyed with OS adaptor (RX 0, TX 0).

LISP-0: Map-Notify 192.168.111.111:4342->192.168.1.1:4342 xTR-ID 0xB11A62C9-
0x8683B62B-0x78A9C166-0xA4EBE930, sending with 1 prefix, nonce 0xB6001A19-
0x690B1D93

```

Debug 3-2: *Map-Server processing UDP based Map-Register message from ETR Edge-1*

Debug 3-3 shows how E1xTR receives the Map-Notify message from Map-Server.

```

LISP: Processing received Map-Notify(4) message on GigabitEthernet1 from
192.168.111.111:4342 to 192.168.1.1:4342

LISP: Processing Map-Notify ID-included, no MS-RTR auth, no EID-notify, 1
record, nonce 0xB6001A19-0x690B1D93, key-id 1, auth-data-len 20, hash-function
sha1, xd

LISP: Processing Map-Notify mapping record for IID 100 172.16.100.101/32 LCAF
2, ttl 1440, action none, not authoritative, 1 locator 192.168.1.1 pri/wei=1/1
lpR

LISP-0: Local dynEID 172_16_100_0-100_white IID 100 prefix 172.16.100.101/32,
Received map notify (state: active, rlocs: 1/1, sources: dynamic).

```

Debug 3-3: *The ETR Edge-1 receives the Map-Notify from Map-Server.*

Registration Phase 4: TCP based Registration-Refresh from MS to ETR

The first part of the Map-Register process is done by using an unreliable UDP for message exchange. Using UDP for sending Map-Registration message though has some drawbacks. ETR sends these messages to Map-Server (MS) every minute. This means that ETR has to construct the message and then send it to MS. MS, in turn, has to process received Map-Register messages and then acknowledge them by sending Map-Notify messages. As said, this is a reoccurring process that happens every minute. When there are more EIDs than can be packed inside one Map-Register message, the information is split into multiple Map-Register messages. This, in turn, increases the load on both ETR and MS. Note that from the MS point of view, there might be tens or even hundreds of ETRs sending Map-Request on a one-minute interval.

IOS-XE supports the Reliable Registration process using TCP. After receiving a LISP Map-Notify message from Map-Server, E1xTR starts the three-way TCP handshake process to establish a reliable TCP-connection with Map-Server. This session is used as a reliable transport channel when exchanging LISP Map-Register messages. The status of TCP-connection is monitored by using a TCP ACK message (dst. port 4342). Figure 3-4 illustrates this process. The LISP Map-Register message is now sent only once instead of a one-minute interval.

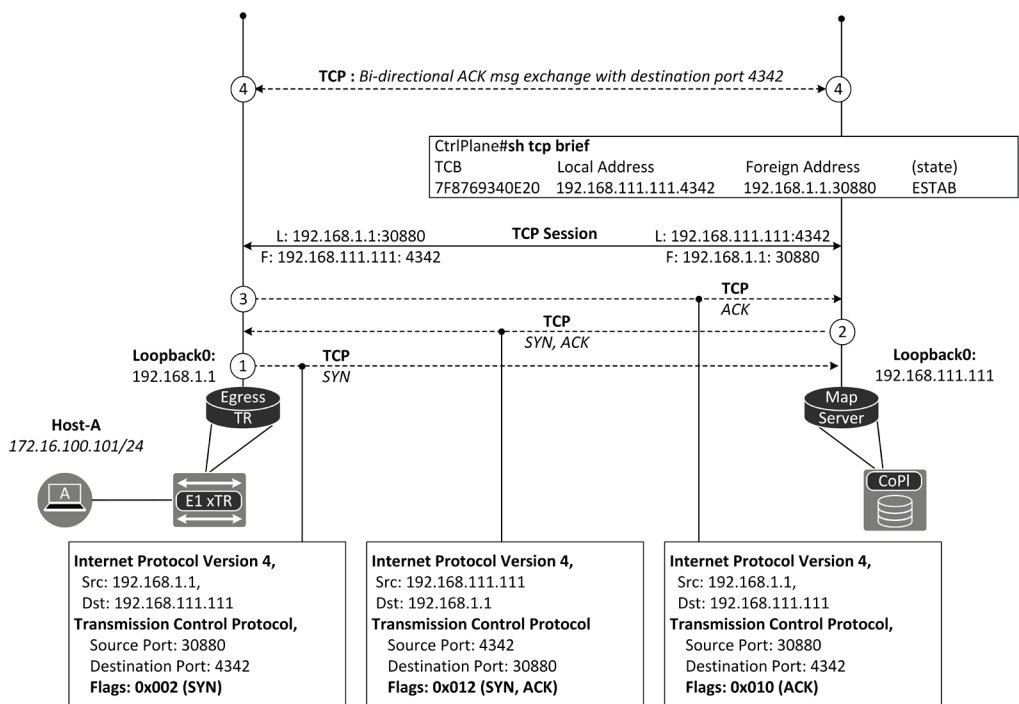


Figure 3-4: LISP Reliable Registration: TCP Three-Way Handshake.

After establishing the TCP session, the Map-Server sends a Registration-Refresh message where it asks E1xTR to send all EIDs that it has. Note that the end marker is always 0x9FACADE9.

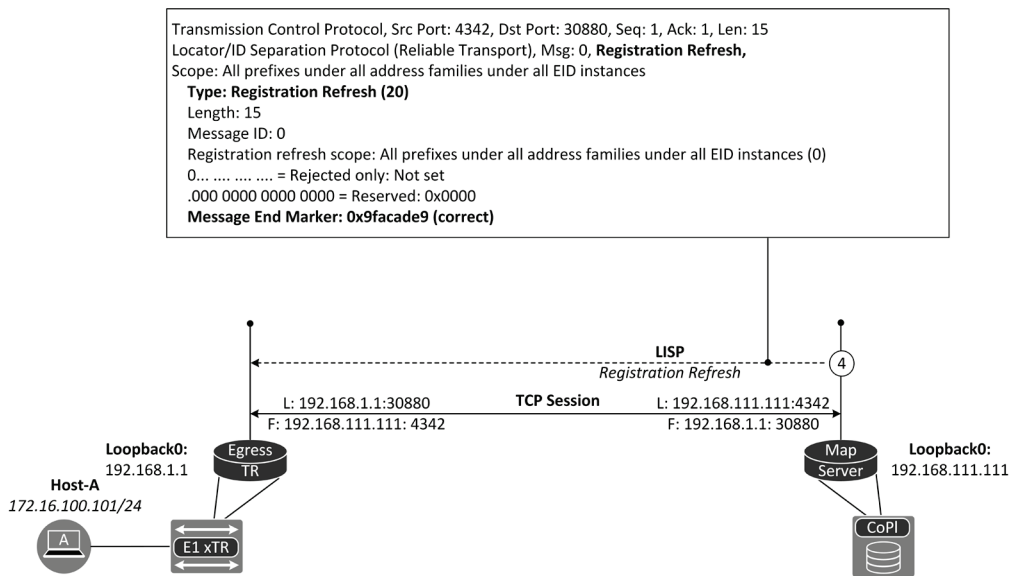


Figure 3-5: LISP Reliable Registration: Registration Refresh.

```
Internet Protocol Version 4, Src: 192.168.111.111, Dst: 192.168.1.1
Transmission Control Protocol, Src Port: 4342, Dst Port: 39626, Seq: 1, Ack: 1,
Len: 15
Locator/ID Separation Protocol (Reliable Transport), Msg: 0, Registration
Refresh, Scope: All prefixes under all address families under all EID instances
Type: Registration Refresh (20)
Length: 15
Message ID: 0
Registration refresh scope: All prefixes under all address families under
all EID instances (0)
0... .... = Rejected only: Not set
.000 0000 0000 0000 = Reserved: 0x0000
Message End Marker: 0x9facade9 (correct)
```

Capture 3-3: *LISP Reliable Registration: Registration-Refresh from Map-Server.*

Example 3-1 shows that there is a TCP connection between the CtrlPlane node and E1xTR.

```
CtrlPlane#sh tcp brief
TCB          Local Address          Foreign Address          (state)
7F8769340E20 192.168.111.111.4342      192.168.1.1.31299      ESTAB
7F87690709C0 192.168.111.111.4342      192.168.2.2.47876      ESTAB
```

Example 3-1: *TCP connection between E1xTR and Map-Server.*

Debug 3-4 shows how the Map-Server accepts the TCP-connection request from E1xTR and how it sends a Registration-Refresh message back to E1xTR.

```
LISP: OS TCP Session VRF: IPv4 default, Peer: 192.168.1.1:31299, Local: UNSPEC,
TCB: 0x7F8769340E20 Accepted connection, local address 192.168.111.111, remote
port 31299 .

LISP-0: Session User Type Error-Reporter/3 Peer 192.168.1.1 Role Passive State
Down , IPv4 RLOC default, Init.

LISP-0: Building reliable registration message registration-refresh for IID 0
EID 0.0.0.0/0 , Refresh Rejected: FALSE, Scope: global/0, EID AFI: invalid/0.
```

Debug 3-4: *TCP connection and Reliable Registration by Map-Server.*

Registration Phase 5: TCP based Map-Register from ETR to MS

After receiving the LISP Registration-Refresh message over a reliable TCP-session from Map-Server, E1xTR replies by constructing a message which includes information about its EIDs. The message carries Mapping Records and Locator Records for all EIDs that E1xTR has, Host-A in subnet 172.16.100.0/24 and Host-B in subnet 172.16.200.0/24 (Host-B not shown in the figure).

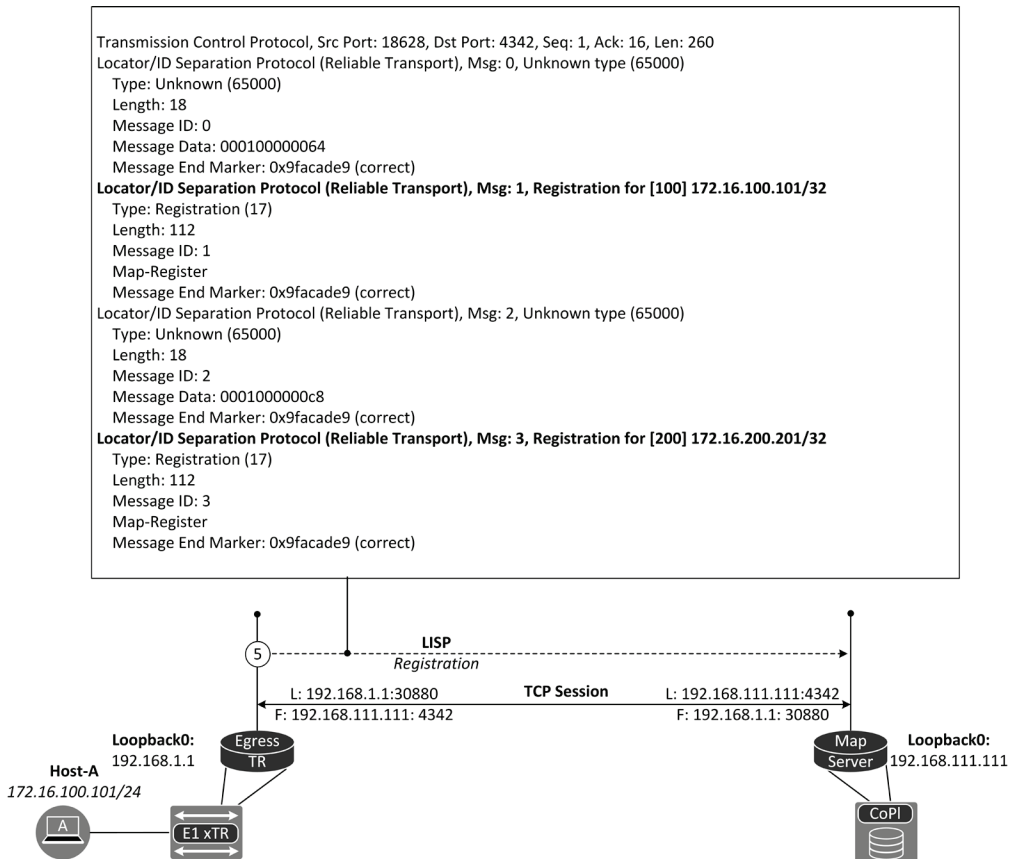


Figure 3-6: LISP Registration: Map-Register from ETR.

The information included in the TCP based reliable LISP Map-Register message is the same that was introduced in UDP based unreliable LISP Map-Register message.

```

Internet Protocol Version 4, Src: 192.168.1.1, Dst: 192.168.111.111
Transmission Control Protocol, Src Port: 39626, Dst Port: 4342, Seq: 1, Ack:
16, Len: 260
Locator/ID Separation Protocol (Reliable Transport), Msg: 0, Unknown type
(65000)
  Type: Unknown (65000)
  Length: 18
  Message ID: 0
  Message Data: 0001000000064
  Message End Marker: 0x9facade9 (correct)
Locator/ID Separation Protocol (Reliable Transport), Msg: 1, Registration for
[100] 172.16.100.101/32
  Type: Registration (17)
  Length: 112
  Message ID: 1
  Map-Register
    .... 1... .. = P bit (Proxy-Map-Reply): Set
    .... 0... .. = S bit (LISP-SEC capable): Not set
    .... .1. .... = I bit (xTR-ID present): Set
    .... ..0 .... = R bit (Built for an RTR): Not set
    .... .. 0000 0000 0000 000. = Reserved bits: 0x0000
    .... .. .. .. ..1 = M bit (Want-Map-Notify): Set
  Record Count: 1
  Nonce: 0x7e694534533d94c7
  Key ID: 0x0001
  Authentication Data Length: 20
  Authentication Data: e810b188478bc1087c3784bf915473d3100ea5a5
  Mapping Record 1, EID Prefix: [100] 172.16.100.101/32, TTL: 1440,
Action: No-Action, Authoritative
  Record TTL: 1440
  Locator Count: 1
  EID Mask Length: 32
  000. .... .. = Action: No-Action (0)
  ...1 .... .. = Authoritative bit: Set
  .... .000 0000 0000 = Reserved: 0x000
  0000 .... .. = Reserved: 0x0
  .... 0000 0000 0000 = Mapping Version: 0
  EID Prefix AFI: LISP Canonical Address Format (LCAF) (16387)
  EID Prefix: [100] 172.16.100.101
    LCAF: Instance ID: 100, Address: 172.16.100.101
      LCAF Header: 00000220000a
      Instance ID: 100
      Address AFI: IPv4 (1)
      Address: 172.16.100.101
    Locator Record 1, Local RLOC: 192.168.1.1, Reachable,
Priority/Weight: 1/1, Multicast Priority/Weight: 1/1
  Priority: 1
  Weight: 1
  Multicast Priority: 1
  Multicast Weight: 1
  Flags: 0x0005
    0000 0000 0000 0... = Reserved: 0x0000

```

```

        .... .1.. = Local: Set
        .... ..0. = Probe: Not set
        .... ...1 = Reachable: Set
    AFI: IPv4 (1)
    Locator: 192.168.1.1
    xTR-ID: 8eda5a9c19ab66fc19ce78315551ea65
    Site-ID: 0000000000000000
    Message End Marker: 0x9facade9 (correct)
Locator/ID Separation Protocol (Reliable Transport), Msg: 2, Unknown type
(65000)
    Type: Unknown (65000)
    Length: 18
    Message ID: 2
    Message Data: 00010000000c8
    Message End Marker: 0x9facade9 (correct)
Locator/ID Separation Protocol (Reliable Transport), Msg: 3, Registration for
[200] 172.16.200.201/32
    Type: Registration (17)
    Length: 112
    Message ID: 3
    Map-Register
        .... 1... = P bit (Proxy-Map-Reply): Set
        .... .0.. = S bit (LISP-SEC capable): Not set
        .... ..1. = I bit (xTR-ID present): Set
        .... ...0 = R bit (Built for an RTR): Not set
        .... 0000 0000 0000 000. = Reserved bits: 0x0000
        .... ...1 = M bit (Want-Map-Notify): Set
    Record Count: 1
    Nonce: 0x338544be79c1d8bb
    Key ID: 0x0001
    Authentication Data Length: 20
    Authentication Data: 0da652e14d3de3b16e75c34233ef3082265c9a2b
    Mapping Record 1, EID Prefix: [200] 172.16.200.201/32, TTL: 1440,
Action: No-Action, Authoritative
    Record TTL: 1440
    Locator Count: 1
    EID Mask Length: 32
    000. .... = Action: No-Action (0)
    ...1 .... = Authoritative bit: Set
    .... .000 0000 0000 = Reserved: 0x000
    0000 .... = Reserved: 0x0
    .... 0000 0000 0000 = Mapping Version: 0
    EID Prefix AFI: LISP Canonical Address Format (LCAF) (16387)
    EID Prefix: [200] 172.16.200.201
        LCAF: Instance ID: 200, Address: 172.16.200.201
            LCAF Header: 00000220000a
            Instance ID: 200
            Address AFI: IPv4 (1)
            Address: 172.16.200.201
    Locator Record 1, Local RLOC: 192.168.1.1, Reachable,
Priority/Weight: 1/1, Multicast Priority/Weight: 1/1
    Priority: 1
    Weight: 1
    Multicast Priority: 1
    Multicast Weight: 1

```

```

Flags: 0x0005
      0000 0000 0000 0... = Reserved: 0x0000
      .... .... .... .1.. = Local: Set
      .... .... .... ..0. = Probe: Not set
      .... .... .... ...1 = Reachable: Set
AFI: IPv4 (1)
Locator: 192.168.1.1
xTR-ID: 8eda5a9c19ab66fc19ce78315551ea65
Site-ID: 0000000000000000
Message End Marker: 0x9facade9 (correct)

```

Capture 3-4: *LISP Reliable Registration: Map-Register from E1xTR.*

Debug 3-5 shows how the E1xTR receives the TCP based reliable Registration-Refresh message from Map-Server. As a reaction, it changes the current UDP-based connection to TCP-connection using the previously established TCP-session and builds a new reliable LISP Map-Register message to Map-Server.

```

LISP: Session VRF default, Local 192.168.1.1, Peer 192.168.111.111, Role:
Active, State: Up, Received reliable registration message registration-refresh
for IID.
LISP-0: IPv4 Map Server IID 100 192.168.111.111, Processing reliable REFRESH
message Scope global.
LISP-0: IPv4 Map Server IID 100 192.168.111.111, prev_state = UDP, event =
RX_REFRESH, new_state = TCP.
LISP-0: IPv4 Map Server IID 100 192.168.111.111, reg_group_ctx src=0.0.0.0
dst=0.0.0.0 nonce=0xB6001A19-0x690B1D93, reg_group RLOCs merge, Stopping
periodic gro.
LISP-0: IPv4 Map Server IID 100 192.168.111.111, reg_group_ctx src=0.0.0.0
dst=0.0.0.0 nonce=0xB6001A19-0x690B1D93, reg_group RLOCs no merge, Stopping
periodic .
LISP-0: IPv4 Map Server IID 100 192.168.111.111, reg_group_ctx src=0.0.0.0
dst=0.0.0.0 nonce=0xB6001A19-0x690B1D93, reg_group No RLOCs merge, Stopping
periodic .
LISP-0: IPv4 Map Server IID 100 192.168.111.111, reg_group_ctx src=0.0.0.0
dst=0.0.0.0 nonce=0xB6001A19-0x690B1D93, reg_group No RLOCs no merge, Stopping
period.
LISP-0: IPv4 Map Server IID 200 192.168.111.111, Processing reliable REFRESH
message Scope global.
LISP-0: IPv4 Map Server IID 200 192.168.111.111, prev_state = UDP, event =
RX_REFRESH, new_state = TCP.
LISP-0: Building reliable registration message wlc subscribe for IID 100 EID
NULL , EID AFI: IPv4/1.
LISP-0: IPv4 Map Server IID 100 192.168.111.111, Built reliable map register
message. Size = 112.

```

Debug 3-5: *Reliable Map-register message by E1xTR.*

Registration Phase 6: TCP based Registration ACK from MS to ETR

When the Map-Server receives the LISP Map-Register message, it acknowledges it by sending a *Registration Ack* containing each EID that was introduced in the Map-register message sent by E1xTR.

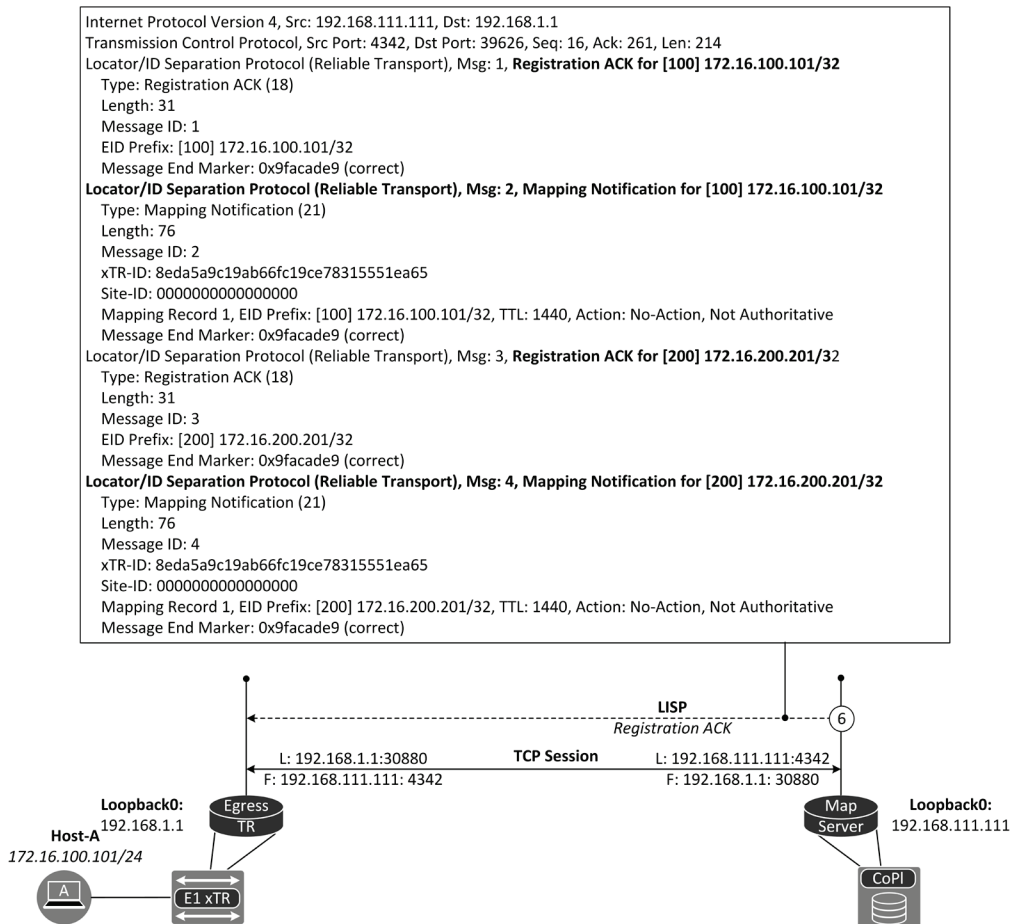


Figure 3-7: LISP Registration: Registration ACK from MS.

```

Internet Protocol Version 4, Src: 192.168.1.1, Dst: 192.168.111.111
Transmission Control Protocol, Src Port: 39626, Dst Port: 4342, Seq: 261, Ack:
230, Len: 0
    Source Port: 39626
    Destination Port: 4342
    [Stream index: 1]
    [TCP Segment Len: 0]
    Sequence number: 261      (relative sequence number)
    [Next sequence number: 261      (relative sequence number)]
    Acknowledgment number: 230      (relative ack number)
    0101 .... = Header Length: 20 bytes (5)
    Flags: 0x010 (ACK)
        000. .... = Reserved: Not set
        ...0 .... = Nonce: Not set
        .... 0... = Congestion Window Reduced (CWR): Not set
        .... 0... = ECN-Echo: Not set
        .... ..0. = Urgent: Not set
        .... ...1 = Acknowledgment: Set
        .... .... 0... = Push: Not set
        .... .... 0.. = Reset: Not set
        .... .... ..0. = Syn: Not set
        .... .... ...0 = Fin: Not set
    [TCP Flags: .....A....]
    Window size value: 16155
    [Calculated window size: 16155]
    [Window size scaling factor: -2 (no window scaling used)]
    Checksum: 0xa882 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
    [SEQ/ACK analysis]
    [Timestamps]

```

Capture 3-5: *LISP Reliable Registration: Acknowledgment from Map-Server.*

Debug 3-6 shows how Map-Server generates the Registration-Ack .

```

LISP: Session VRF default, Local UNSPEC, Peer 192.168.1.1, Role: Passive,
State: Up, Received reliable registration message wlc subscribe for IID 100
EID 0.0.0.0/0 , EID.

LISP-0: MS rr client IID 100 192.168.1.1 Created (AF: SVC_IP_IAF_IPv4, state:
idle, session: Down).

LISP-0: Session User Type reliable-registration-ms/4 Peer 192.168.1.1 Role
Passive State Down MS 0 IID 100 AF IPv4 reliable registration client
192.168.1.1, Init.

LISP-0: MS WLC client IID 100 AFI IPv4 Peer 192.168.1.1, Created.

LISP-0: MS WLC client IID 100 AFI IPv4 Peer 192.168.1.1, Scheduling full
update.

```

```

LISP: Processing Map-Register proxy, map-notify, no merge, no security, no
mobile-node, not to-RTR, no fast-map-register, no EID-notify, ID-included, 1
record, nonce 0xB6d

LISP: Processing Map-Register mapping record for IID 100 172.16.100.101/32 LCAF
2, ttl 1440, action none, authoritative, 1 locator
    192.168.1.1 pri/wei=1/1 LpR

LISP-0: IID 100 Parsing SVC_IP_IAF_IPv4 registration for prefix
172.16.100.101/32.

LISP: Processing IP Map-Register mapping record for IID 100 172.16.100.101/32
LCAF 2, ttl 1440, action none, authoritative, 1 locator

    192.168.1.1 pri/wei=1/1 LpR

LISP-0: IID 100 Processing SVC_IP_IAF_IPv4 registration for prefix
172.16.100.101/32.

LISP: Session VRF default, Local UNSPEC, Peer 192.168.1.1, Role: Passive,
State: Up, Received reliable registration message registration for IID 100 EID
172.16.100.101/3.

LISP-0: MS registration IID 100 prefix 172.16.100.101/32 192.168.1.1
SVC_IP_IAF_IPv4 site Network-Times, Found existing registration, updating.

LISP-0: MS EID IID 100 prefix 172.16.100.101/32 SVC_IP_IAF_IPv4 site Network-
Times, Scheduling map notifications for prefix 172.16.100.101/32.

LISP-0: MS EID IID 100 prefix 172.16.100.101/32 SVC_IP_IAF_IPv4 site Network-
Times, Scheduling reliable map notification for prefix 172.16.100.101/32 to ETR
192.168.1.1 w.

LISP: Session VRF default, Local UNSPEC, Peer 192.168.1.1, Role: Passive,
State: Up, Received reliable registration message wlc subscribe for IID 200
EID 0.0.0.0/0 , EID.

LISP-0: MS rr client IID 200 192.168.1.1 Created (AF: SVC_IP_IAF_IPv4, state:
idle, session: Down).

LISP-0: Session User Type reliable-registration-ms/4 Peer 192.168.1.1 Role
Passive State Down MS 0 IID 200 AF IPv4 reliable registration client
192.168.1.1, Init.

LISP-0: MS WLC client IID 200 AFI IPv4 Peer 192.168.1.1, Created.

LISP-0: Building reliable registration message registration-ack for IID 100
EID 172.16.100.101/32 .

```

Debug 3-6: *Registration-Ack message by Map-Server.*

Debug 3-7 shows how E1xTR receives the Registration-ACK message from the Map-Server.

```
LISP: Session VRF default, Local 192.168.1.1, Peer 192.168.111.111, Role:
Active, State: Up, Received reliable registration message registration-ack for
IID 100.

LISP-0: IPv4 Map Server IID 100 192.168.111.111, Processing reliable ACK
message for prefix 172.16.100.101/32.

LISP: Session VRF default, Local 192.168.1.1, Peer 192.168.111.111, Role:
Active, State: Up, Received reliable registration message mapping-notification
for IID.

LISP-0: Local dynEID 172_16_100_0-100_White IID 100 prefix 172.16.100.101/32,
Received map notify (state: active, rlocs: 1/1, sources: dynamic).

LISP-0: IPv4 Map Server IID 100 192.168.111.111, Processing reliable NOTIFY
message for prefix 172.16.100.101/32.
```

Debug 3-7: E1xTR receives the Registration-Ack from Map-Server.

Registration Phase 7: Reliable TCP ACK from ETR to MS

As a reaction to the LISP Registration Ack message received from Map-Server, E1xTR sends a TCP-Ack message. Now the registration process is complete from the E1xTR perspective.

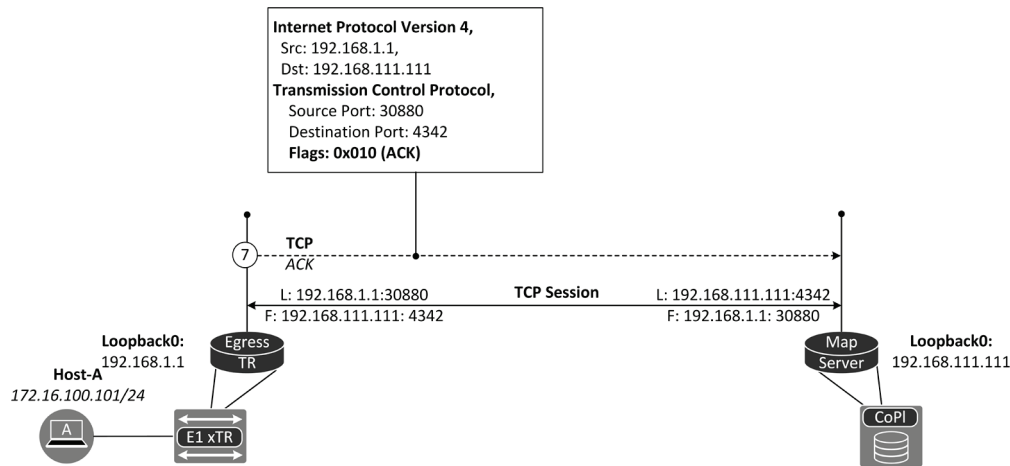


Figure 3-8: LISP Registration: Reliable Transport ACK.

Capture 3-6 shows the complete capture.

```

Internet Protocol Version 4, Src: 192.168.1.1, Dst: 192.168.111.111
Transmission Control Protocol, Src Port: 39626, Dst Port: 4342, Seq: 261, Ack:
230, Len: 0
    Source Port: 39626
    Destination Port: 4342
    [Stream index: 1]
    [TCP Segment Len: 0]
    Sequence number: 261      (relative sequence number)
    [Next sequence number: 261      (relative sequence number)]
    Acknowledgment number: 230      (relative ack number)
    0101 .... = Header Length: 20 bytes (5)
    Flags: 0x010 (ACK)
        000. .... = Reserved: Not set
        ...0 .... = Nonce: Not set
        .... 0... = Congestion Window Reduced (CWR): Not set
        .... .0.. = ECN-Echo: Not set
        .... ..0. = Urgent: Not set
        .... ...1 .... = Acknowledgment: Set
        .... .... 0... = Push: Not set
        .... .... .0.. = Reset: Not set
        .... .... ..0. = Syn: Not set
        .... .... ...0 = Fin: Not set
        [TCP Flags: .....A.....]
    Window size value: 16155
    [Calculated window size: 16155]
    [Window size scaling factor: -2 (no window scaling used)]
    Checksum: 0xa882 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
    [SEQ/ACK analysis]
    [Timestamps]

```

Capture 3-6 LISP Reliable Registration: Acknowledgment from ExTR.

Registration Phases 1-6 Summary

Figure 3-9 summarizes the LISP Registration process.

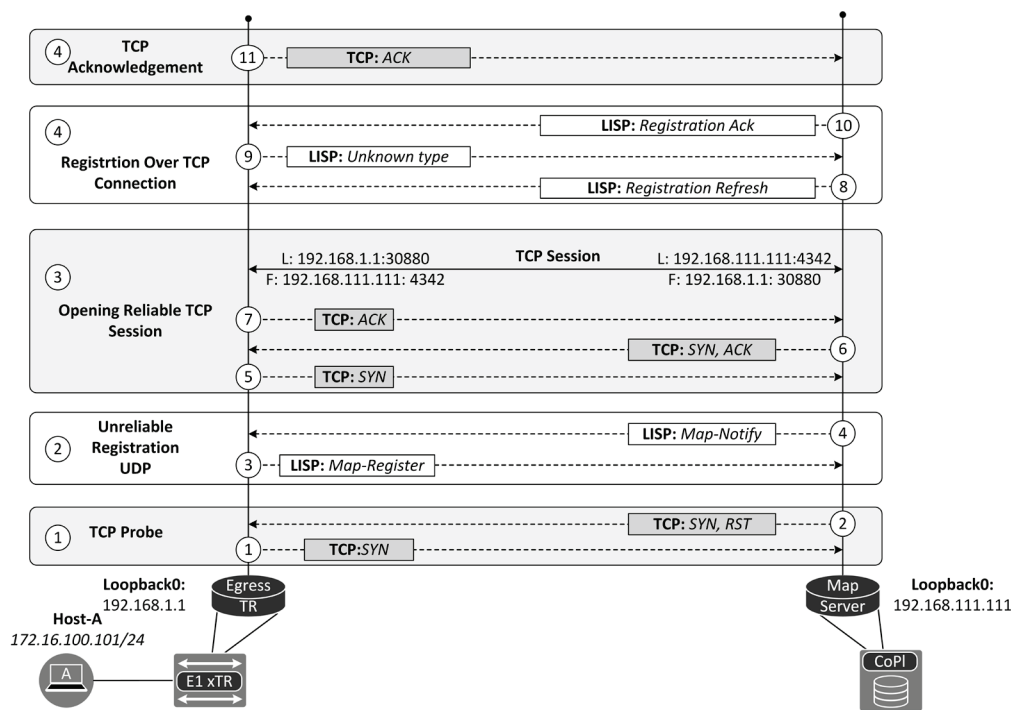


Figure 3-9: LISP Registration: Process Summary.

EID-to-RLOC Registration Verification

The example below shows that the Map-Resolver component of Control Plane node has a site named as “Network-Times” which has Instance-Id 100 and 200. Both ETRs Edge-1 and Edge-2 have registered their connected hosts using the reliable registration process. The command **show lisp site** also verifies that all registrations have been done using reliable TCP transport.

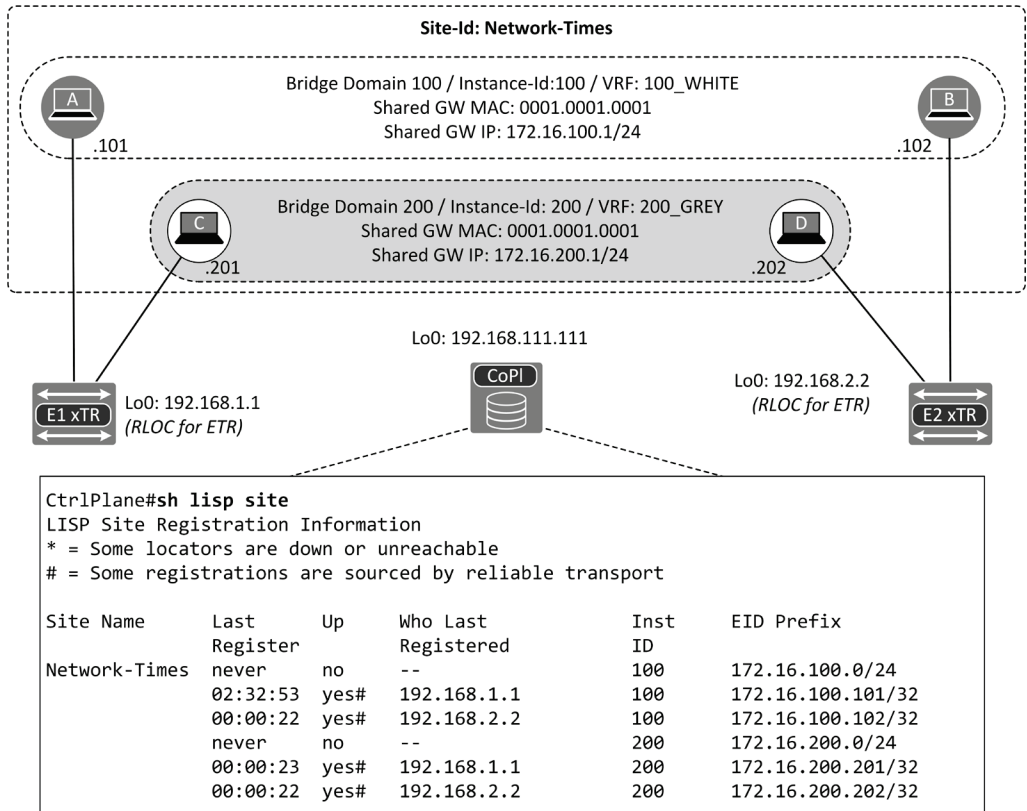


Figure 3-10: *show lisp site*.

Example 3-11 shows detailed information about EID 172.16.100.101/32. The EID 172.16.100.101/32 belongs to Dynamic-EID 172.16.100.0/24 defined under instance 100 and it is valid for one day. It is registered by ETR with RLOC 192.168.1.1 which is identified by xTR ID shown in the example. EID-prefix “nonce” is also shown in the example. The example also shows the ETR specific information such as priority and weight values. The “no security-capability” means that LISP advanced security features are not used with ETR 192.168.1.1 and Control Plane messages integrity is based on “nonce” value.

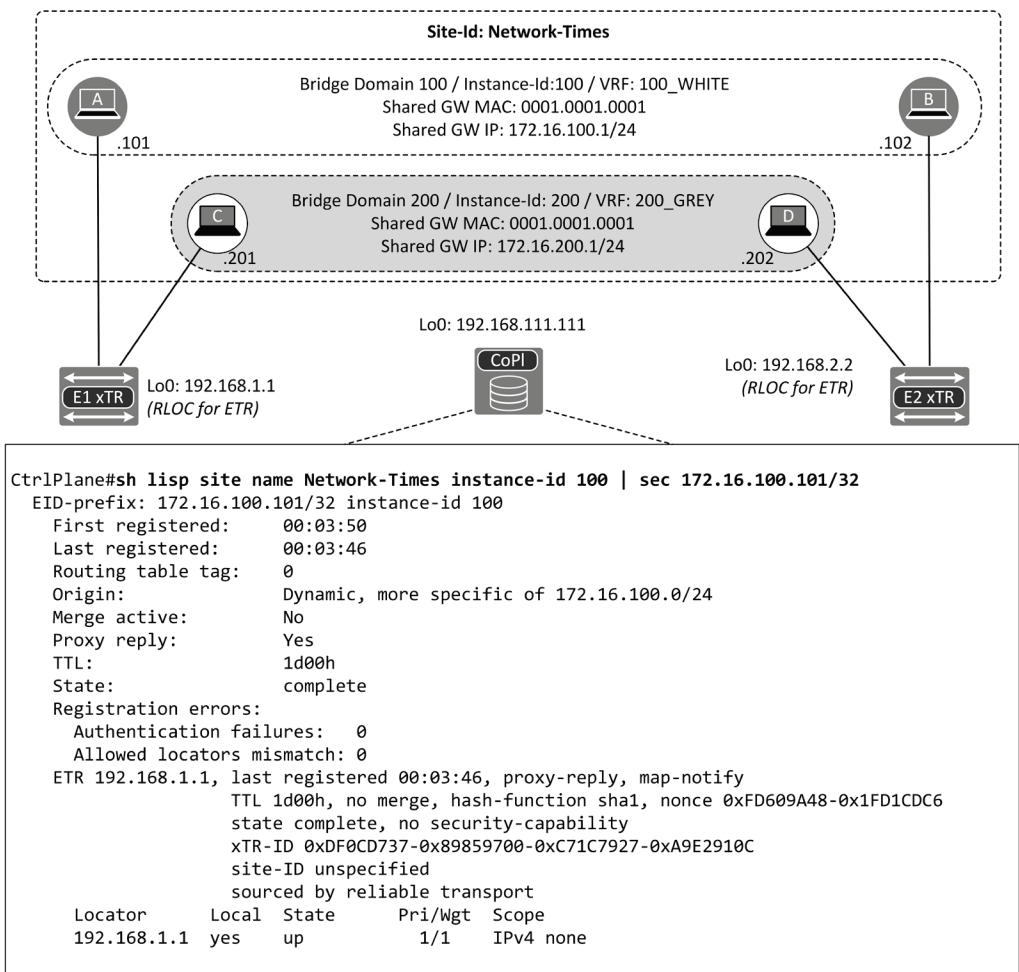


Figure 3-11: show lisp site name Network-Times instance-id 100 | sec 172.16.100.101/32.

The example below illustrates the information related to local EID 172.16.100.101/32 on E1xTR. EID belongs to Layer 3 VPN VRF 100_White with Instance-Id 100. Locator Status Bit (LSB) is set one meaning locator is reachable (Loopback 0 is Up). The EID 172.16.100.101/32 matches the dynamic-EID table 172_10_100_0_100_White and the Map-Register message is sent to Map-Server using RLOC information defined in RLOC-SET1. In addition, the example shows the Locator priority and weight values and the state of the RLOC.

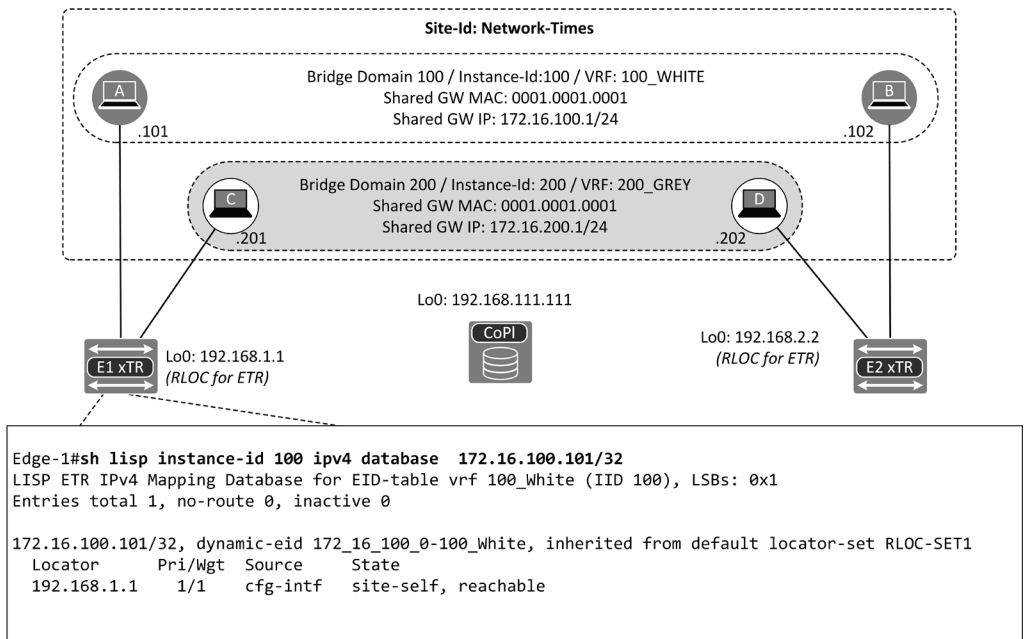


Figure 3-12: *show lisp instance-id 100 ipv4 database 172.16.100.101/32.*

The example 3-13 shows the Dynamic-EID tables on VRF 100_White. EID-prefix is 172.16.100.0/24 and the RLOC for this prefix is defined in RLOC-SET1. dynamic EIDs discovered and added to this table are sent to global Map-Server (192.168.111.111). There is one dynamic EID 172.16.100.101/32 discovered and installed into the Dynamic-EID table.

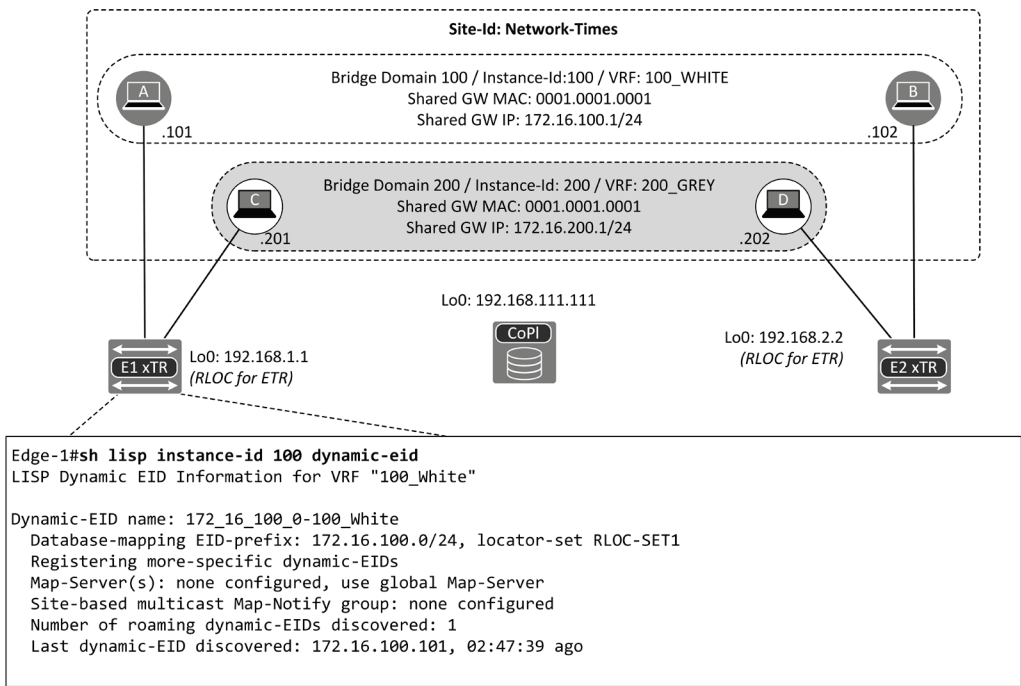


Figure 3-13: *show lisp instance-id 100 dynamic-eid.*

References

- [RFC 6830] D. Farinacci et al., "The Locator/ID Separation Protocol (LISP)", RFC 6830, January 2013.
- [RFC 6833] V. Fuller and D. Farinacci., "Locator/ID Separation Protocol (LISP) Map-Server Interface", RFC 6833, January 2013.
- [LISP Control Plane] D. Farinacci et al., "The Locator/ID Separation Protocol (LISP) Control Plane", draft-ietf-lisp-rdc6833bis-25, June 16, 2019.
- [VXLAN-GPE] J. Lemon et al., "Croup Policy Encoding with VXLAN GPE", draft-ietf-vxlan-gpe-gbp-02, March 2, 2019.
- [LISP Rel-Tran] C. Cassar et al., "LISP Reliable Transport", draft-kouvelas-lisp-reliable-transport-02, March 26, 2015.

The LISP Network: Evolution of the Next Generation of Data Networks
ISBN-10: 1-58714-471-9 – Victor Moreno, and Dino Farinacci

LISP Network Deployment and Troubleshooting: The Complete Guide to LISP
Implementation on IOS-XE, IOS-XR, and NX-OS
ISBN-10: 1-58714-506-5 – Tarique Shakil, Vinit Jain, and Yves Louis

Chapter 4: Control-Plane: EID-to-RLOC Request

Request Phase 1: Local Proxy-ARP by ITR

This chapter describes the Map-Request and Map-Reply processes by using a simple ping example from Host-A to Host-B. Let's say that this is the first time when Host-A starts communication with Host-B. First, Host-A has to resolve the IP-MAC binding, so it sends an ARP-Request, where it asks what is the MAC address of host who owns the IP address 172.16.100.102. The ARP-Request is sent as an L2 broadcast, and it eventually reaches E1xTR. Under the routing interface of Bridge-Domain 100 on E1xTR, there is a command **ip local proxy-arp**, meaning that E1xTR is allowed to answer ARP-Requests on behalf of hosts whose MAC-IP bindings are requested. E1xTR sends the ARP-Reply using its Anycast MAC address 0001.0001.0001.

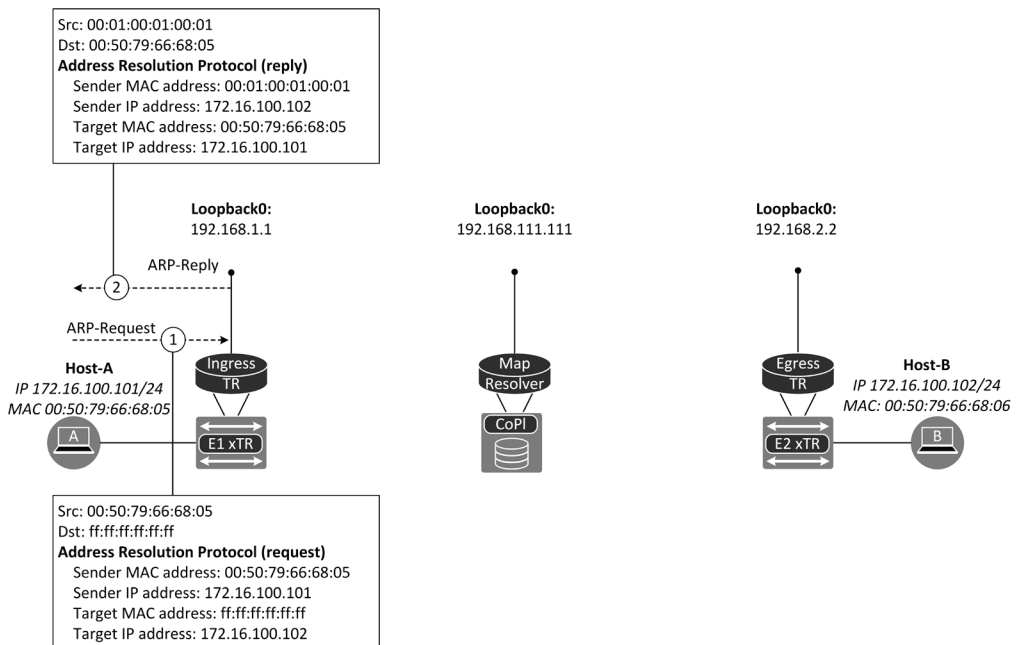


Figure 4-1: Local Proxy-ARP by E1xTR.

Request Phase 2: Encapsulated Control Message (ECM) by ITR

Figure 4-2 illustrates the ECM process. At this phase, E1xTR does not know the location of Host-B because this is the first time when one of its connected hosts starts communication with it. The E1xTR sends an *Encapsulated Control Message* (ECM) that carries the Map-Request message to Map-Resolver. The destination IP in the outer IP header is the RLOC of Map-Resolver and the source IP address is the RLOC of ITR. UDP destination port is 4342. Requested EID is used as both source and destination IP address in the inner IP header.

The type field in the LISP Control Message header states that this is a Map-Request message. The *M-bit*, when set, means that Map-Reply Record is included in the message. The “nonce” -value is used for the same purpose that was explained in the Map-Register section. Source EID information is encoded into LISP Canonical Address Format (LCAF) which has assigned an Address Family Identifier (AFI) 16387 by IANA. LCAF Type-2 is used for segmentation. It describes the source EID and its Instance-ID. When Map-Resolver receives the Map-Request it knows from which site instance the EID-to-RLOC database lookup should do. The last fields of Control-Plane header describe a sending ITR RLOC.

The *Map-Request Record* uses LCAF for describing the requested EID and its related Instance-Id. The Map-Reply Record defines the source EID. It includes the EID-to-RLOC information used for source EID. The information is the same that what was used in Map-Register message including IID, EID, TTL information about EID and RLOC, Priority/Weight of sending xTR. This way the receiving ETR/MS is able to cache the information if needed.

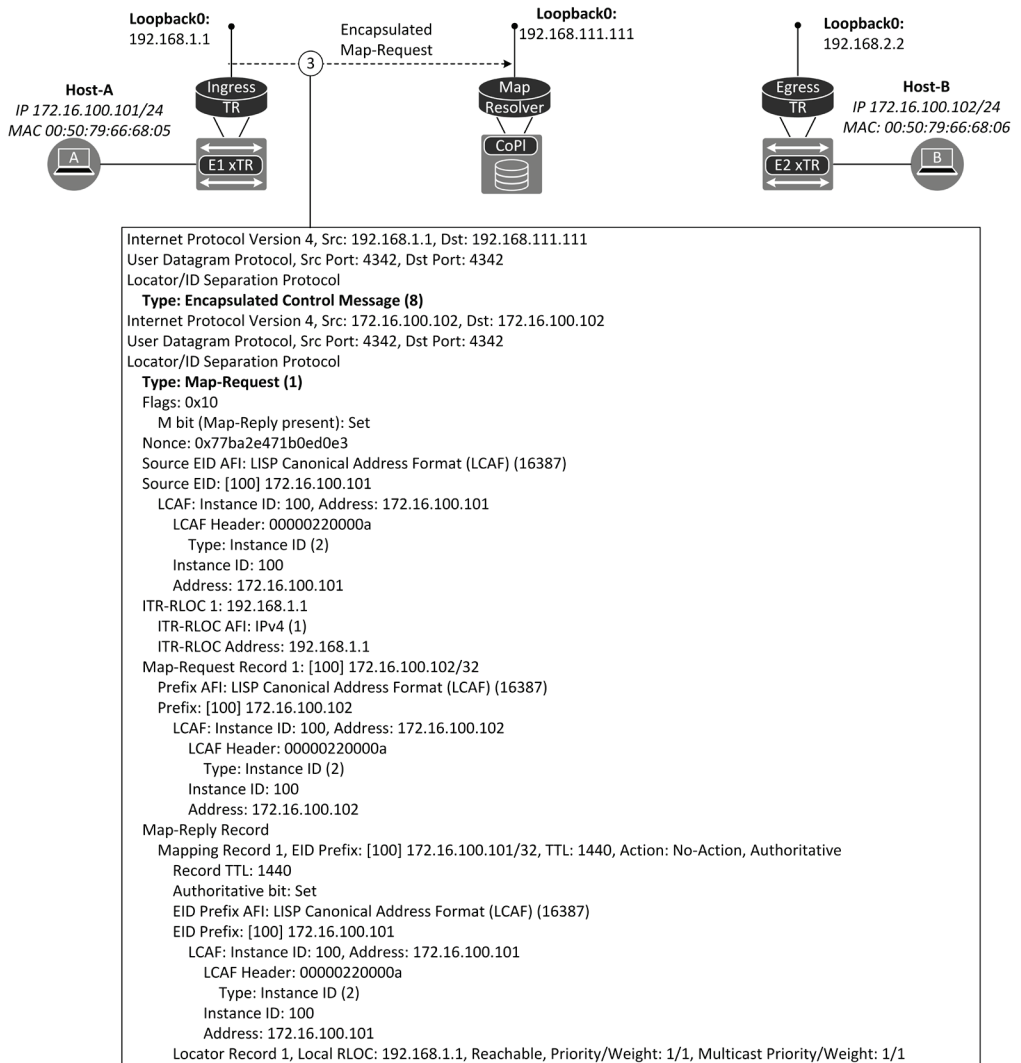


Figure 4-2: Encapsulated Control Message: Map-Request by E1xTR.

```
Internet Protocol Version 4, Src: 192.168.1.1, Dst: 192.168.111.111
User Datagram Protocol, Src Port: 4342, Dst Port: 4342
Locator/ID Separation Protocol
  1000 .... = Type: Encapsulated Control Message (8)
  .... 0... = S bit (LISP-SEC capable): Not set
  .... .0... = D bit (DDT-originated): Not set
  .... ..00 0000 0000 0000 0000 0000 0000 = Reserved bits: 0x00000000
Internet Protocol Version 4, Src: 172.16.100.102, Dst: 172.16.100.102
User Datagram Protocol, Src Port: 4342, Dst Port: 4342
Locator/ID Separation Protocol
  0001 .... = Type: Map-Request (1)
  .... 0100 00.. = Flags: 0x10
  .... 0... = A bit (Authoritative): Not set
  .... .1... = M bit (Map-Reply present): Set
  .... ..0. = P bit (Probe): Not set
  .... ...0 = S bit (Solicit-Map-Request): Not set
  .... .... 0... = p bit (Proxy ITR): Not set
  .... .... .0.. = s bit (SMR-invoked): Not set
  .... .... ..00 0000 000. = Reserved bits: 0x000
  .... .... .... ..0 0000 = ITR-RLOC Count: 0
Record Count: 1
Nonce: 0x77ba2e471b0ed0e3
Source EID AFI: LISP Canonical Address Format (LCAF) (16387)
Source EID: [100] 172.16.100.101
  LCAF: Instance ID: 100, Address: 172.16.100.101
    LCAF Header: 00000220000a
      Reserved bits: 0x00
      Flags: 0x00
      Type: Instance ID (2)
      Reserved bits: 0x20
      Length: 10
      Instance ID: 100
      Address AFI: IPv4 (1)
      Address: 172.16.100.101
ITR-RLOC 1: 192.168.1.1
  ITR-RLOC AFI: IPv4 (1)
  ITR-RLOC Address: 192.168.1.1
Map-Request Record 1: [100] 172.16.100.102/32
  Reserved: 0x00
  Prefix Length: 32
  Prefix AFI: LISP Canonical Address Format (LCAF) (16387)
  Prefix: [100] 172.16.100.102
    LCAF: Instance ID: 100, Address: 172.16.100.102
      LCAF Header: 00000220000a
        Reserved bits: 0x00
        Flags: 0x00
        Type: Instance ID (2)
        Reserved bits: 0x20
        Length: 10
        Instance ID: 100
        Address AFI: IPv4 (1)
        Address: 172.16.100.102
Map-Reply Record
```



```

Mapping Record 1, EID Prefix: [100] 172.16.100.101/32, TTL: 1440,
Action: No-Action, Authoritative
    Record TTL: 1440
    Locator Count: 1
    EID Mask Length: 32
    000. .... = Action: No-Action (0)
    ...1 .... = Authoritative bit: Set
    ....0000 0000 = Reserved: 0x000
    0000 .... = Reserved: 0x0
    ....0000 0000 0000 = Mapping Version: 0
    EID Prefix AFI: LISP Canonical Address Format (LCAF) (16387)
    EID Prefix: [100] 172.16.100.101
        LCAF: Instance ID: 100, Address: 172.16.100.101
            LCAF Header: 00000220000a
                Reserved bits: 0x00
                Flags: 0x00
                Type: Instance ID (2)
                Reserved bits: 0x20
                Length: 10
                Instance ID: 100
                Address AFI: IPv4 (1)
                Address: 172.16.100.101
        Locator Record 1, Local RLOC: 192.168.1.1, Reachable,
Priority/Weight: 1/1, Multicast Priority/Weight: 1/1
    Priority: 1
    Weight: 1
    Multicast Priority: 1
    Multicast Weight: 1
    Flags: 0x0005
        0000 0000 0000 0... = Reserved: 0x0000
        .... .... .1.. = Local: Set
        .... .... ..0. = Probe: Not set
        .... .... ...1 = Reachable: Set
    AFI: IPv4 (1)
    Locator: 192.168.1.1

```

Capture 4-1 *Encapsulated Control Message: Map-Request by E1xTR.*

Debug 4-1 illustrates the process where Host-A wants to send data (ICMP-Request) to Host-B. At this phase, E1xTR does not know how to reach requested the target IP address 172.16.100.102 so it generates the Map-Request and sends it to Map-Resolver wrapped inside the Encapsulated Control Message.

```
LISP: IPv4, Created link layer neighbor request entry for 172.16.100.102 on
BDI100 with timeout 00:00:59.

LISP-0: Remote EID IID 100 prefix 172.16.100.102/32, Change state to incomplete
(sources: <signal>, state: unknown, rlocs: 0).

LISP-0: Remote EID IID 100 prefix 172.16.100.102/32, [incomplete] Scheduling
map requests delay 00:00:00 min_elapsed 00:00:01 (sources: <signal>, state:
incomplete, rlocs: 0).

LISP-0: Remote EID IID 100 prefix 172.16.100.102/32, Process data signal, map
request already issued (sources: <signal>, state: incomplete, rlocs: 0).

LISP: IPv4, Created link layer neighbor proxy reply entry for source
172.16.100.101, target 172.16.100.102 on BDI100 with timeout 00:00:10.

LISP-0: Remote EID IID 100 prefix 172.16.100.102/32, Extranet IID passed to CEF
is 16777216 (sources: <signal>, state: incomplete, rlocs: 0).

LISP-0: IID 100 Request processing of remote EID prefix map requests to IPv4.

LISP: Send map request type remote EID prefix

LISP: Send map request for EID prefix IID 100 172.16.100.102/32

LISP-0: Remote EID IID 100 prefix 172.16.100.102/32, Send map request (1)
(sources: <signal>, state: incomplete, rlocs: 0).

LISP-0: EID-AF IPv4, Sending map-request from 172.16.100.102 to 172.16.100.102
for EID 172.16.100.102/32, ITR-RLOCs 1, nonce 0x23AC364E-0x392EF091 (encap src
192.168.1.1, dst 192.168.111.111).
```

Debug 4-1: *Encapsulated Control message sent by ITR Edge-1 to Map-Resolver.*

Request Phase 3: Map-Reply by MR

When Map-Resolver receives the Encapsulated Control Message carrying Map-Request message, it decapsulates the message. Then it checks if it has information about requested EID in its database.

Because the E2xTR (Edge-2) has registered EID 172.16.100.102 with Map-Register message and with P-bit (Proxy-Map-Reply) the MS is able to answer the Map-Request on behalf of E2xTR. Map-Resolver sends an EID-to-RLOC mapping information to requesting ITR E1xTR using the same “nonce” value that was used in Map-Request message.

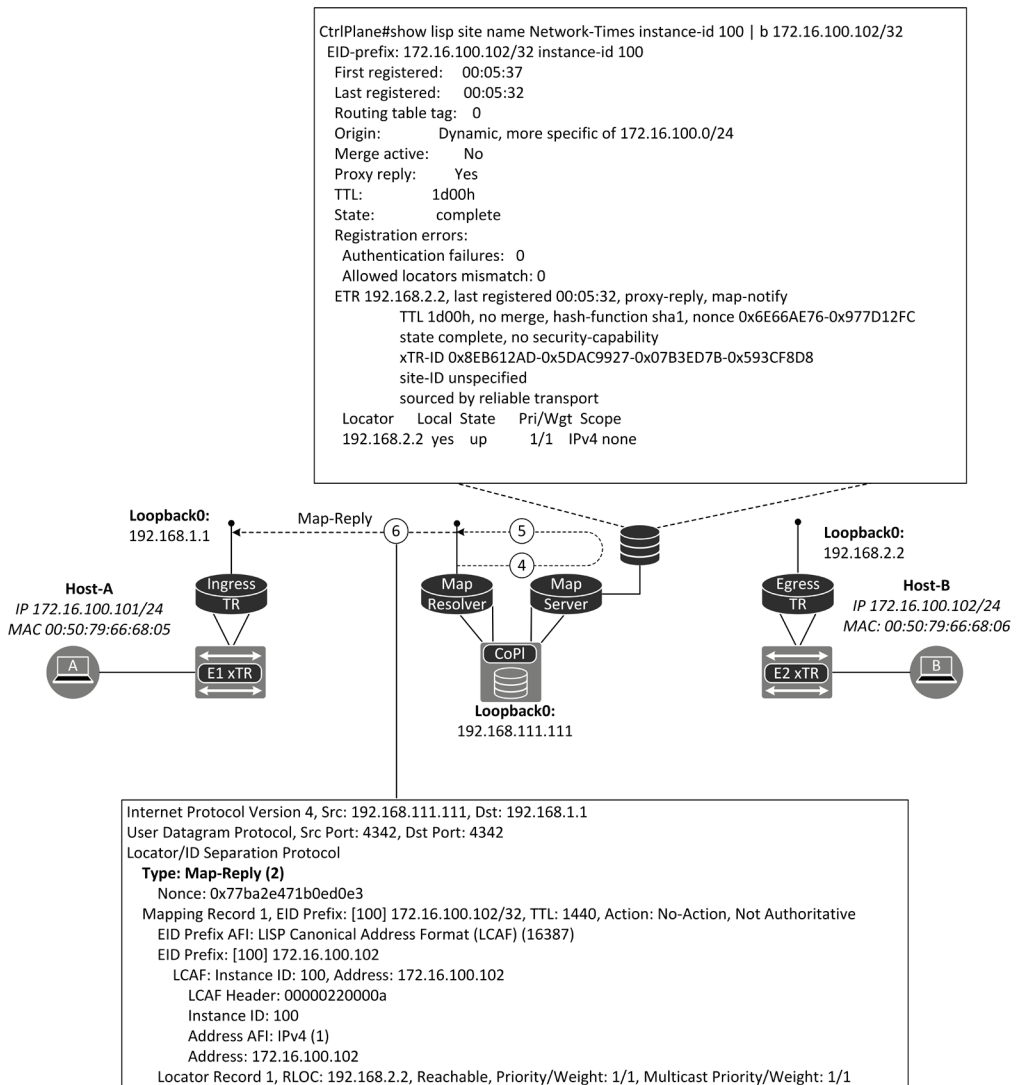


Figure 4-3: Map-Reply by Control-Plane node.

```

Ethernet II, Src: 50:00:00:04:00:00 (50:00:00:04:00:00), Dst: 50:00:00:01:00:02
(50:00:00:01:00:02)
Internet Protocol Version 4, Src: 192.168.111.111, Dst: 192.168.1.1
User Datagram Protocol, Src Port: 4342, Dst Port: 4342
Locator/ID Separation Protocol
  0010 .... = Type: Map-Reply (2)
  .... 0... = P bit (Probe): Not set
  .... .0.. = E bit (Echo-Nonce locator reachability
algorithm enabled): Not set
  .... ..0. = S bit (LISP-SEC capable): Not set
  .... ...0 0000 0000 0000 0000 = Reserved bits: 0x000000
Record Count: 1
Nonce: 0x77ba2e471b0ed0e3
Mapping Record 1, EID Prefix: [100] 172.16.100.102/32, TTL: 1440, Action:
No-Action, Not Authoritative
  Record TTL: 1440
  Locator Count: 1
  EID Mask Length: 32
  000. .... = Action: No-Action (0)
  ...0 .... = Authoritative bit: Not set
  .... .000 0000 0000 = Reserved: 0x0000
  0000 .... = Reserved: 0x0
  .... 0000 0000 0000 = Mapping Version: 0
  EID Prefix AFI: LISP Canonical Address Format (LCAF) (16387)
  EID Prefix: [100] 172.16.100.102
    LCAF: Instance ID: 100, Address: 172.16.100.102
      LCAF Header: 00000220000a
        Reserved bits: 0x00
        Flags: 0x00
        Type: Instance ID (2)
        Reserved bits: 0x20
        Length: 10
        Instance ID: 100
        Address AFI: IPv4 (1)
        Address: 172.16.100.102
    Locator Record 1, RLOC: 192.168.2.2, Reachable, Priority/Weight: 1/1,
Multicast Priority/Weight: 1/1
  Priority: 1
  Weight: 1
  Multicast Priority: 1
  Multicast Weight: 1
  Flags: 0x0001
    0000 0000 0000 0... = Reserved: 0x0000
    .... .0.. = Local: Not set
    .... ..0. = Probe: Not set
    .... ...1 = Reachable: Set
  AFI: IPv4 (1)
  Locator: 192.168.2.2

```

Capture 4-2: LISP Registration: Map-Reply from Map-Resolver.

Debug 4-2 shows how Map-Resolver receives the LISP Encapsulated Control Message carrying Map-Request from E1xTR and how it answers to E1xTR on behalf of E2xTR 192.168.2.2

```
LISP: Processing received Encap-Control(8) message on GigabitEthernet1 from
192.168.1.1:4342 to 192.168.111.111:4342

LISP: Processing received Map-Request(1) message on GigabitEthernet1 from
172.16.100.102:4342 to 172.16.100.102:4342

LISP: Received map request for IID 100 172.16.100.102/32, source_eid IID 100
172.16.100.101, ITR-RLOCs: 192.168.1.1, records 1, nonce 0x23AC364E-0x392EF091

LISP-0: MS EID IID 100 prefix 172.16.100.102/32 SVC_IP_IAF_IPv4 site Network-
Times, Sending proxy reply to 192.168.1.1.
```

Debug 4-2: *Map-Resolver processing ECM with Map-Request receive from ITR Edge-1.*

Request Phase 4: Map-Request probe by ITR

When the E1xTR receives the LISP Map-Reply message from Map-Resolver, it stores the EID-to-RLOC information into *Local Map-Cache* with TTL 24 hours. Before forwarding the ICMP-Request sent by Host-A, E1xTR sends a Map-Request message with P-bit (Probe) set to RLOC 192.168.2.2. The P-bit indicates that this Map-Request is used as an RLOC reachability probe.

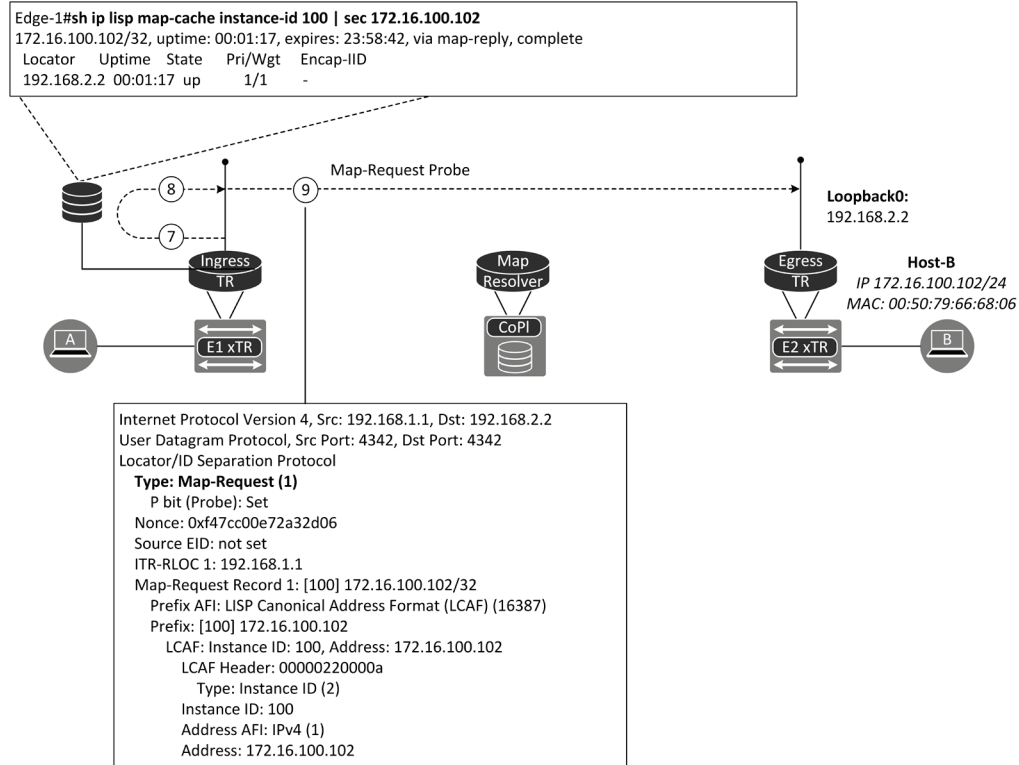


Figure 4-4: *Map-RequestProbe* from E1xTR to E2xTR.

```

Ethernet II, Src: 50:00:00:01:00:01 (50:00:00:01:00:01), Dst: 50:00:00:03:00:00
(50:00:00:03:00:00)
Internet Protocol Version 4, Src: 192.168.1.1, Dst: 192.168.2.2
User Datagram Protocol, Src Port: 4342, Dst Port: 4342
Locator/ID Separation Protocol
    0001 .... = Type: Map-Request (1)
    .... 0010 00.. .... = Flags: 0x08
        .... 0... .... = A bit (Authoritative): Not set
        .... .0.. .... = M bit (Map-Reply present): Not set
        .... ..1. .... = P bit (Probe): Set
        .... ...0 .... = S bit (Solicit-Map-Request): Not set
        .... .... 0... .... = p bit (Proxy ITR): Not set
        .... .... .0.. .... = s bit (SMR-invoked): Not set
    .... .... ..00 0000 000. .... = Reserved bits: 0x0000
    .... .... .... ..0 0000 = ITR-RLOC Count: 0
Record Count: 1
Nonce: 0xf47cc00e72a32d06
Source EID AFI: Reserved (0)

```

```

Source EID: not set
ITR-RLOC 1: 192.168.1.1
    ITR-RLOC AFI: IPv4 (1)
    ITR-RLOC Address: 192.168.1.1
Map-Request Record 1: [100] 172.16.100.102/32
    Reserved: 0x00
    Prefix Length: 32
    Prefix AFI: LISP Canonical Address Format (LCAF) (16387)
    Prefix: [100] 172.16.100.102
        LCAF: Instance ID: 100, Address: 172.16.100.102
            LCAF Header: 00000220000a
                Reserved bits: 0x00
                Flags: 0x00
                Type: Instance ID (2)
                Reserved bits: 0x20
                Length: 10
            Instance ID: 100
            Address AFI: IPv4 (1)
            Address: 172.16.100.102

```

Capture 4-3: *Map-request from E1xTR.*

Debug 4-3 shows how ITR receives Map-Reply from Map-Resolver. It verifies the reachability of RLOC for 172.17.100.102 and stores the mapping information into Mapping-cache. Then it generates the Map-Request with Probe-bit set to RLOC 192.168.2.2 to verify that it is actually live. This check is done in addition to the RIB check.

```

LISP: Processing received Map-Reply(2) message on GigabitEthernet1 from
192.168.111.111:4342 to 192.168.1.1:4342

LISP: Received map reply nonce 0x23AC364E-0x392EF091, records 1

LISP: Processing Map-Reply mapping record for IID 100 172.16.100.102/32 LCAF 2,
ttl 1440, action none, not authoritative, 1 locator
    192.168.2.2 pri/wei=1/1 lpR

LISP-0: Map Request IID 100 prefix 172.16.100.102/32 remote EID prefix[LL],
Received reply with rtt 5ms.

LISP: Processing mapping information for EID prefix IID 100 172.16.100.102/32

LISP-0: Remote EID IID 100 prefix 172.16.100.102/32, Change state to reused
(sources: <map-rep>, state: incomplete, rlocs: 0).

LISP-0: Remote EID IID 100 prefix 172.16.100.102/32, Starting idle timer (delay
00:02:30) (sources: <map-rep>, state: reused, rlocs: 0).

LISP-0: IAF IID 100 SVC_IP_IAF_IPv4, Persistent db: ignore writing request,
disabled.

```

```
LISP-0: Remote EID IID 100 prefix 172.16.100.102/32, Change state to complete
(sources: <map-rep>, state: reused, rlocs: 0).

LISP: RIB Watch Group default 192.168.2.2/32 , created.

LISP: RIB Watch Group default 192.168.2.2/32 , scheduling RIB update.

LISP-0: Remote EID IID 100 prefix 172.16.100.102/32, RLOCs pending rwatch
update, defer fwd update (sources: <map-rep>, state: complete, rlocs: 0).

LISP-0: Remote EID IID 100 prefix 172.16.100.102/32, 1 RLOCs pending rwatch
update, defer fwd update (sources: <map-rep>, state: complete, rlocs: 0).

LISP-0: Remote EID IID 100 prefix 172.16.100.102/32, Recalculated RLOC status
bits from 0x0 to 0x1 (sources: <map-rep>, state: complete, rlocs: 1).

LISP-0: Remote EID IID 100 prefix 172.16.100.102/32, 1 RLOCs pending rwatch
update, defer fwd update (sources: <map-rep>, state: complete, rlocs: 1).

LISP: RIB Watch Group default 192.168.2.2/32 , installing in RIB.

LISP-0: Remote shrRLOC 192.168.2.2, Reachability notification, up* allow*
remote.

LISP-0: Remote EID IID 100 prefix 172.16.100.102/32, No more RLOCs pending
rwatch update, schedule deferred fwd update (sources: <map-rep>, state:
complete, rlocs: 1).

LISP-0: Remote EID IID 100 prefix 172.16.100.102/32, Extranet IID passed to CEF
is 16777216 (sources: <map-rep>, state: complete, rlocs: 1).

LISP-0: IID 100 Request processing of remote EID RLOC map requests to IPv4.

LISP: Send map request type remote EID RLOC

LISP: Send map request for EID prefix IID 100 172.16.100.102/32

LISP-0: Remote EID IID 100 prefix 172.16.100.102/32 [RRLOCset 0x7FC23C714DA0
1/1] 192.168.2.2 pri/wei=1/1, Send RLOC probe (sources: <map-rep>, state:
complete, rlocs: 1).

LISP-0: EID-AF IPv4, Sending probe map-request from 192.168.1.1 to 192.168.2.2
for EID 172.16.100.102/32, ITR-RLOCs 1, nonce 0xE21F74F8-0xABF03C00.
```

Debug 4-3: *E1xTR receiving Map-Reply from Map-Resolver.*

Request Phase 5: Map-Reply Probe by RLOC

When the E2xTR with RLOC 192.168.2.2 receives the Map-Request with a P-bit, it responds to a message with an LSIP Map-Reply message with a P-bit indicating that this is the Map-Reply probe. In addition, E2xTR copies the “nonce” value from Map-Request Probe to Map-Reply message. When E1xTR receives the Map-Reply Probe, it starts sending ICMP-Requests from Host-A to Host-B.

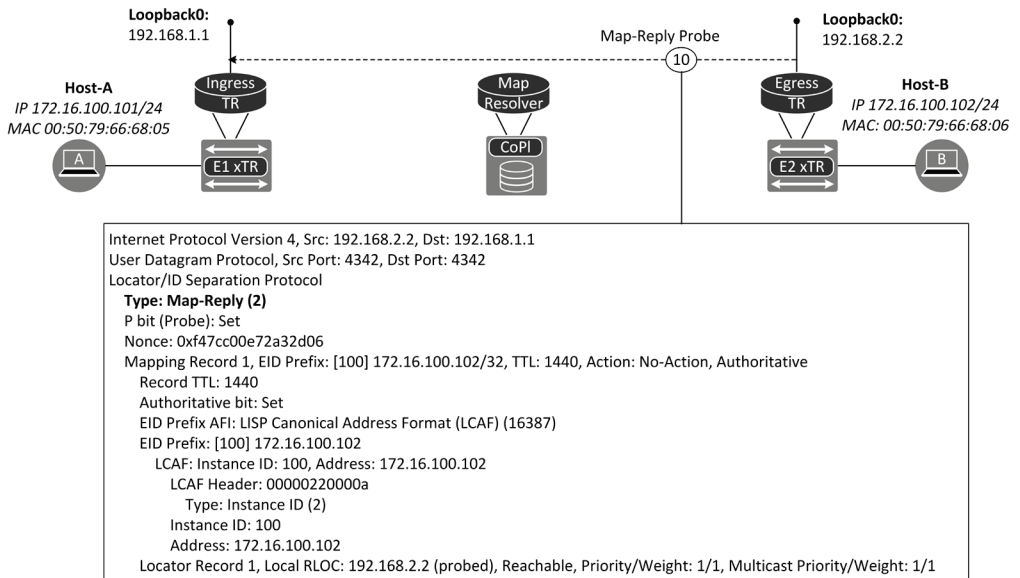


Figure 4-5: Map-Reply Probe from E2xTR to E1xTR.

Capture 4-4 shows the captured packet.

```
Internet Protocol Version 4, Src: 192.168.2.2, Dst: 192.168.1.1
User Datagram Protocol, Src Port: 4342, Dst Port: 4342
Locator/ID Separation Protocol
  0010 .... = Type: Map-Reply (2)
  .... 1... = P bit (Probe): Set
  .... .0.. = E bit (Echo-Nonce locator reachability
algorithm enabled): Not set
  .... ..0. = S bit (LISP-SEC capable): Not set
  .... ...0 0000 0000 0000 0000 = Reserved bits: 0x00000
Record Count: 1
Nonce: 0xf47cc00e72a32d06
Mapping Record 1, EID Prefix: [100] 172.16.100.102/32, TTL: 1440, Action:
No-Action, Authoritative
  Record TTL: 1440
  Locator Count: 1
  EID Mask Length: 32
  000. .... = Action: No-Action (0)
  ...1 .... = Authoritative bit: Set
  .... .000 0000 0000 = Reserved: 0x000
  0000 .... = Reserved: 0x0
  .... 0000 0000 0000 = Mapping Version: 0
  EID Prefix AFI: LISP Canonical Address Format (LCAF) (16387)
  EID Prefix: [100] 172.16.100.102
    LCAF: Instance ID: 100, Address: 172.16.100.102
      LCAF Header: 00000220000a
        Reserved bits: 0x00
        Flags: 0x00
        Type: Instance ID (2)
        Reserved bits: 0x20
        Length: 10
        Instance ID: 100
        Address AFI: IPv4 (1)
        Address: 172.16.100.102
      Locator Record 1, Local RLOC: 192.168.2.2 (probed), Reachable,
Priority/Weight: 1/1, Multicast Priority/Weight: 1/1
    Priority: 1
    Weight: 1
    Multicast Priority: 1
    Multicast Weight: 1
    Flags: 0x0007
    AFI: IPv4 (1)
    Locator: 192.168.2.2
```

Capture 4-4: Map-Reply Probe from E2xTR.

Debug 4-4 shows how E2xTR receives the probe from E1xTR and replies to it by sending Map-Reply with the Probe-bit set.

```
LISP: Processing received Map-Request(1) message on GigabitEthernet1 from
192.168.1.1:4342 to 192.168.2.2:4342

LISP: Received map request for IID 100 172.16.100.102/32, source_eid UNSPEC,
ITR-RLOCs: 192.168.1.1, records 1, nonce 0xE21F74F8-0xABF03C00, probe

LISP: Processing map request record for EID prefix IID 100 172.16.100.102/32

LISP-0: Sending map-reply from 192.168.2.2 to 192.168.1.1.
```

Debug 4-4: *E2xTR receives a Probe from E1xTR.*

Debug 4-5 shows how E1xTR receives the Probe message.

```
LISP: Processing received Map-Reply(2) message on GigabitEthernet1 from
192.168.2.2:4342 to 192.168.1.1:4342

LISP: Received map reply nonce 0xE21F74F8-0xABF03C00, records 1, probe

LISP: Processing Map-Reply mapping record for IID 100 172.16.100.102/32 LCAF 2,
ttl 1440, action none, authoritative, 1 locator

192.168.2.2 pri/wei=1/1 LPR
```

Debug 4-5: *E1xTR receives a Probe from E2xTR.*

Map-Request Process Summary: Phases 1-5

This chapter describes the Map-Request/Map-Reply processes.

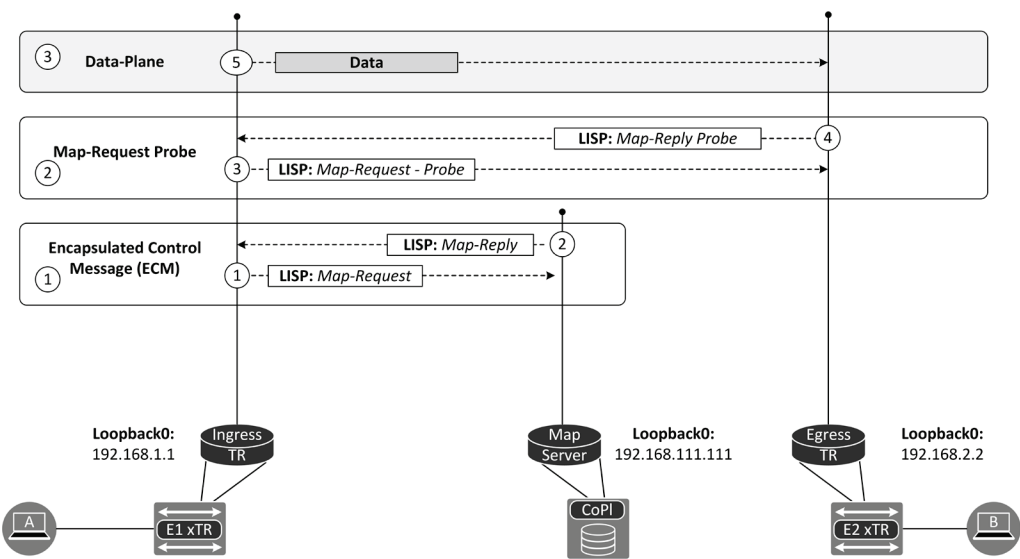


Figure 4-6: Summary of Map-Request process.

References

- [RFC 6830] D. Farinacci et al., "The Locator/ID Separation Protocol (LISP)", RFC 6830, January 2013.
- [RFC 6833] V. Fuller and D. Farinacci., "Locator/ID Separation Protocol (LISP) Map-Server Interface", RFC 6833, January 2013.
- [LISP Control Plane] D. Farinacci et al., "The Locator/ID Separation Protocol (LISP) Control Plane", draft-ietf-lisp-rdc6833bis-25, June 16, 2019.
- [LISP Rel-Tran] C. Cassar et al., "LISP Reliable Transport", draft-kouvelas-lisp-reliable-transport-02, March 26, 2015.

Chapter 5: Data Plane: VXLAN Encapsulation

VXLAN Data-Plane operation

This chapter explains the VXLAN Data-Plane in LISP Campus Fabric. Figure 5-1 illustrates the situation where Host-A 172.16.100.101 sends an ICMP-echo to Host-B 172.16.100.102. Host-A does not know the Mac address of Host-B and it sends an ARP-Request. E1xTR is a gateway for BDI100 and it is allowed to answer ARP-Request messages concerning hosts connected remote ETRs. This example assumes that E1xTR has already received Map-Reply about Host-B EID-to-RLOC mapping information and stored it into LISP Mapping-Cache.

- Phase 1.** Host-A sends an ARP-Request to find the MAC-IP address binding information of Host-B. The message is sent as a Layer 2 broadcast.
- Phase 2.** E1xTR receives the ARP-Request. Due to **ip local-proxy** definition under interface BDI100, it is able to answer using its own Anycast-MAC address on behalf of Host-B.
- Phase 3.** Host-A starts sending ICMP-Echos to Host-B.
- Phase A.** Interface BDI100 in E1xTR belongs to the VRF 100_White and it is attached to LISP Mobility Group **172_16_100_0-100_White**.
- Phase B.** Dynamic-EID table **172_16_100_0-100_White**, in turn, belongs to LISP Instance 100. Network 172.16.100.0/24 is mapped into this EID-table and IPv4 packets to destination 172.16.100.xx/24 sent out of E1xTR will use the Loopback 0 as a source IP address (defined in RLOC-SET1) in the outer IP header.
- Phase C.** The IPv4 packets belonging to LISP Instance 100 and sent by E1xTR will be encapsulated with VXLAN tunnel headers.
- Phase D.** E1xTR has an EID-to-RLOC mapping information in its Mapping-Cache learned via Map-Reply message from Map-Server. The destination RLOC is 192.168.2.2

Phase E. E1xTR local routing table includes the route to RLOC 192.168.2.2. Based on this information it is able to build VXLAN tunnel headers and forward packet towards the destination.

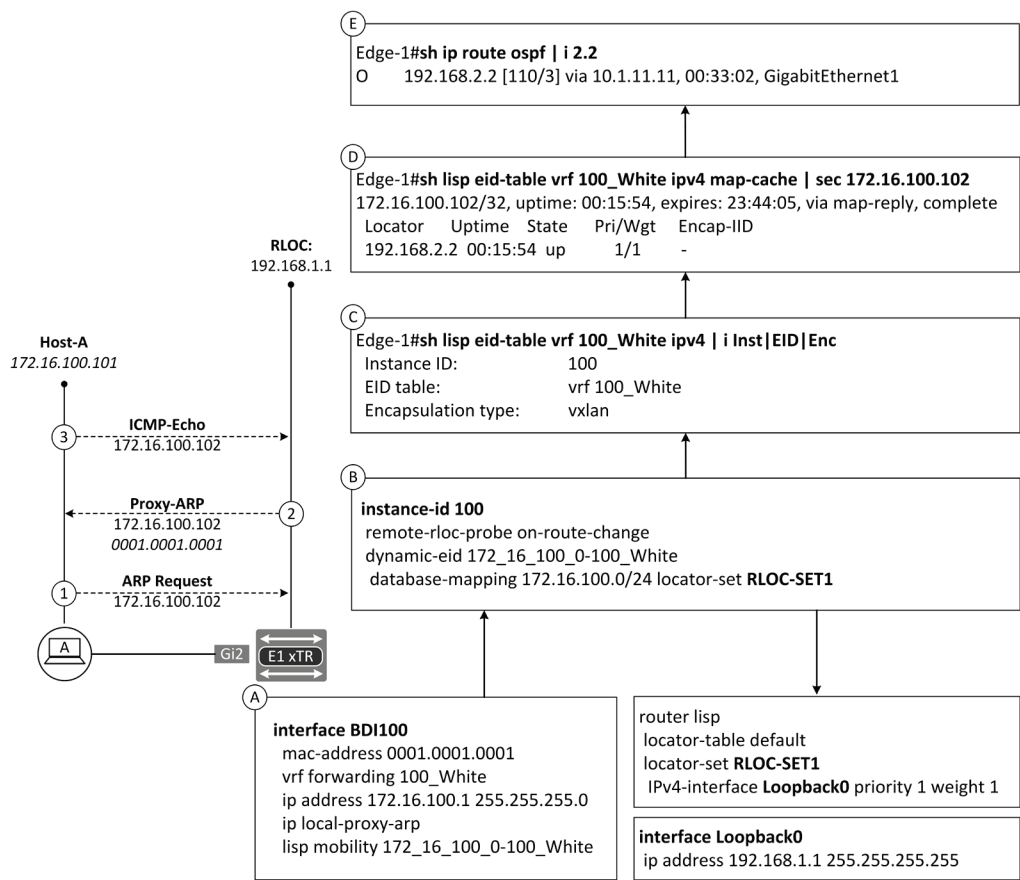


Figure 5-1: Data-Plane operation on the Ingress Tunnel Router E1xTR.

Figure 5-2 illustrates how E1xTR constructs the VXLAN encapsulated packet. The source IP address in outer IP-Header is its own RLOC address 192.168.1.1. The destination IP address is taken from the EID-to-RLOC Mapping-Cache entry that defines the RLOC used for 172.16.100.102/32. VXLAN is MAC-in-UDP/IP encapsulation where the destination UDP port is 4789. The original Virtual Extensible LAN (VXLAN) Data-Plane encapsulation defined in RFC 7348 does not carry the information about carried protocol and it only supports Ethernet frame encapsulation. The *Generic Protocol Extension for VXLAN (VXLAN-GPE)*

extends the original VXLAN header with protocol type field, OAM protocol used, and the VXLAN GPE version. IOS-XE based Campus Fabric solution use VXLAN-GPE extension in Data-Plane. Flag bits in the VXLAN-GPE header defines that there is an additional VXLAN-GPE sub-header called Group-Based Policy (VXLAN-GBP).

The *D-bit* (*Don't learn bit*) in VXLAN-GBP states that the receiving ETR must not learn the source address defined in the encapsulated frame.

The *I-Bit* (*Instance bit*) indicates that the Virtual Network-Id (VNI) is carried within the sub-header. This bit points to *VXLAN Network Identifier (VNI)* field which carries the LISP instance-Id for EID.

The inner Ethernet header source MAC address is the MAC address of E1xTR and the destination MAC address is Cisco specific “dummy” MAC address that is not owned by any device. The inner IP header carries the original ICMP-Request message IP addresses.

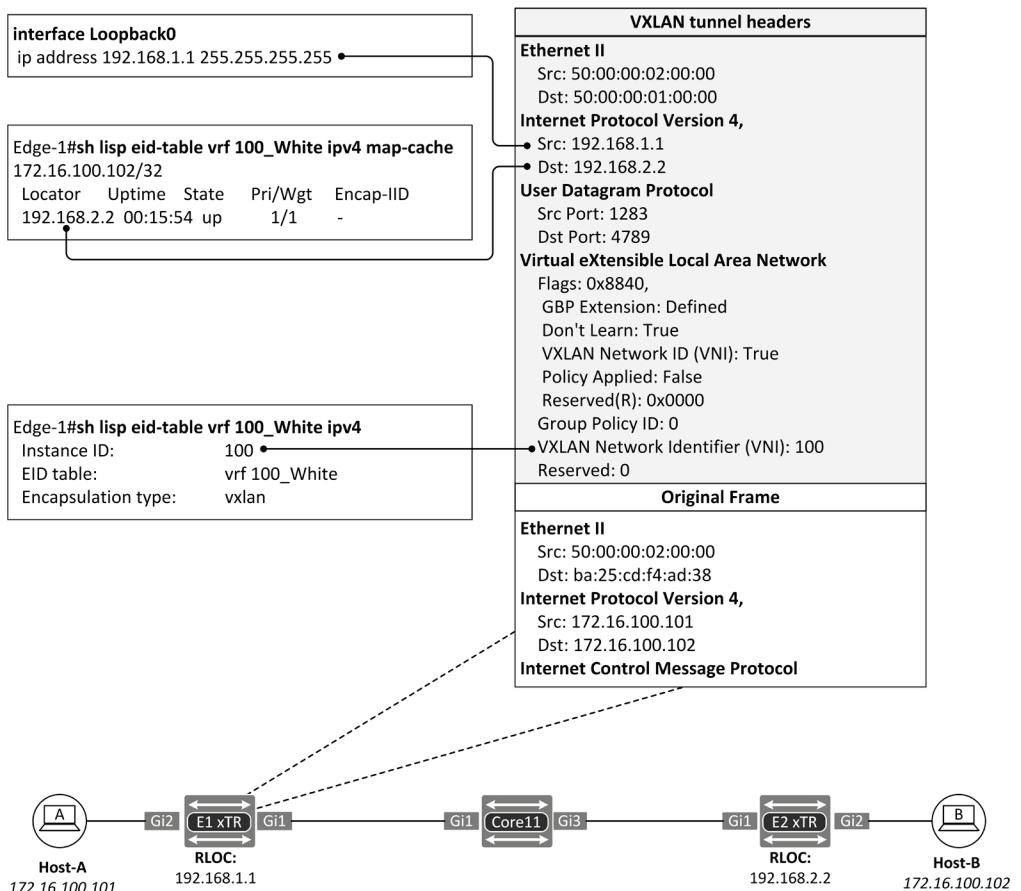


Figure 5-2: Data-Plane operation on the Ingress Tunnel Router E1xTR continues.

Core router Core11 receives the IP packet from E1xTR. It makes a routing decision based on the destination IP address of the outer IP header and forwards the packet to E2xTR.

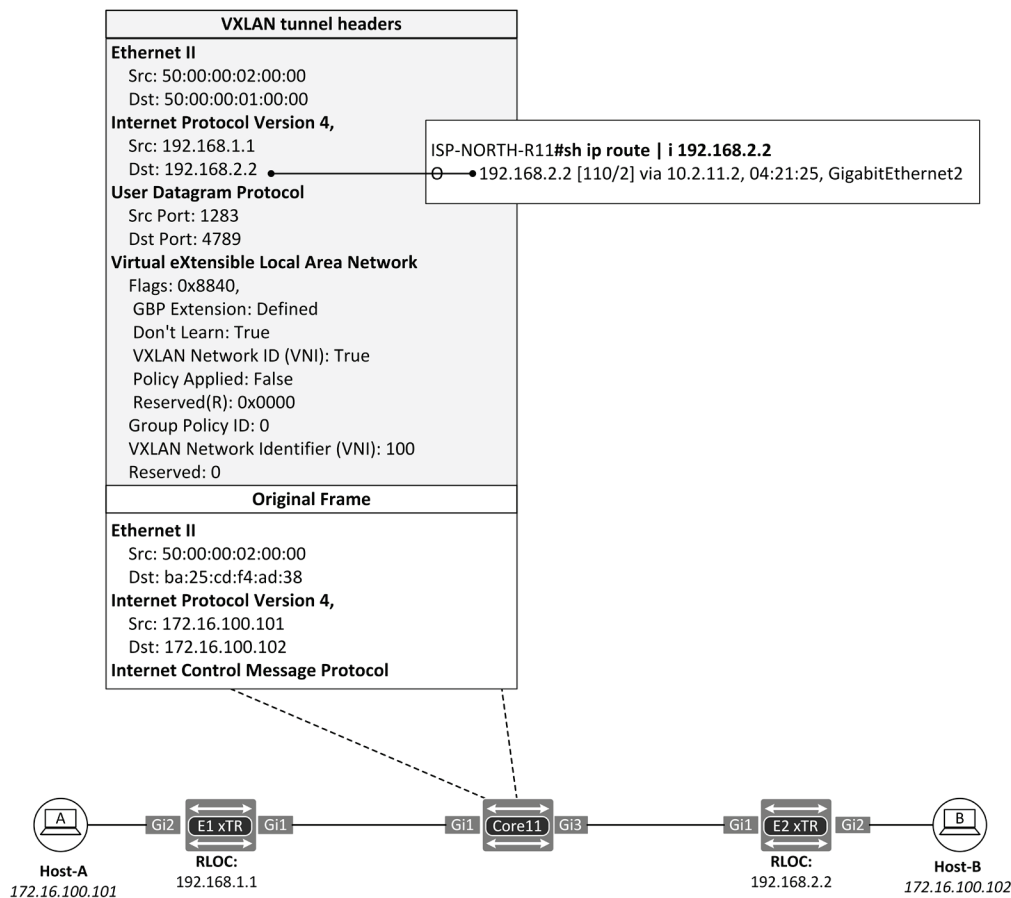


Figure 5-3: Data-Plane operation on the Core router Core11.

E2xTR receives the IP packet originated by E1xTR. It notices, based on destination UDP port 4789, that this packet is VXLAN encapsulated. Based on the VXLAN Network Identifier, E2xTR knows that this packet belongs to VRF 100_White. The destination IP address belongs to VRF 100_White and it is connected into Interface GigabitEthernet2.EFP100. E2xTR sends the original ICMP-Request to Host-B using BDI100 Anycast-MAC address 0001.0001.0001 as a source address. Note that because of “don’t learn bit”, E2xTR doesn’t learn the source EID from the packet.

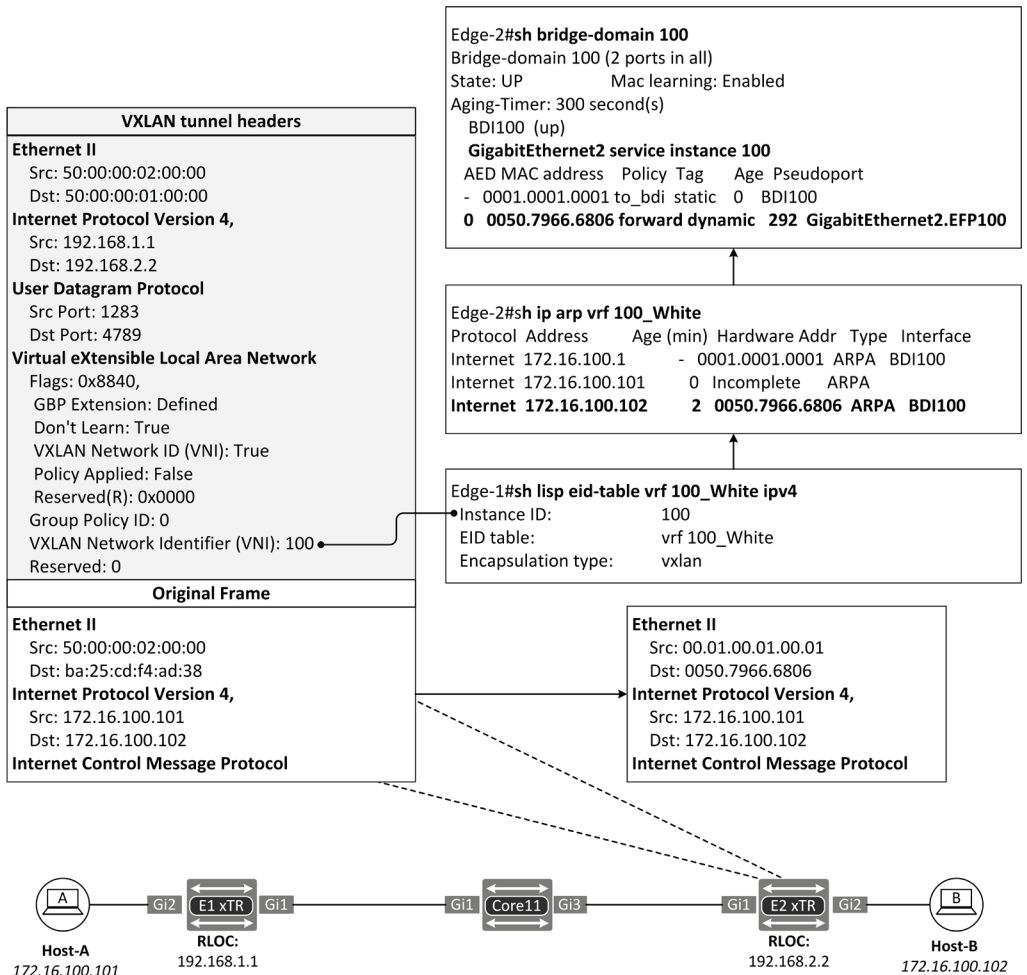


Figure 5-4: Data-Plane operation on Egress Tunnel Router E2xTR.

When Host-B receives the ICMP-Echo it replies by sending an ICMP-Reply. The previously described operation is repeated in reversed order.

SGT Information in Data-Plane

The Micro-Segmentation in Campus Fabric can be done by using Scalable Group Tag (SGT) as a user/group/machine/segment identifier. The SGT information in the SD-Access solution is pushed from the Cisco Identity Service Engine (ISE). This example, however, uses static SGT mode. Note that the configuration is done under the Layer 3 interface, which means that all routed traffic will be tagged with SGT. Figure 5-5 illustrates the VXLAN encapsulated ICMP-Request message from Host-A to Host-B and it shows how the SGT information configured under the BD100 is carried as Group Policy Id in the VXLAN header.

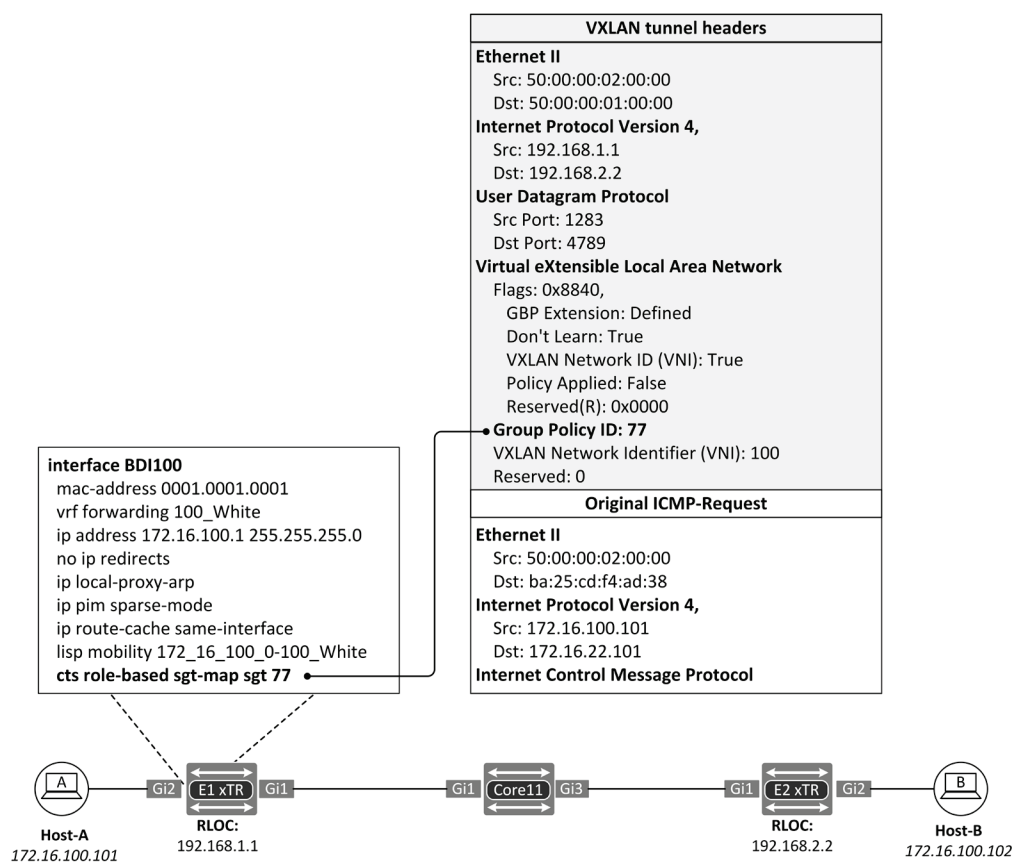


Figure 5-5: Scalable Group Tag in VXLAN Header.

References

- [RFC 7438] M. Mahalingam et al., "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", RFC 7438, August 2014.
- [VXLAN-GBP] J. Lemon et al., "Group Policy Encoding with VXLAN GPE", draft-ietf-vxlan-gpe-gbp-02, March 2, 2019.
- [VXLAN-GPE] F. Maino et al., "Generic Protocol Extension for VXLAN", draft-ietf-nvo3-vxlan-gpe-09. December 5, 2019

Chapter 6: LISP Control-Plane Security

The Basic LISP Control-Plane Security

The basic LISP security mechanism has two main functions. First, Map-Server validates EID-to-RLOC mapping information sent by an ETR before installing it into a database. ETR and Mapping-Server use the same Shared Secret Key that is used for calculating authentication data value carried within the Map-Register message. In the case of rogue ETR, the authentication data differs and Map-Server rejects the message. In addition, the EID carried within the Map-Register message has to be found from the Instance specific EID-to-RLOC entry. As a last check, Map-Server verifies that the nonce value is higher than what was recently received from this particular ETR.

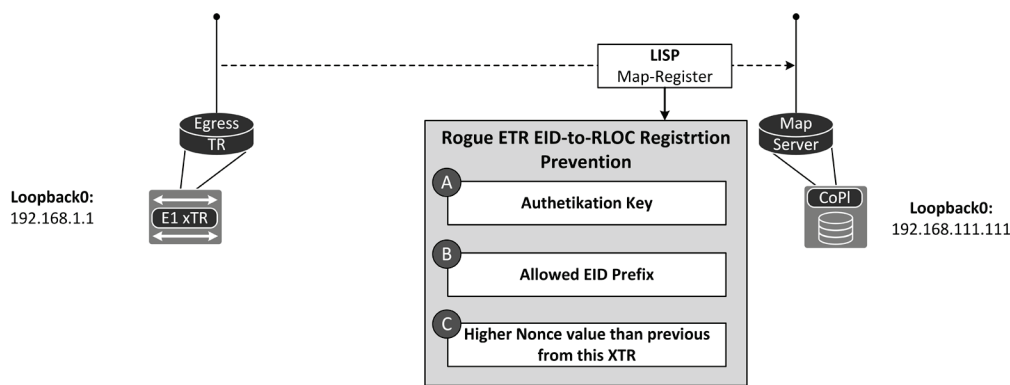


Figure 6-1: Map-Register Integrity and Authentication process by Map-Server.

The ITR, in turn, validates the Map-Reply message by comparing the nonce value received in Map-Reply to the value used in Map-Request message.

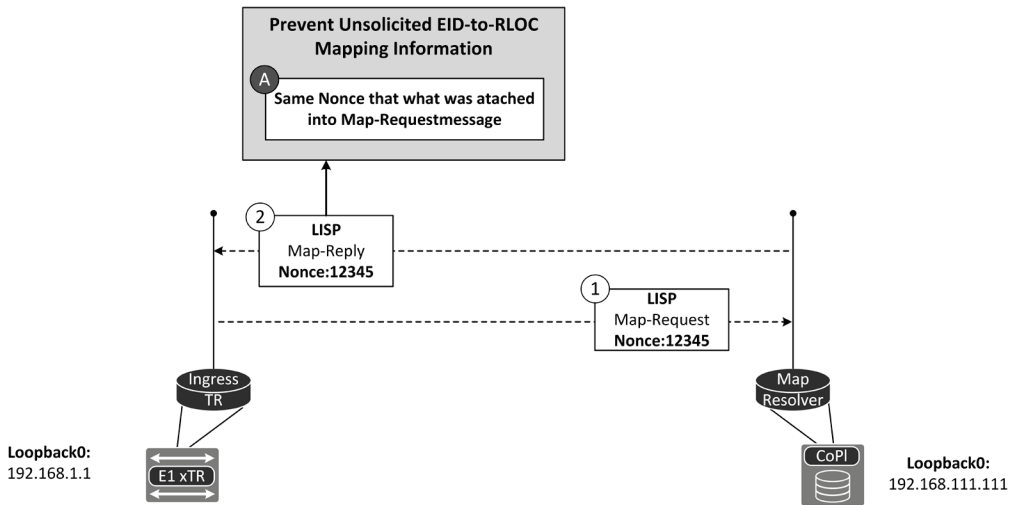


Figure 6-2: Map-Reply Integrity integrity check by ITR.

Map-Register validation process

When an ETR constructs a Map-Register message, it calculates the *Message Authentication Code* (MAC) from the Map-Register message payload (Authentication Data value is set to zero) along with a shared secret-key “nwktimes”. It uses SHA-1 cryptographic hash-function that splits data into 512-bit blocks and then iteratively compresses these blocks resulting 160-bits (20-Bytes) output. This process is called the *Hash-based Message Authentication Code* (HMAC). The process, where the SHA-1 cryptographic hash algorithm along with a shared secret-key is used to provide MAC, is called HMAC-SHA-1. The resulting MAC is used as Authentication Data. The Shared Secret key is identified in the Key-Id field.

In addition, ETR generates the per-message nonce value which is always greater than its previously sent LISP Control-Plane message. Each Map-Register also includes information about sending ETR in the xTR-Id field.

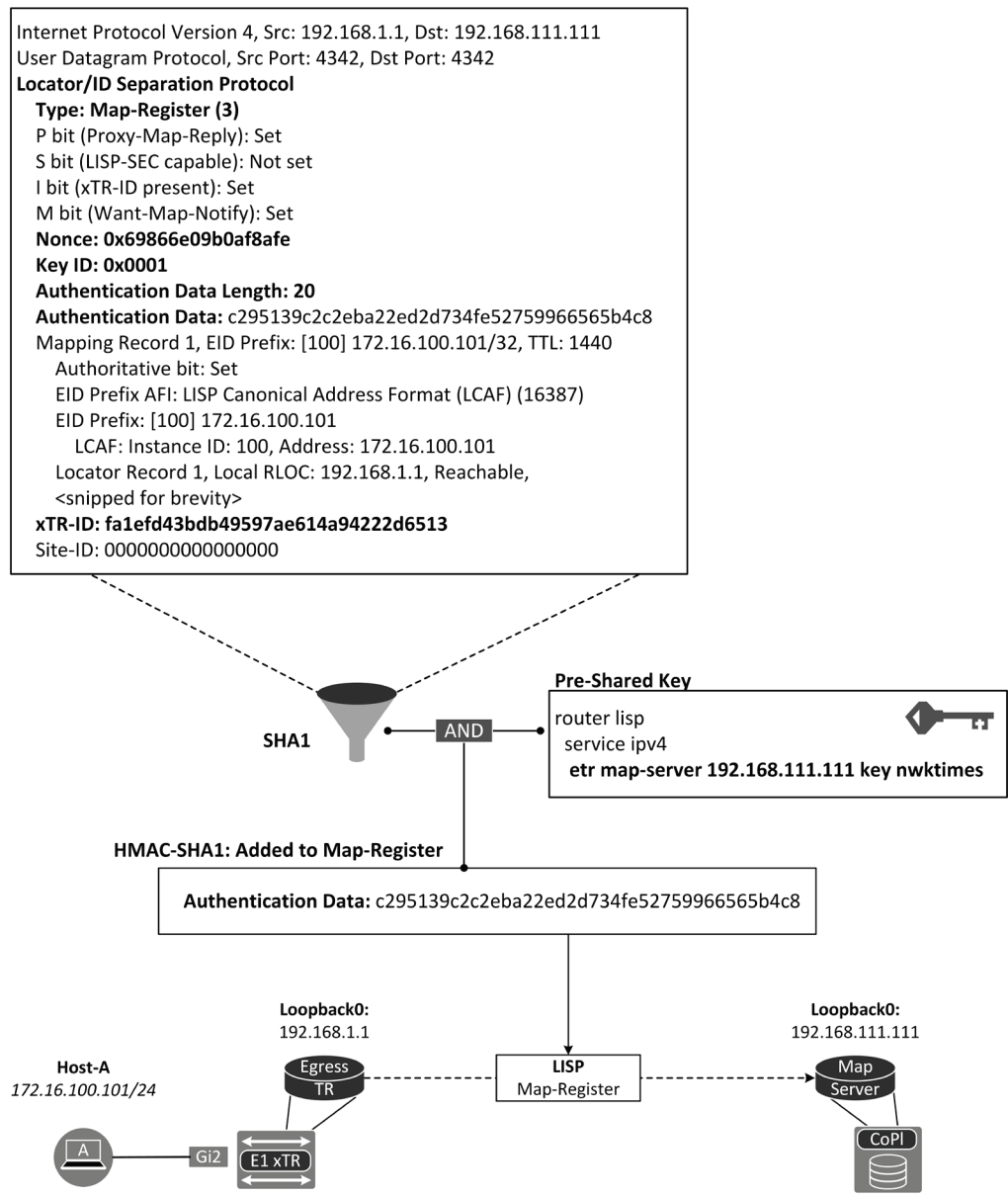


Figure 6-3: Map-Register processing of ETR Edge-1.

When Map-Server receives the Map-Register message from ETR, it validates the message before installing EID-to-RLOC mapping information into the Mapping database. It runs the same keyed Hash-Based Message Authentication process using the key-id defined in the message. It compares the output and if they are equal the authentication succeeds. Map-Server also verifies that the nonce value is greater than the last nonce value received from this xTR-Id. The last check is to verify that the Instance-Id 100 includes the EID-Prefix where the registered EID belongs to. If all these validations succeed, the Map-Server installs the EID-to-RLOC information into the Mapping database.

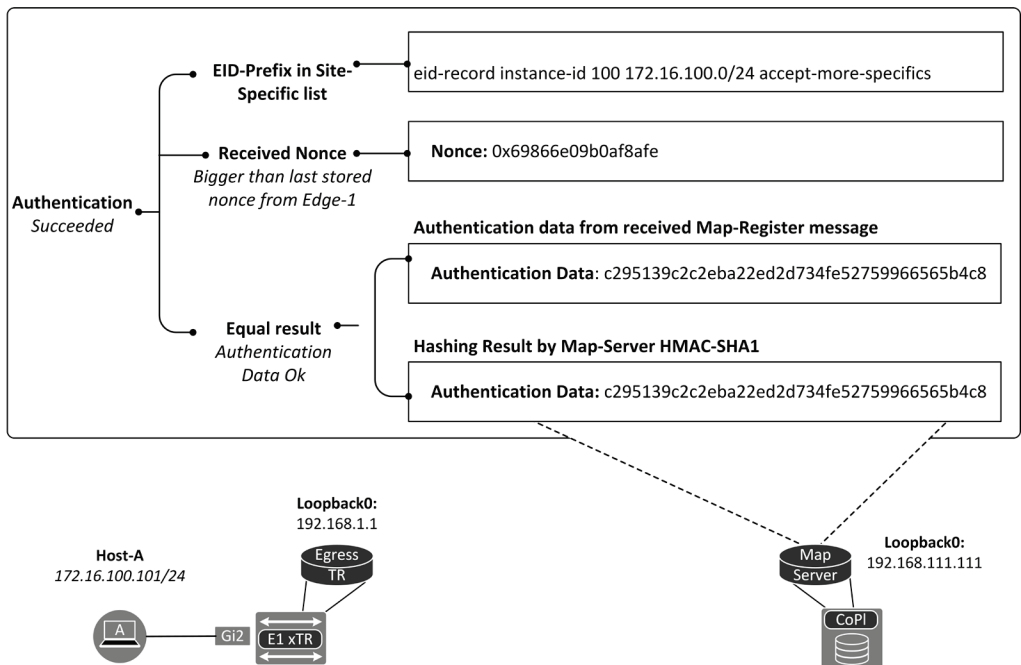


Figure 6-4: Integrity and Authentication process against Map-Register by Map-Server.

Figure 6-5 illustrates the EID-to-RLOC mapping database entry that describes the EID 172.16.100.101/32.

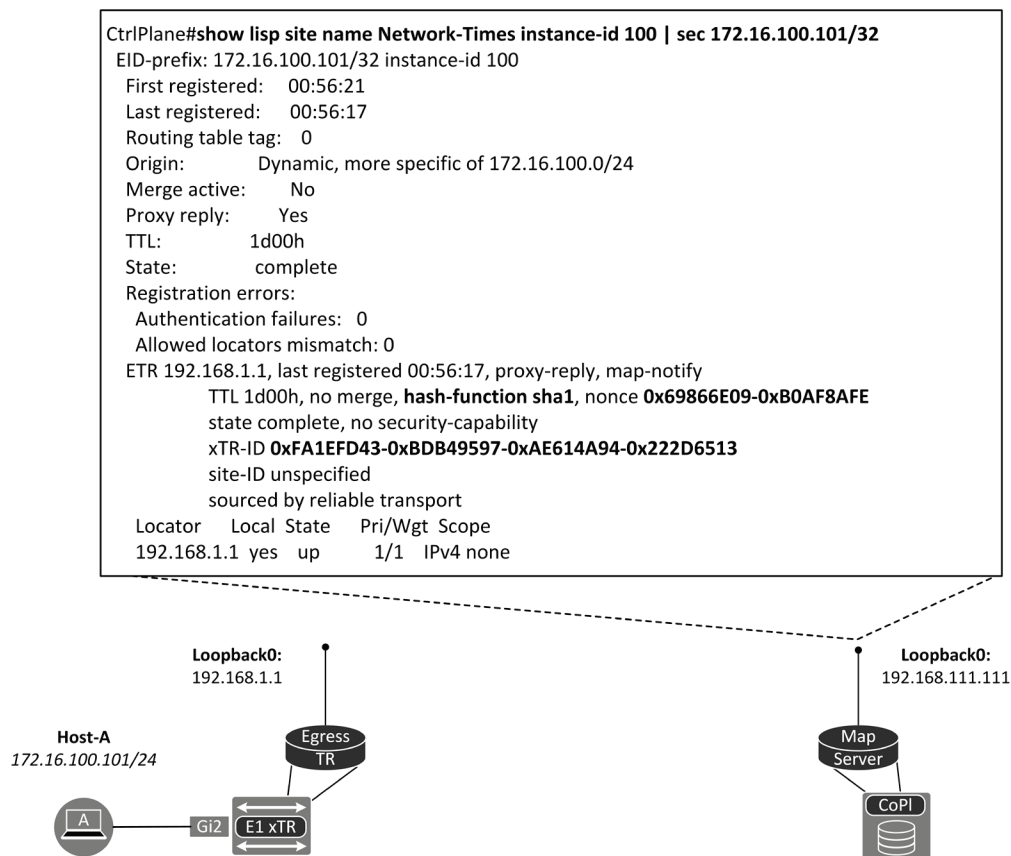


Figure 6-5: show lisp site name Network-Times instance-id 100 | sec 172.16.100.101/32

Map-Reply validation process

The basic security processes used by Ingress Tunnel Router is just an integrity check. When an ITR sends an EID-to-RLOC Map-Request message to the Map-Resolver, it generates a nonce value in the same way that what its ETR “role” did when it sent a Map-Register message to the Map-Server. When Map-Resolver constructs a Map-Reply message, it copies the nonce value from received Map-Request message into its Map-Reply. When ITR receives the Map-Reply, it compares the original nonce to received nonce. If values are equal, the integrity checks succeed. Figure 6-6 illustrates this process.

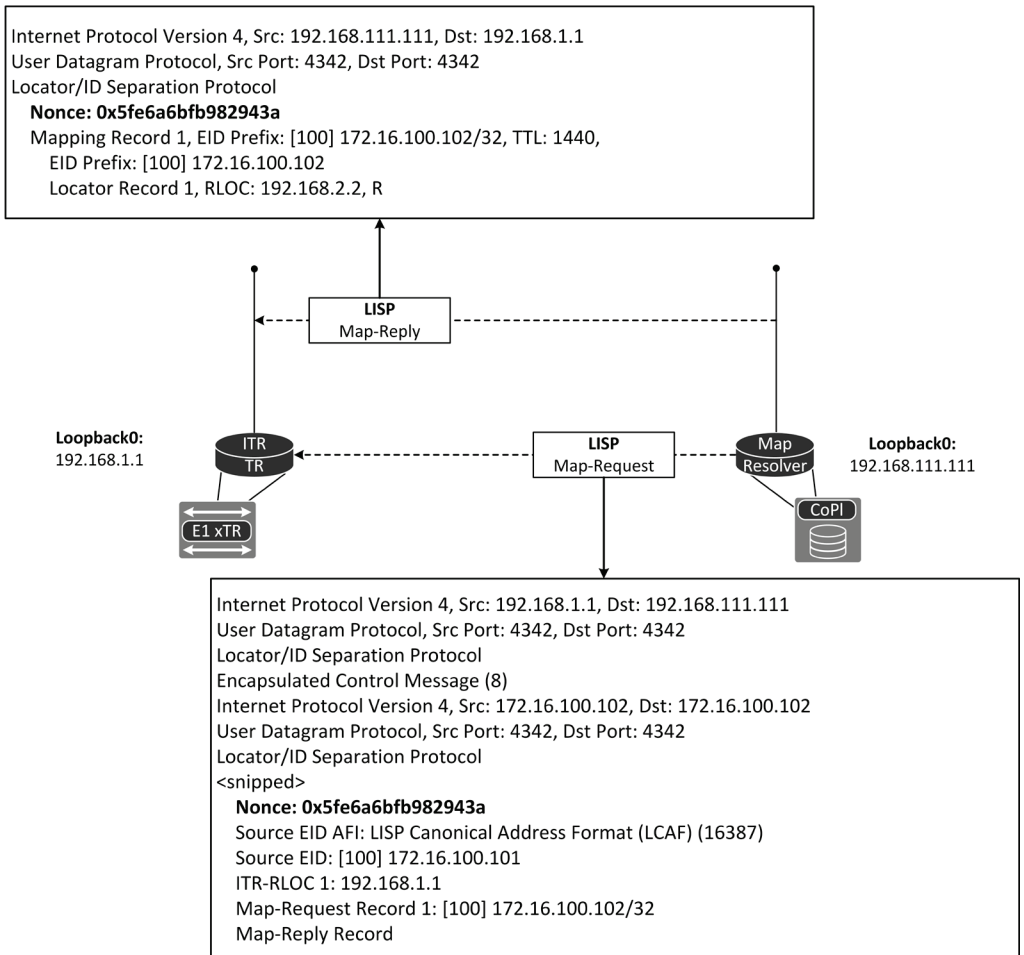


Figure 6-6: *Map-Reply Integrity By ITR.*

LISP-SEC Control-Plane Security

LISP enhanced Control-Plane security model LISP-SEC (LISP-SEC) strengthens the integrity checking and adds an authorization verification to the Map-Request and Map-Reply messages exchanged between the Ingress Tunnel Router and the Map-Resolver. This section discusses the operation of LISP-SEC without going into a detailed explanation of the hash algorithm or key generation processes.

ITR Map-Request Process

Both ITR and Map-Resolver know the Pre-Shared Key “nwktimes”. When LISP-SEC is enabled on ITR, it uses this key as a seed for generating the Pseudo Random Keys (PRK) which are used as an ITR One-Time-Key (ITR-OTK) for Map-Reply message authorization and integrity checking. To get the job done, the ITR uses the HMAC Key Derivation Function (HKDF). HKDF uses the *extract-then-expand model*, where it first extracts a fixed-sized Pseudo Random Key (PRK) from Input Key Material (IKM) that is PSK, and then expand that PRK to many additional PRKs. Figure 6-7 illustrates this process

- Phase 1.** The E1xTR starts the Map-Request process. The first phase is an extract-phase where the ITR uses the Pre-Shared Key (grey key) as an Input Key Material to generate a fixed-sized Pseudo-Random Key (white key) using HKDF.
- Phase 2.** The E1xTR then expands the Pseudo-Random Key into multiple keys by running the HKDF algorithm. ITR selects one of them to be used as a One-Time Key (ITR-OTK) for only this Map-Request message. It caches the ITR-OTK along with the nonce value generated for this particular Map-Request message.
- Phase 3.** E1xTR encrypts the ITR-OTK by using the Pre-Shared Key and HMAC. The resulting encrypted ITR-OTK is added into the Extended Authentication Data header in LISP Encapsulated Control Message (ECM). The S-bit in LISP ECM indicates the presence of authentication data wrapped in the ECM. The authentication data header is illustrated in figure 6-8.

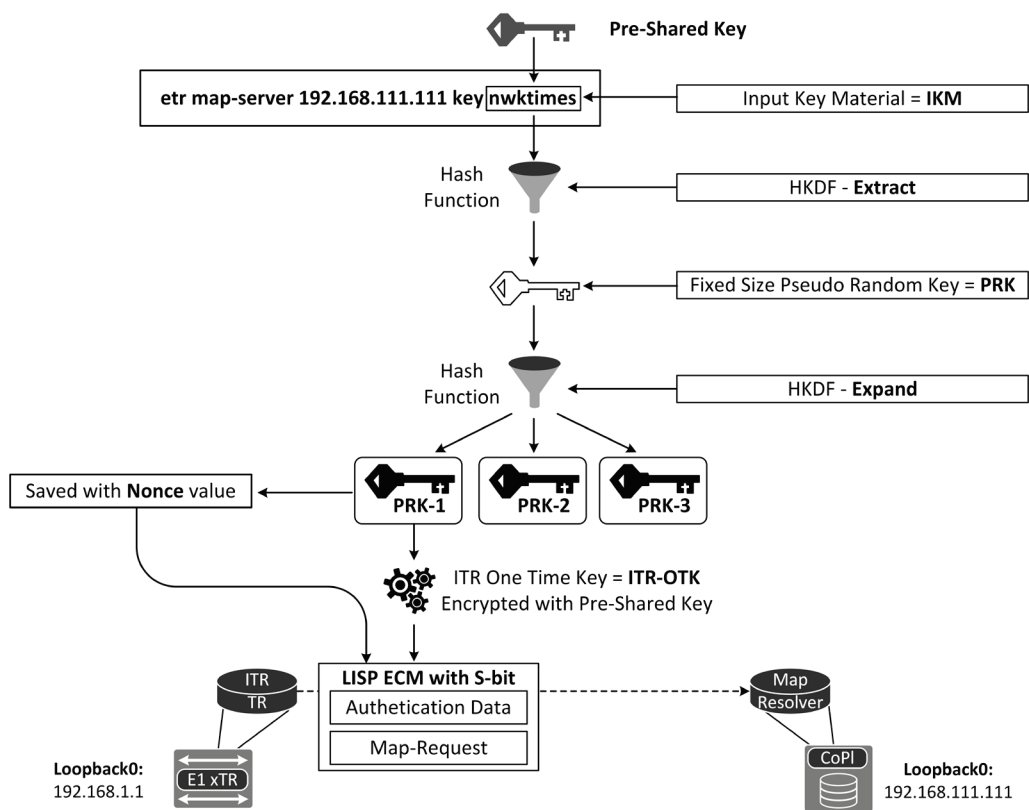


Figure 6-7: Map-Reply Integrity By ETR.

ECM Authentication Data header consists of two parts. The first part is One-Time Key Authentication Data and the second one is the Endpoint-Identifier Authentication Data.

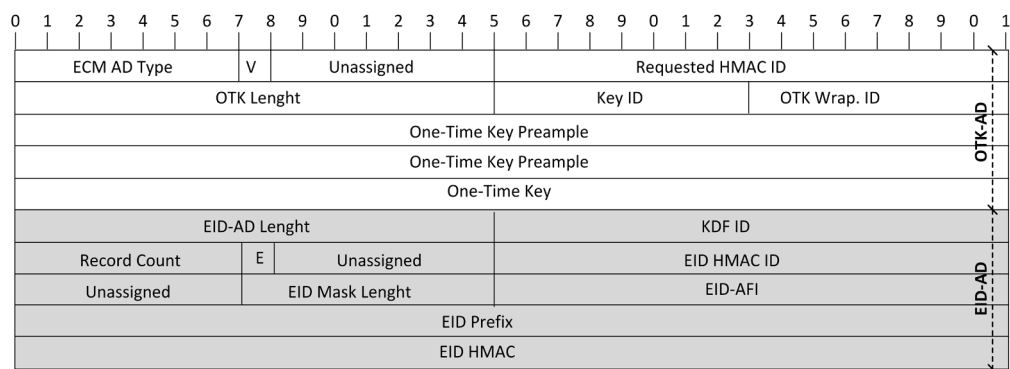


Figure 6-8: LISP-SEC Authentication Data header.

- ECM AD Type: Type 1 for the LISP-SEC Extended Control Message (LISP-SEC-ECM-EXT).
- V-bit: Key Version bit. This bit is toggled when sending ITR changes to the OTK Wrapping Key (see OTK Wrap. Id).
- Req. HMAC Id: This field describes the suggested HMAC algorithm to be used to protect the integrity of One-Time Key Authentication Data in Map-Reply message Map-Server/ETR. Supported HMAC IDs are NONE, AUTH-HMAC-SHA-1-96 and AUTH-HMAC-SHA-256-128. ITR can only suggest these HMAC IDs, not force to MS/ETR to use them.
- OTK Length: The Length of the OTK Authentication-Data portion from ECM AD Type to One-Time-Key.
- Key ID: In the case of multiple Pre-Shared keys, this field indicates which one is used for derived a per-message ITR-OTK.
- OTK Wrap. ID: This field describes the Key Derivation Function used for the generation of per-message OTK as well as the Encryption Algorithm to use for its protection. OTK Wrap IDs are [1] NULL-KEY-WRAP-128 and [2] AES-KEY-WRAP-128+HKDF-SHA256. HKDF-SHA256 derives the per-message OTK that is encrypted with the AES-KEY-WRAP-128 Algorithm. This field is used by replier when it decrypts the ITR-OTK.

OTK Preamble: If the OTK is not encrypted, this field is 0. When 128-bit OTK is used, all bits in this field are 0.

OTK Key: Per-message encrypted ITR-OTK.

EID Authentication-Data fields are included in ECM. However, some of these fields are used by ITR only to suggest to what HMAC the Map-Server/ETR may use in Map-Reply message. EID-AD fields are added into the Map-Reply message and explained in the next section.

Capture 6-1 illustrates the ECM packet sent by E1xTR. The S-bit indicates that the LISP-SEC extension header is wrapped into ECM. The HEX part includes both the Extension and the actual Map-Request messages.

```

Internet Protocol Version 4, Src: 192.168.1.1, Dst:
192.168.111.111
User Datagram Protocol, Src Port: 4342, Dst Port: 4342
Locator/ID Separation Protocol
  Type: Encapsulated Control Message (8)
    S bit (LISP-SEC capable): Set
    D bit (DDT-originated): Not set
    Reserved bits: 0x00000000
  Data (160 bytes)

0000  01 00 00 01 00 1c 00 01 00 00 00 00 00 00 00 00
0010  92 be b6 46 37 6b 78 54 9d 10 a6 71 6f 5d 9c da
0020  00 04 00 00 45 c0 00 7c 01 4e 00 00 ff 11 98 75
0030  ac 10 64 66 ac 10 64 66 10 f6 10 f6 00 68 c0 6f
0040  14 00 00 01 28 8d fa 75 b0 2b 7f 4e 40 03 00 00
0050  02 20 00 0a 00 00 00 64 00 01 ac 10 64 65 00 01
0060  c0 a8 01 01 00 20 40 03 00 00 02 20 00 0a 00 00
0070  00 64 00 01 ac 10 64 66 00 00 05 a0 01 20 10 00
0080  00 00 40 03 00 00 02 20 00 0a 00 00 00 64 00 01
0090  ac 10 64 65 01 01 01 01 00 05 00 01 c0 a8 01 01
    Data: 01000001001c00010000000000000000092beb646376b7854...
    [Length: 160]

```

Capture 6-1: ECM with LISP Extension Header sent by ITR.

Map-Server Map-Reply process

Figure 6-9 illustrates the Map-Server LISP-SEC process.

- Phase 1.** When Map-Resolver receives the Map-Request wrapped in ECM, it consults the Map-Server EID-to-RLOC Mapping database and checks the longest match EID-Prefix for requested EID. Note that both Map-Server and Map-Resolver are in the Control Plane Node and which is acting as a Proxy-ETR mode meaning that it is authorized to respond to Map-Request messages on behalf of an authoritative ETR (the xTR that has registered the EID-to-RLOC mapping information to Map-Server).
- Phase 2.** Map-Resolver decrypts the encrypted ITR-OTK by using the pre-shared key identified in the Key-Id field on received ECM. Map-Resolver then generates an MS-OTK using a Key Derivation Function. It attaches the KDF ID used for MS-OTK generation into LISP Map-Reply messages LISP-SEC extension headers.
- Phase 3.** MS uses MS-OTK key and HMAC for calculating the EID Message Authentication Code (HMAC) over Map-Reply Message. The resulting HMAC is then attached to the EID-AD HMAC field.
- Phase 4.** Map-Resolver also calculates the Packet Authentication Data (PKT-AD) over Map-Reply and LISP-SEC extension header and attached it to the PKT-HMAC field.

MS also copies the received Nonce value into Map-Reply message though the process is not described in the figure.

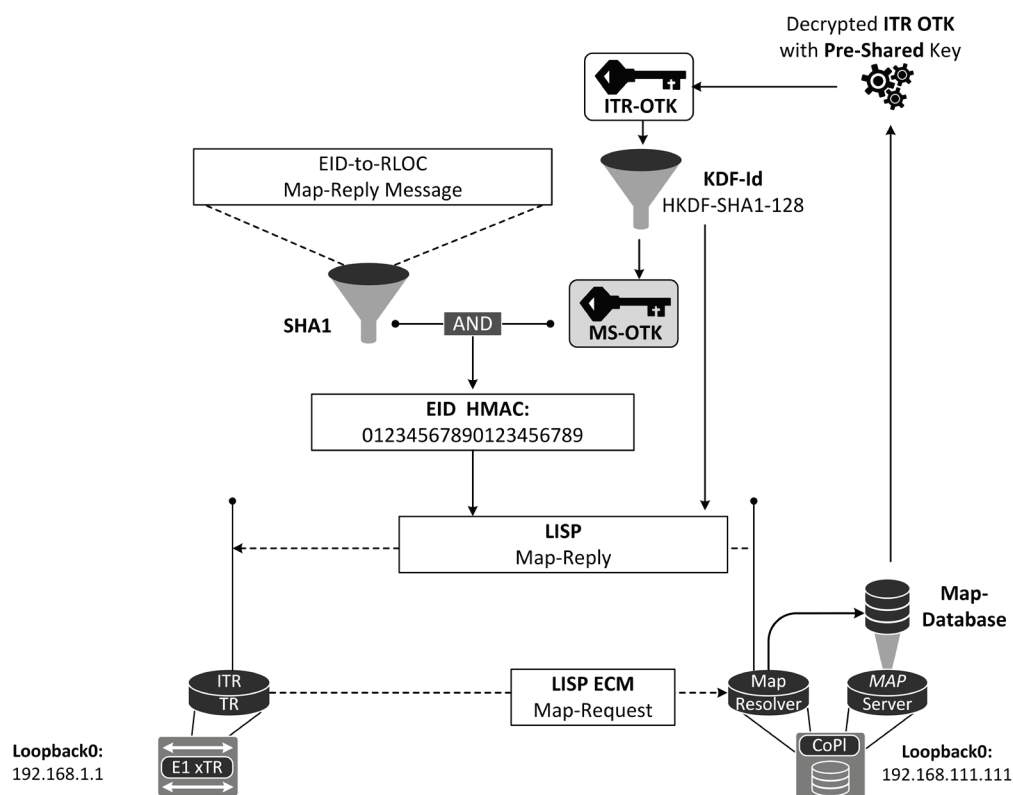


Figure 6-9: Map-Reply Integrity By ETR.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1		
MR AD Type								Unassigned																								EID-AD	
EID AD Lenght												KDF ID																					
Record Count								Unassigned								EID HMAC ID																	
Unassigned								EID Mask Lenght								EID-AFI																	
EID Prefix																																	
EID HMAC																																	
PKT AD Lenght																PKT HMAC ID																	PKT-AD
PKT HMAC																																	

Figure 6-10: LISP_SEC Extension header in Map-Reply message.

MR AD Type: Type 1 defines LISP-SEC Authentication Data.

EID-AD Length: EID Authentication Data Length.

- KDF-ID: Key Derivation Function Identifier used for generating the MS-OTK. HKDF-SHA1-128 (1).
- Record Count: The EID-Record included in Map-Reply message.
- EID-HMAC-ID: Used to protect the security of the EID-AD HMAC. Identifies the HMAC Algorithm. Supported HMAC IDs are NONE, AUTH-HMAC-SHA-1-96 and AUTH-HMAC-SHA-256-128.
- EID-Mask length: As its name describes, this field describes the mask length of EID.
- EID-AFI: EID-Prefix Address Family Identifier. The IPv4 identifier is 1.
- EID-Prefix: This is the longest match for the requested EID by ITR.
- EID HMAC: EID HMAC calculated over Map-Reply message using MS-OTK and Hash-based Message Authentication Code (HMAC).
- PKT-AD: Packet Authentication Data Length.
- PKT HMAC ID: HMAC Identifier. Supported HMAC IDs are NONE, AUTH-HMAC-SHA-1-96 and AUTH-HMAC-SHA-256-128.
- PKT HMAC: This field is calculated over the entire Map-Reply message including the LISP-SEC extension header. Note that for simplicity, Figure 6-9 does not illustrate the Packet Authentication Data process.

Capture 6-2 illustrates the Map-Reply with LISP-SEC extension header sent by Map-Responder.

Internet Protocol Version 4, Src: 192.168.111.111, Dst: 192.168.1.1
User Datagram Protocol, Src Port: 4342, Dst Port: 4342
Locator/ID Separation Protocol
Type: Map-Reply (2)
P bit (Probe): Not set
E bit (Echo-Nonce locator reachability algorithm enabled): Not set
S bit (LISP-SEC capable): Set

```

Reserved bits: 0x00000
Record Count: 1
Nonce: 0x67d771c3ddc97414
Mapping Record 1, EID Prefix: [100] 172.16.100.102/32, TTL: 1440,
Action: No-Action, Not Authoritative
    Record TTL: 1440
    Locator Count: 1
    EID Mask Length: 32
    Action: No-Action (0)
    Authoritative bit: Not set
    Reserved: 0x000
    Reserved: 0x0
    Mapping Version: 0
    EID Prefix AFI: LISP Canonical Address Format (LCAF) (16387)
    EID Prefix: [100] 172.16.100.102
        LCAF: Instance ID: 100, Address: 172.16.100.102
            LCAF Header: 00000220000a
            Instance ID: 100
            Address AFI: IPv4 (1)
            Address: 172.16.100.102
        Locator Record 1, RLOC: 192.168.2.2, Reachable,
Priority/Weight: 1/1, Multicast Priority/Weight: 1/1
    Priority: 1
    Weight: 1
    Multicast Priority: 1
    Multicast Weight: 1
    Flags: 0x0001
        Reserved: 0x0000
        Local: Not set
        Probe: Not set
        Reachable: Set
    AFI: IPv4 (1)
    Locator: 192.168.2.2
Data (48 bytes)
    Data: 010000000001c000000100000100200001ac10646610da2dd4...
    [Length: 48]

0000  50 00 00 02 00 00 50 00 00 01 00 00 08 00 45 c0
0010  00 80 01 22 00 00 fe 11 c8 c9 c0 a8 6f 6f c0 a8
0020  01 01 10 f6 10 f6 00 6c 37 3f 22 00 00 01 67 d7
0030  71 c3 dd c9 74 14 00 00 05 a0 01 20 00 00 00 00
0040  40 03 00 00 02 20 00 0a 00 00 00 64 00 01 ac 10
0050  64 66 01 01 01 01 00 01 00 01 c0 a8 02 02 01 00
0060  00 00 00 1c 00 00 01 00 00 01 00 20 00 01 ac 10
0070  64 66 10 da 2d d4 ba c3 b3 e8 ba fe cd 45 00 10
0080  00 01 19 e9 d0 ee 0b 2c 89 5d 47 ed 38 83

```

Capture 6-2: ECM with LISP Extension Header sent by ITR.

ITR Authentication Process

Figure 6-11 illustrates the ITR EID Authorization process.

Phase 1. When E1xTR receives the Map-Reply message, it calculates the MS-OTK by using the Key Derivation Function described in the LISP extension header of the Map-Reply message. Using MS-OTK and HMAC (HMAC Id also included in an extension header), it calculates the EID-AD. This locally calculated EID-AD is compared against received EID-AD, and if they are equal, the authentication succeeds.

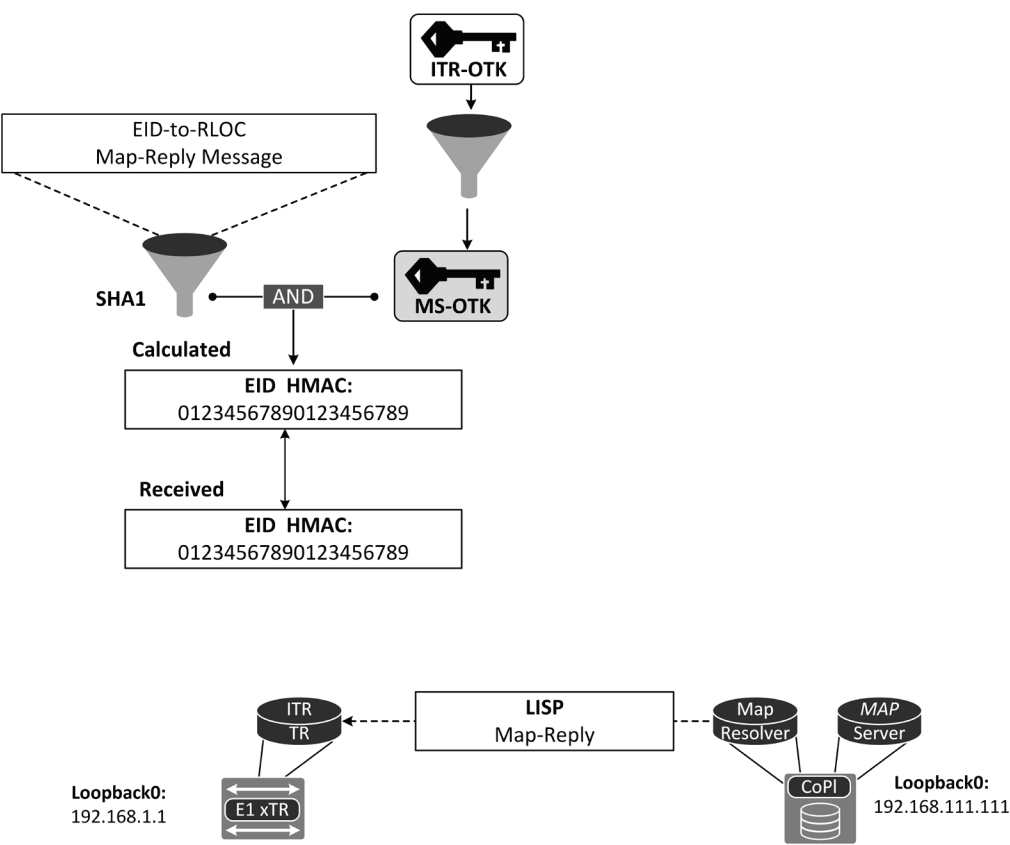


Figure 6-11: Map-Reply Integrity By ETR.

Configuration

The configuration is straightforward, enable LIPS-SEC under router lisp using command **security** *[strong]*. The **strong** option specifies that also Map-Reply messages are authenticated. In practice, this means that LISP-SEC has to be enabled on both ITR and MR.

References

- [RFC 2104] H. Krawczyk et al., HMAC: “Keyed-Hashing for Message Authentication”, RFC 2104, February 1997.
- [RFC 5869] H. Krawczyk and P. Eronen, “HMAC-based Extract-and-Expand Key Derivation Function (HKDF)”, RFC 5869, May 2010.
- [RFC 6830] D. Farinacci et al., “The Locator/ID Separation Protocol (LISP)”, RFC 6830, January 2013.
- [RFC 6833] V. Fuller and D. Farinacci, “Locator/ID Separation Protocol (LISP) Map-Server Interface”, RFC 6833, January 2013.
- [VXLAN-GPE] F. Maino et al., “Generic Protocol Extension for VXLAN”, draft-ietf-nvo3-vxlan-gpe-09, December 2019.
- [LISP Control Plane] D. Farinacci et al., “The Locator/ID Separation Protocol (LISP) Control Plane”, draft-ietf-lisp-rdc6833bis-25, June 16, 2019.
- [LISP-SEC] F. Maino et al., “LISP-Security (LISP-SEC)”, draft-ietf-lisp-sec-19, July 23, 2019.

Chapter 7: Campus Fabric External Connections

Introduction

This chapter explains the solution used in order to establish IP connectivity between Campus Fabric and non-LISP sites such as the Internet and shared services. To be able to forward traffic between Campus fabric and non-LISP site EID-prefixes used in Campus Fabric have to be reachable in the non-LISP site and the other way around. Figure 7-1 illustrates the simplified Control-Plane operation model where EID-prefixes are advertised from the Campus Fabric to the non-LISP site. Figure 7-2, in turn, illustrates how non-LISP site IP prefixes are advertised into Campus Fabric.

EID-Prefix advertisement from Campus Fabric to non-LISP Site

The following steps describe to Control Plane operation in order to advertise EID-Prefixes from Campus Fabric to non-LISP Sites.

- Phase 1.** Host-A joins the network and generates a GARP message.
- Phase 2.** E1xTR receives the GARP-message. It sends a Map-register message to the Map-Server (Control Plane Node).
- Phase 3.** Map-Server stores the mapping information into its EID-to-RLOC database.
- Phase 4.** To advertise routes from EID-to-RLOC to the routing process, the Control Plane Node installs the mapping information to its VRF specific Routing Information Base (RIB).
- Phase 5.** From the RIB, the information is taken to the BGP process.

- Phase 6.** Control Plane Node sends the EID reachability information within the BGP-Update message to its iBGP peer Border-PxTR. Border-PxTR has two roles, it is a Proxy-ITR for ingress traffic and a Proxy-ETR for egress traffic. The device where these two proxy functionalities are co-located is called Proxy-xTR or shortly PxTR in this book.
- Phase 7.** Border-PxTR receives the BGP-Update message. Border-PxTR imports the route from the Adj-RIB-in to the BGP Loc-RIB based on Route-Target 1:100 (used with VRF 100_White). The route is installed from the BGP Loc-RIB into the RIB of VRF 100_White.
- Phase 8.** Border PxTR forwards BGP-Update via Adj-RIB-Out to its eBGP peer Shared-Ro22. From the Border-PxTR perspective, this is a VRF 100_White specific eBGP session.
- Phase 9.** Shared-Ro22 receives the BGP-update message and installs the NLRI into BGP Loc-RIB and from there into the global RIB. Note that Shared-Ro22 is not VRF aware.

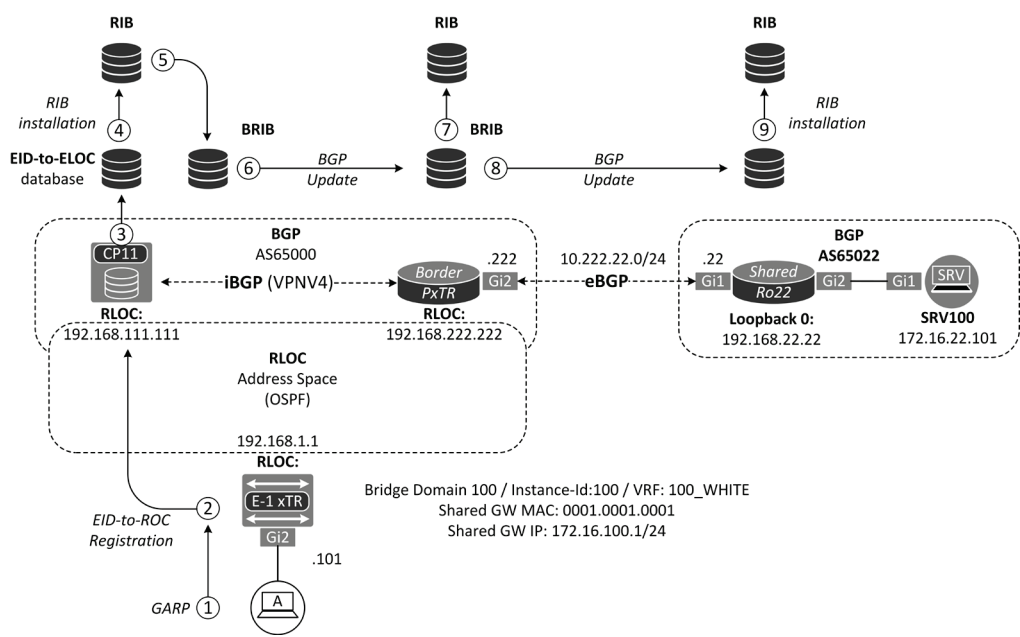


Figure 7-1: EID-Prefix Advertisement from Campus Fabric to non-LISP Site.

IP-Prefix advertisement from non-LISP Site to Campus Fabric.

The following steps describe to Control Plane operation in order to advertise non-LISP Site prefixes to Campus Fabric.

- Phase 1.** Shared-Ro22 has a connected network 172.16.22.0/24.
- Phase 2.** The network is taken into the BGP process and advertised to the eBGP peer Border-PxTR through the Adj-RIB-Out.
- Phase 3.** Border-PxTR receives the BGP-Update message over the VRF 100_White eBGP peering. It installs the route from Adj-RIB-In to Loc-RIB. During the import process, it adds the Route-Distinguisher 1:100 as a prefix for network 172.16.22.0/24. The route is then exported from the Loc-RIB into the RIB of VRF 100_White.
- Phase 4.** Border-PxTR imports the routing information into the LISP process.
- Phase 5.** Border-PxTR sends a LISP Map-Register message to the Map-Server. The network 172.16.22.0/24 is used as an EID and the RLOC IP address of Border-PxTR is used as Locator-Id. The Map-Server publishes the EID-to-RLOC mapping information in the mapping database.

At this stage, every node on both Campus Fabric and the non-LISP site has the necessary information to forward IP-Packets between Host-A in Campus Fabric and SRV100 in non-LISP Sites. E1xTR is not aware of the external network 172.16.22.0/24 at this phase because there has not been any data towards network 172.16.22.0/24. E1xTR.

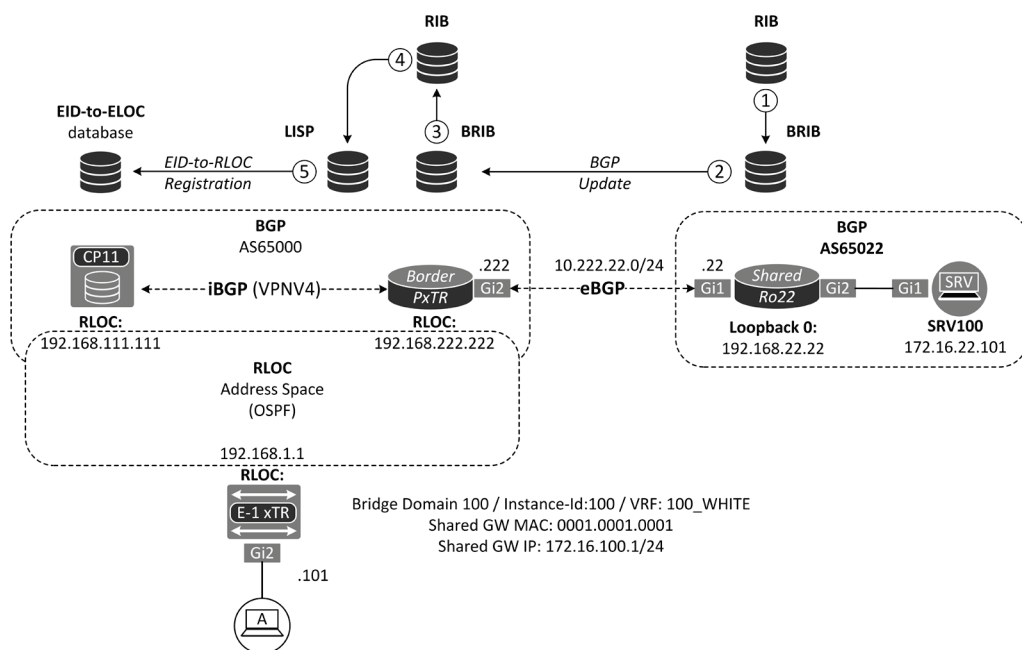


Figure 7-2: *EID-Prefix Advertisement from non-LISP Site to Campus Fabric.*

Data-Plane operation

The Data-operation on LISP based Campus Fabric requires Control-Plane operation (Map-Request) when data is sent to destination EID that has neither RIB nor LISP Mapping-Cache information. This means that the destination EID-to-RLOC mapping information is requested from the Map-Server by ITR and the received information in Map-reply is stored into Map-Cache. After the Map-Request/Map-Reply message exchange, the data flow can be forwarded towards the destination EID. Figure 7-3 explains the processes where host SRV100 in non-LISP site sends an ICMP-Request to Host-A in Campus Fabric.

ICMP-Request from SRV100 (non-LISP site) to Host-A (Campus Fabric).

- Phase 1.** Host SRV100 sends the first ICMP-Request message to Host-1.
- Phase 2.** Shared-Ro22 in the non-LISP site receives the ICMP-Request message. Shared-Ro22 makes the routing lookup and because it is a hit, it forwards the message to next-hop-router Border-PxTR in Campus Fabric.
- Phase 3.** Border-PxTR receives the ICMP-Request message. It first makes routing lookup from the RIB of VRF 100_White. The result is negative, so then it makes a LISP Mapping-Cache lookup, and the result is also negative. Note that the RIB lookup is excluded from the picture. Next, Border PxTR sends a Map-Request to Map-Resolver to find out the EID-to-RLOC mapping information.
- Phase 4.** Map-Resolver receives the EID-to-RLOC Map-request. It consults the EID-to-RLOC mapping-database published by Map-Server.
- Phase 5.** Map-Resolver responds to Border-PxTR by sending a Map-Reply message with the information of RLOC (E1xTR) that has published the EID of Host-A.
- Phase 6.** Border-PxTR receives the Map-Reply and stores the information into LISP Mapping-Cache.
- Phase 7.** Host SRV100 sends the second ICMP-Request to Host-1.
- Phase 8.** Shared-Ro22 in the non-LISP site received the ICMP-Request. It makes routing lookup and because it is a hit, it forwards the packet to next-hop-router Border-PxTR in Campus Fabric.
- Phase 9.** Border-PxTR receives the ICMP-Request message. It checks the LISP Mapping-Cache and due to Map-Reply received from Map-Resolver, it finds out the EID-to-RLOC information. It encapsulates the original ICMP-Request message with VXLAN header (MAC/IP/UDP/VXLAN) and forwards the packet towards the E1xTR. Core router forwards packet based on the outer IP-header (not shown in the figure).

Phase 10. E1xTR receives the VXLAN encapsulated ICMP-Request message. Based on the destination IP address in the outer IP header, it knows that it should process the message. It checks the VN-Id from the VXLAN header and notices that the packet belongs to Instance-ID 100. It checks the MAC address-table and forwards the packet to Host-A.

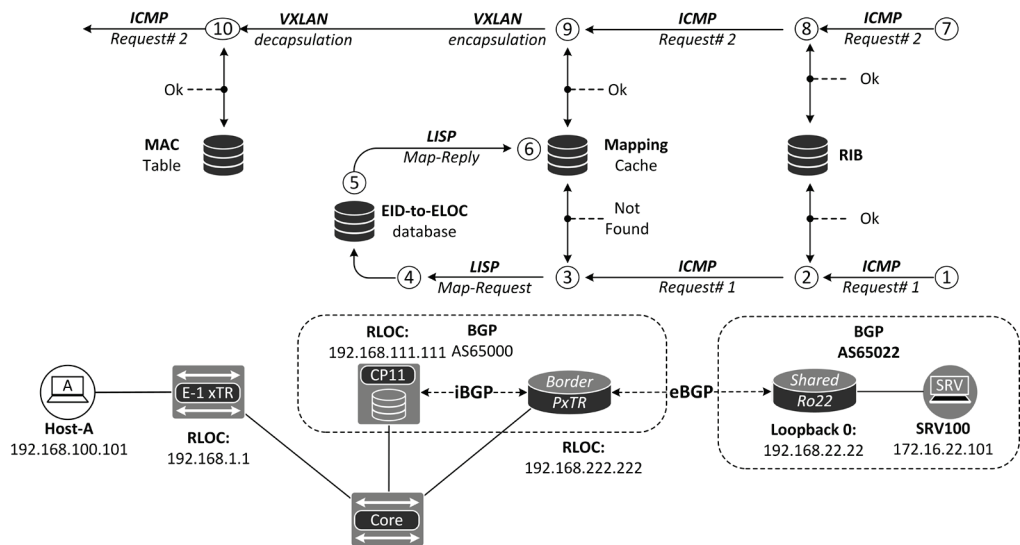


Figure 7-3: ICMP-Request from non-LISP site to Campus Fabric.

ICMP-Reply from Host-A (Campus Fabric) to SRV100 (non-LISP site).

The ICMP-Reply message forwarding follows the same processes as what was seen when SRV100 sends the first ICMP-Request message.

- Phase 1.** Host-A receives the ICMP-Query message. It replies with the ICMP-Reply message.
- Phase 2.** E1xTR receives the packet. It checks the RIB of the VRF 100_White and then the LISP Mapping-Cache. Because this is the first packet to SRV100, there is no EID-to-RLOC mapping entry yet. Next, Border ITR Edge-1 sends a Map-Request to Map-Resolver.
- Phase 3.** Map-Resolver receives the EID-to-RLOC Map-request. It consults the EID-to-RLOC mapping-database published by Map-Server.
- Phase 4.** Map-Server has EID-to-RLOC mapping in its database. Map-Responder replies to E1xTR by sending a Map-Reply message about network 172.16.22.0/24 with Locator set RLOC 192.168.222.222
- Phase 5.** E1xTR receives the Map-Reply message and installs it into Mapping-Cache.
- Phase 6.** Host-A sends the second ICMP-Reply. E1xTR forwards the ICMP-Reply message based on the Map-Cache entry. It encapsulates the ICMP-Reply message with the VXLAN header and forwards the packet to the Border-PxTR. Once again, the Core router makes its forwarding decision base on the outer IP header.
- Phase 7.** Border-PxTR receives the VXLAN encapsulated ICMP-Reply message. Based on the destination IP address in the outer header, it knows that it should process the packet. It checks the VN-Id from the VXLAN header and notices that the packet belongs to Instance-ID 100 that in turn is attached to VRF 100_White. It removes the VXLAN encapsulation and routes packet based on RIB of VRF 100_White to Shared-Ro22.
- Phase 8.** Shared-Ro22 receives the ICMP-Reply message it forwards the ICMP-Reply message to SRV100.

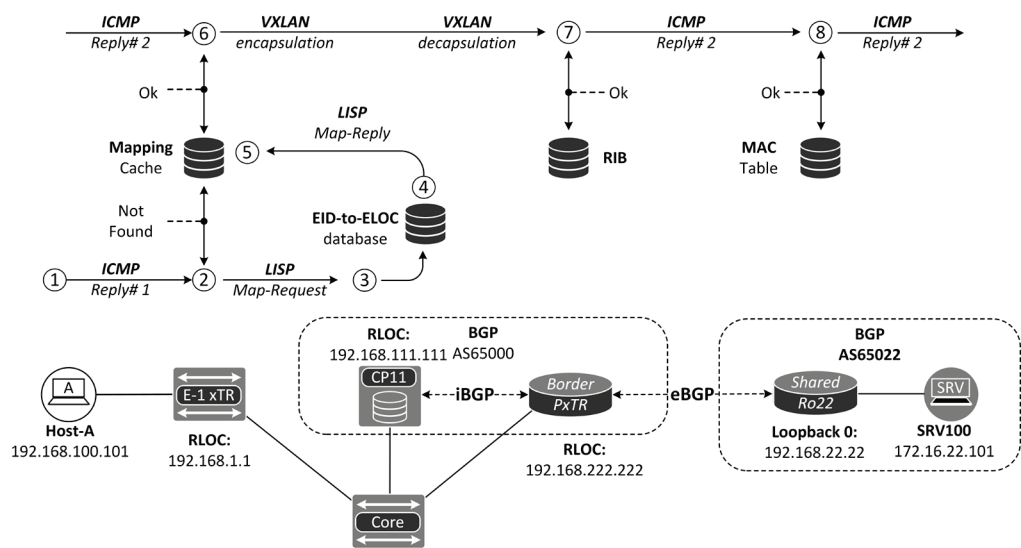


Figure 7-4: ICMP-Reply send by Host-A (Campus Fabric) to SRV100 (non-LISP site)

Traffic packet after Mapping-Cache Updates.

Figure 7-5 explains the data forwarding process after the LISP Mapping-Cache update process.

- Phase 1.** Host SRV100 starts pinging the Host-1. Shared-Ro22 in the non-LISP site receives the ICMP-Request message. It makes routing lookup and because it is a hit, it forwards the packet to next-hop-router Border-PxTR in Campus Fabric.
- Phase 2.** Border-PxTR receives the ICMP-Request message and forwards it based on the EID-to-RLOC (EID: 192.168.100.101-RLOC 192.168.1.1) mapping information. The message is wrapped inside the VXLAN header.

Phase 3. Edge-1 receives the packet. It verifies the VN-Id from the VXLAN header (VN-Id 100 = IID 100), and then removes the VXLAN encapsulation and forwards the message to Host-A.

Phases 4-6. The ICMP-Reply message is processed in the same way as the ICMP-Request message.

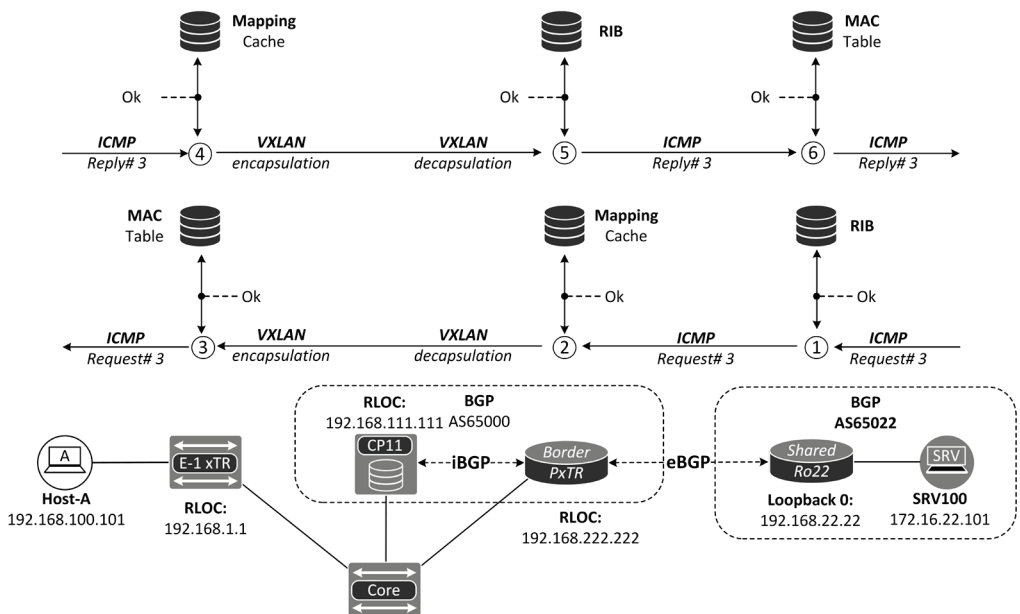


Figure 7-5: Packet flow after the LISP Map-Cache update process.

Configuration

This section introduces the configuration needed for Campus Fabric external connection. Complete device configuration can be found in Appendix B.

BGP Peering Configuration

Figure 7-6 illustrates the basic BGP setup. The iBGP Peering between Control Plane Node and Border-PxTR is afi VPNv4 MP-BGP. The eBGP peering between Border-PxTR and Shared-Ro22, from the Border-PxTR perspective, belongs to VRF 100_White. The eBGP peering from the Shared-Ro22 point of view is seen as the global BGP process and routes learned from Border-PxTR are installed into the global RIB. Another solution is to use VRFs also in Shared-Ro22. This solution though requires additional configuration if service/connections are required to be accessible from different Campus Fabric instances.

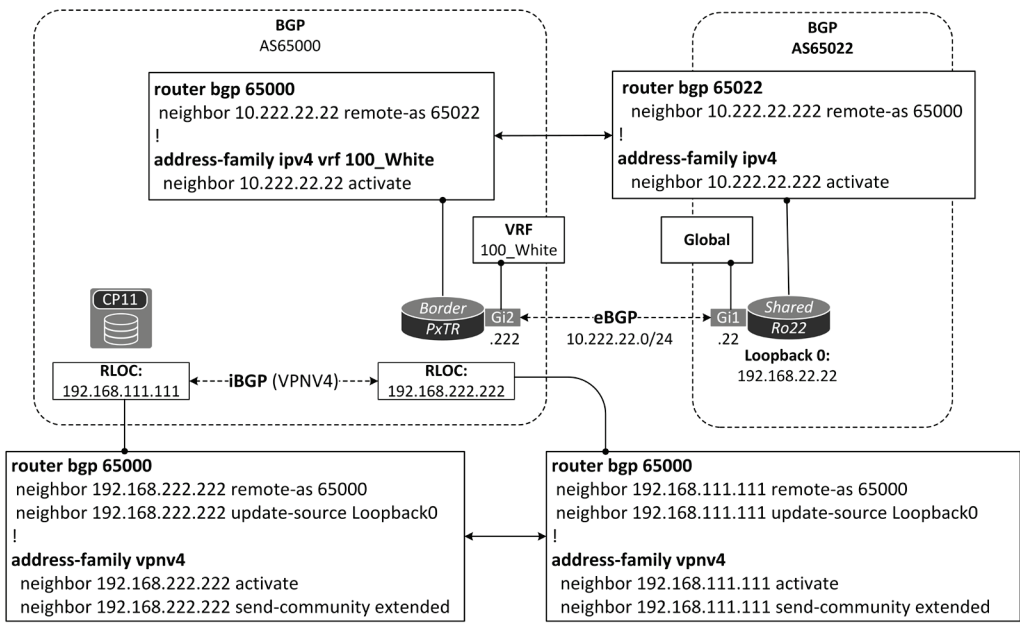


Figure 7-6: BGP peering Configuration.

Control Plane Node Configuration

Figure 7-7 illustrates the processes and configurations that are needed in order to Control Plane Node can send BGP-Update messages about its EID-to-RLOC Mapping Database to PxTR Border Node. These BGP-Update messages are required in order to Border-PxTR can send the Campus Fabric EID NLRI to non-LISP BGP speakers.

Phase 1. Create vrf 100_White in Control Plane Node.

Phase 2. Attach the VRF 100_White into IPv4 service under Instance-Id 100 with the command **eid-table vrf 100_White**. Export LISP routes from EID-to-RLOC Mapping-Database to RIB using the command **route-export site-registrations**. You can optionally set the AD of the exported route by using the command **distance site-registrations [AD-value]**.

Phase 3. Redistribute LISP (EID host routes) routes as an aggregate by using the command **aggregate-address 172.16.100.0 255.255.255.0 summary-only**, otherwise, host routes with mask /32 are advertised. The metric can be set by using the command **redistribute lisp metric [metric]**. Chapter 8 explains the LISP-to-RIB prefix filtering in order to avoid forwarding loops.

Network 172.16.22.0/24 used in this example belongs to the organization internal shared service domain which offer internal service such as DHCP. This means that the network has to be defined under the IPv4 service of site Network-Times as a part of Instance-Id 100 (VRF 100_White) by using the command **eid-record instance-id 100 172.16.22.0/24 accept-more-specifics**. This way the Map-Responder is able to reply to Map-Requests with Map-Reply messages where the authoritative PxTR is in Locator records RLOC. If networks in the shared segment are not defined under the site, Map-Server will respond with Negative Map-Reply which has no Locator Record. Edge switches will install information into Map-Cache but forwarding destination is set to Proxy-ETR defined in the LISP configuration of Edge switches. This is the case with the Internet-PxTR connection.

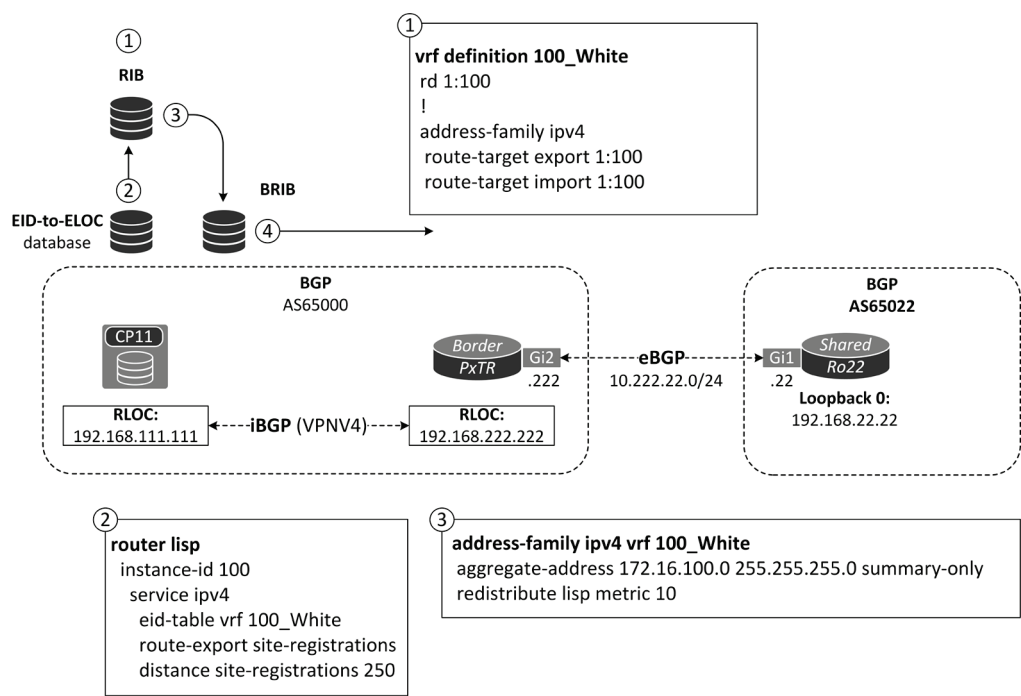


Figure 7-7: Exporting EID-Prefixes from LISP to BGP.

Border-PxTR Configuration (egress direction)

Figure 7-8 illustrates the process and configuration that are needed in order to Border-PxTR can send BGP-Update messages to Shared-Ro22.

- Phase 1.** Create vrf 100_White in Border Node.
- Phase 2.** Define BGP IPv4 afi for VRF 100_White in Border and configure eBGP peering with Shared-Ro22. Note that interface Gi2 belongs to VRF 100_White only in Border-PxTR.
- Phase 3.** Configure the Border-PxTR role to be both Proxy-ETR and Proxy-ITR.

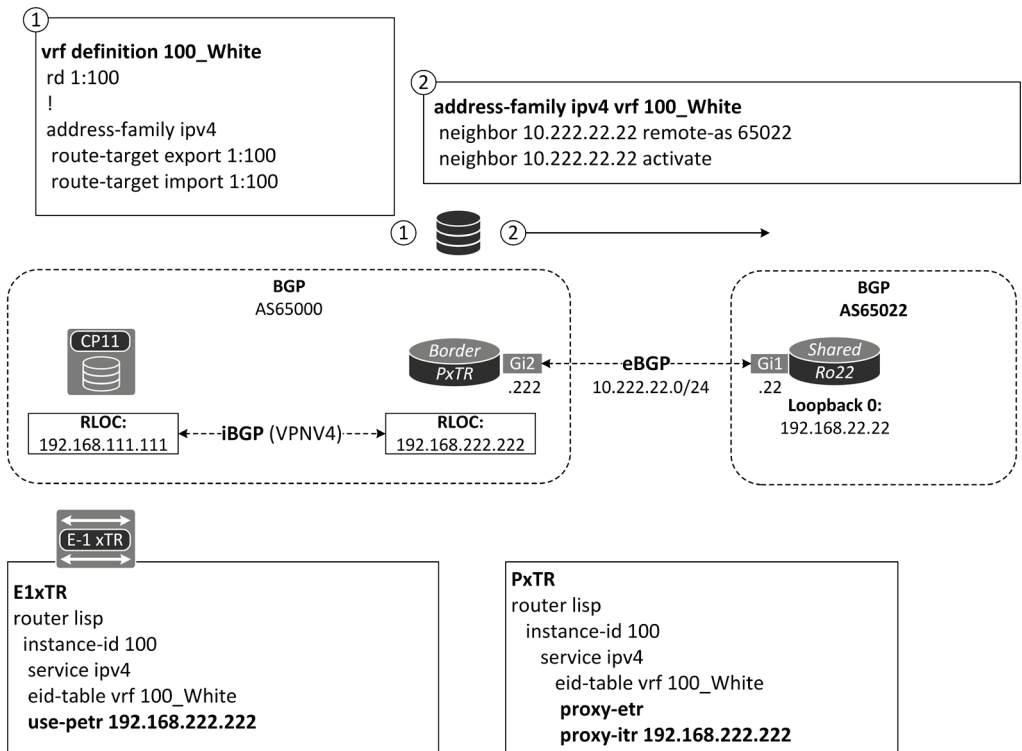


Figure 7-8: EID-Prefixes to a non-LISP BGP peer.

Border-PxTR Configuration (ingress direction)

Figure 7-9 illustrates the required configuration in the Border-PxTR in order to register the IP-Prefix learned from non-LISP Site to Map-Server.

Phase 1. Attach the VRF 100_White into IPv4 service under Instance-Id 100 with the command **eid-table vrf 100_White**. Use the RLOC IP address defined in RLOC-SET1 (Loopback 0) using the command **route-import database bgp 65000 locator-set RLOC-SET1**. AD can be optionally set by using the command **distance site-registration [AD-value]**.

Phase 2. In order to trigger the Map-Register process when traffic comes from non-LISP site use command **map-cache site-registration**.

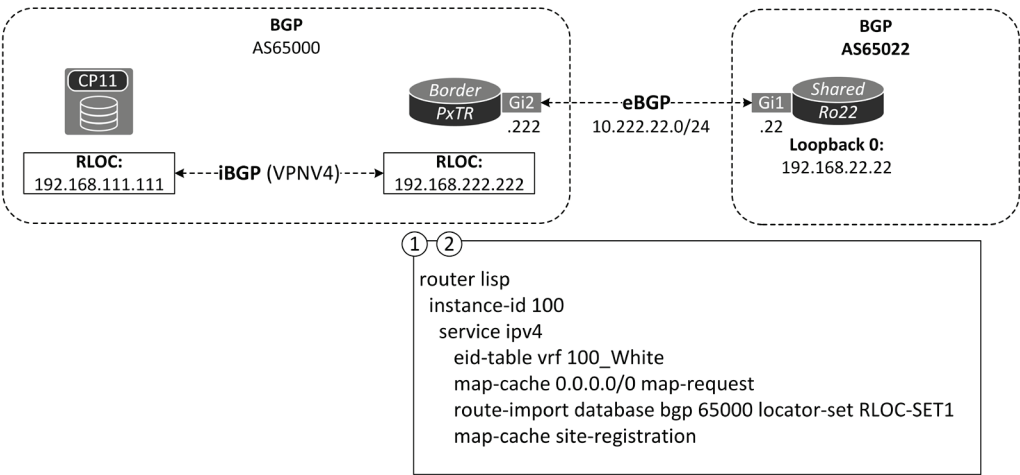


Figure 7-9: IP-Prefixes from non-LISP Site to Campus Fabric.

Control-Plane Operation: Campus Fabric to non-LISP site

This section illustrates the Control-Plane process in greater detail. In the first phase, neither Host-A nor SRV100 are connected to the network. The first section explains the operation from the Host-A EID-prefix processing perspective.

- Phase 1.** Host-A boots up and sends a GARP-message.
- Phase 2.** E1xTR receives the GARP-Message. The message is received on an interface that belongs to Layer 2 bridge-group that is attached to VRF 100_White that in turn is the VRF for Instance-Id 100. E1xTR sends a LISP Map-Register message to the Map-Server (Control Plane Node) where it describes the RLOC address used with the EID 192.168.100.101/32 belonging to the Instance 100.
- Phase 3.** Map-Server receives the Map-Register message. It stores the information into the EID-to-RLOC database under site “Network-Times” using Instance-Id 100. Also, due to command **route-export site-registrations**, it also exports the EID-to-RLOC mapping information into its VRF 100_White routing table.
- Phase 4.** The routing information is exported from the RIB to the BGP process and the EID NLRI is sent within BGP-Update message to Border-PxTR as an aggregate route.

LISP: Processing received Map-Register(3) message on GigabitEthernet1 from 192.168.1.1:4342 to 192.168.111.111:4342

LISP: Processing IP Map-Register mapping record for IID 100 172.16.100.101/32 LCAF 2, ttl 1440, action none, authoritative, 1 locator
192.168.1.1 pri/wei=1/1 LpR

LISP-0: MS Site EID IID 100 prefix 172.16.100.101/32 SVC_IP_IAF_IPv4 site Network-Times, Created dynamic site EID prefix entry.

LISP-0: MS registration IID 100 prefix 172.16.100.101/32 192.168.1.1 SVC_IP_IAF_IPv4 site Network-Times, Created new registration.

RT(100_White): updating lisp 172.16.100.101/32 (0x1) [local lbl/ctx:1048577/0x0] : via 0.0.0.0 Nu0

RT(100_White): network 172.16.0.0 is now variably masked

RT(100_White): add 172.16.100.101/32 via 0.0.0.0, lisp metric [250/1]

RT(100_White): updating bgp 172.16.100.0/24 (0x1) [local lbl/ctx:1048577/0x0] : via 0.0.0.0 Nu0 0 RT(100_White): add 172.16.100.0/24 via 0.0.0.0, bgp metric [200/0]

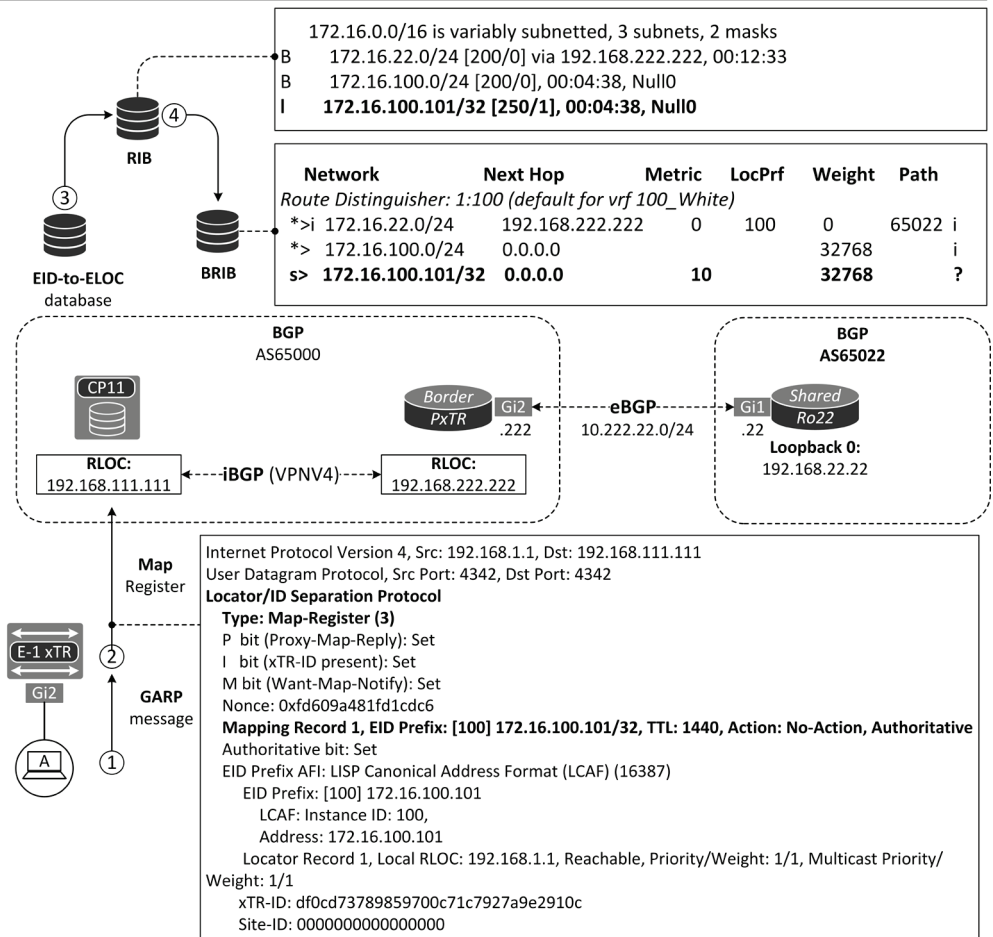


Figure 7-10: EID-Prefix Processing.

Phase 5. Control Plane node sends the BGP-Update message to its VPNv4 iBGP peer router Border-PxTR. The MP_REACH-NLRI Path-Attribute carries information about redistributed subnet 172.16.100.0/24. The Next-Hop Address is set to BGP Router-Id of advertising router with RD 0:0 (according to RFC 4364). The BGP-Update message also carries the Extended community value Route-Target 1:100.

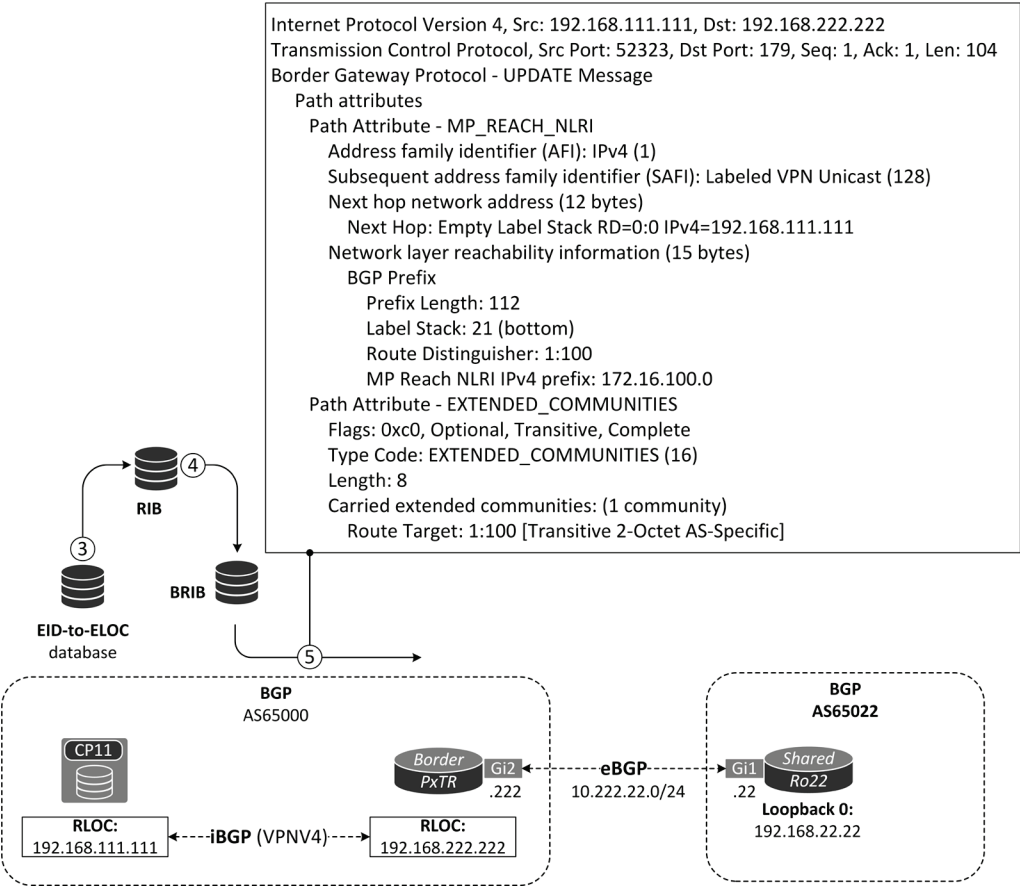


Figure 7-11: BGP Update from Control-Plane Node to Border-PxTR.

Phase 6. Border-PxTR receives the BGP-Update message from the Control Plane Node. The NLRI information is imported into the BGP Loc-RIB from the Adj-RIB-In based on the Route-Target 1:100 which is used in VRF 100_White. When importing the NLRI information into the BGP Loc-RIB table, the received RD 0:0 is changed to RD 1:100 which is used with VRF 100_White on Border-PxTR. The next-hop-router is Control Plane Node.

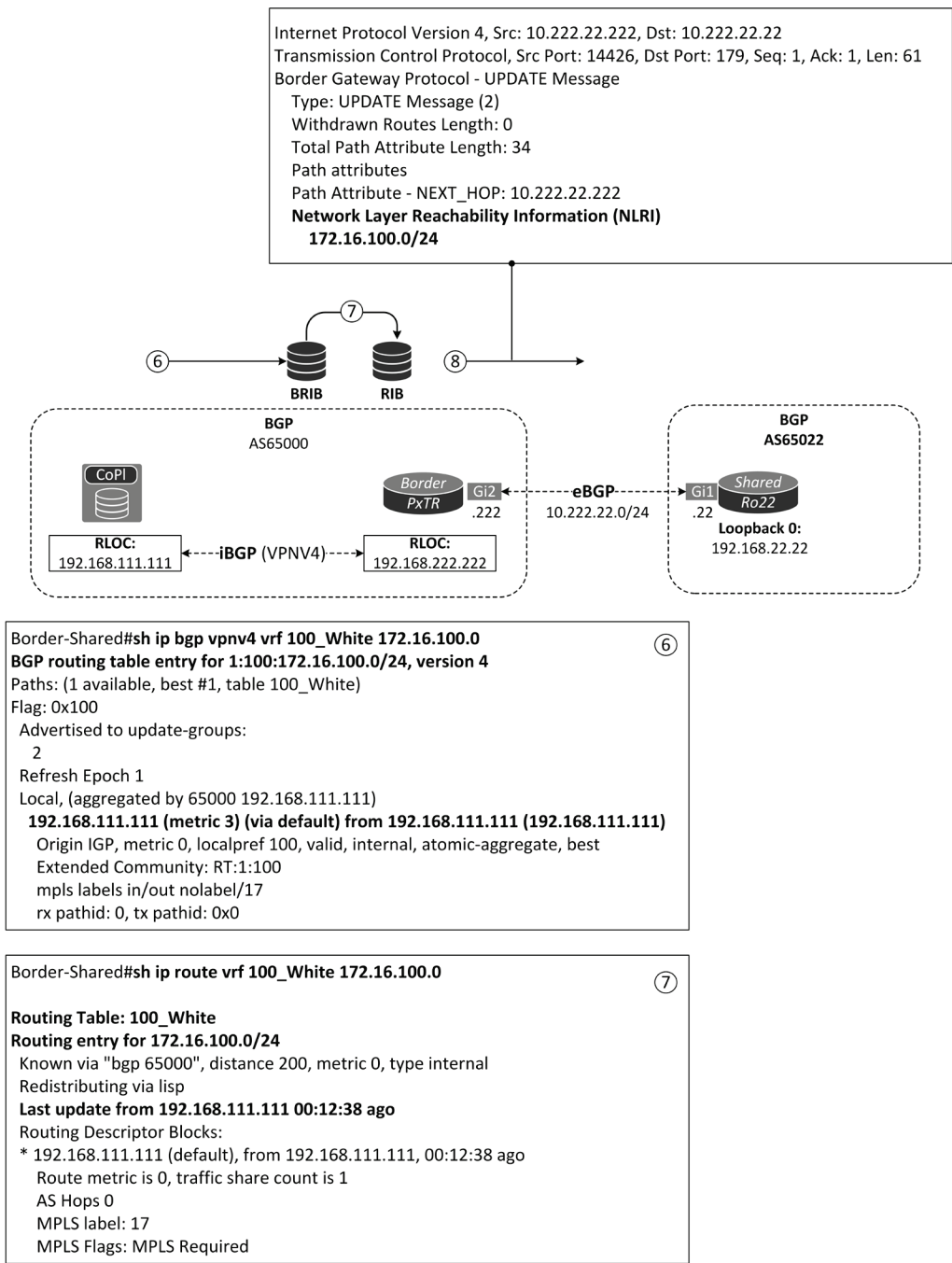
Phase7. Border-PxTR installs the route into the VRF 100_White RIB.

Phase 8. Next, Border-PxTR sends a BGP-Update message to its IPv4 eBGP peer Shared-Ro22. Because the BGP-Update message is sent to eBGP peer, Border-PxTR changes set its egress interface IP address as a next-hop for advertised NLRI.

When Shared-Ro22 receives the BGP-Update message, it installs the NLRI in its global BGP table. From there it imports the route into the global routing table.

If we summarize the process, the E1xTR receives the GARP message from Host-A. It generates the LISP Map-Register message and sends it to Map-Server. Map-Server installs the registration information in Mapping-Database and export information also into RIB where it is exported to the BGP process. Next, Map-Server sends a BGP-Update message to its VPNv4 iBGP peer about aggregate-address 172.16.100.0/24. Border-PxTR receives the BGP-Update message and imports the NLRI based on Route-Target 1:100. Next, it installs the route in its VRF 100_White specific RIB and advertises the information to its IPv4 eBGP peer Shared-Ro22 in the non-LISP domain.

At this phase, the subnet 172.16.100.100/24, where Host-A is attached to, is advertised from Campus Fabric to the non-LISP site. This means that Shared-Ro22 is able to forward traffic to 172.16.100.101.



Control-Plane Operation: non-LISP site to Campus Fabric

Figure 7-13 illustrates the process of route propagation from non-LISP sites to Campus Fabric.

- Phase 1.** Shared-Ro22 advertises its connected network 172.16.22.0/24 to its IPv4 eBGP peer router Border-PxTR. It uses its outgoing interface address as a next-hop-address.
- Phase 2.** Border-PxTR installs route from Adj-RIB-In into the BGP Loc-RIB. During importing route from Adj-RIB-In into Loc-RIB, the Route-Distinguisher is added into BGP table entry even though the original update did not carry RD for NLRI. The same process applies to Route-Target, it is added into the BGP table even though the original update did not carry Route-Target Extended-Community Path-Attribute.
- Phase 3.** Border-PxTR imports route from RIB Loc-RIB into the LISP process. Note that redistribution is always taken from the RIB, not from the routing protocol table such as BGP Loc-RIB or OSPF database.

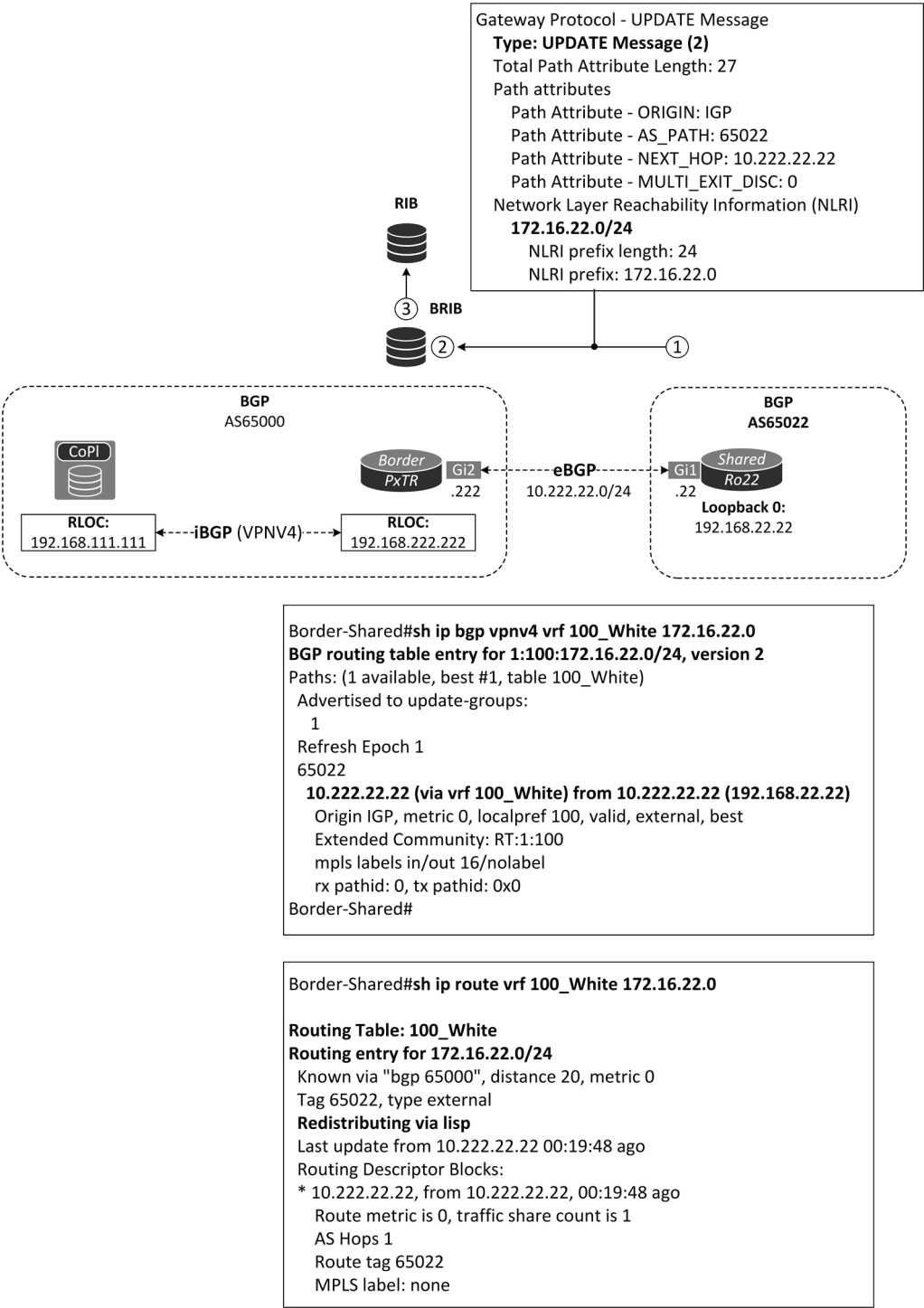


Figure 7-13: Non-LISP Site Subnet Processing.

Phase 4. Border-PxTR imports the BGP route from the RIB to the LISP process.

Phase 5. Border-PxTR sends the LISP Map-Register message to Map-Server.

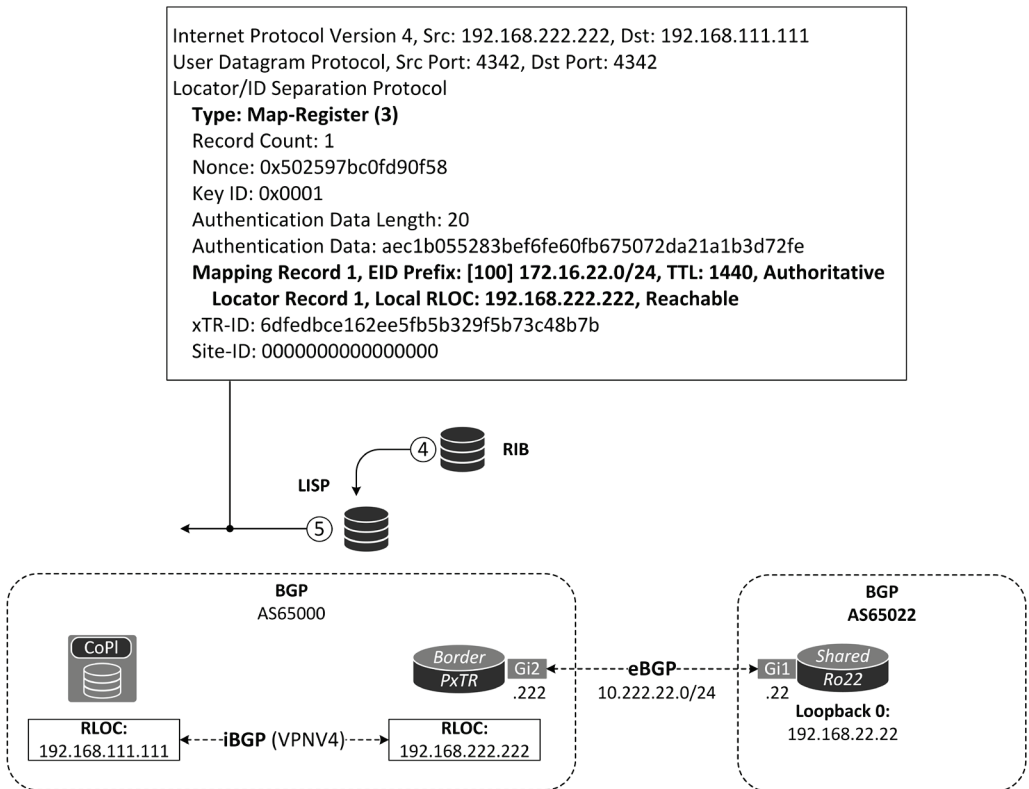


Figure 7-14: non-LISP Site Subnet Processing on Border-PxTR.

Phase 6. Map-Server receives the Map-Register message from Border-PxTR. Because the EID carried in the update is allowed in the site Network-Times Instance-Id 100, it is installed into the EID-to-RLOC mapping database. The information included in the database is just like any other host-specific EID information but with mask /24.

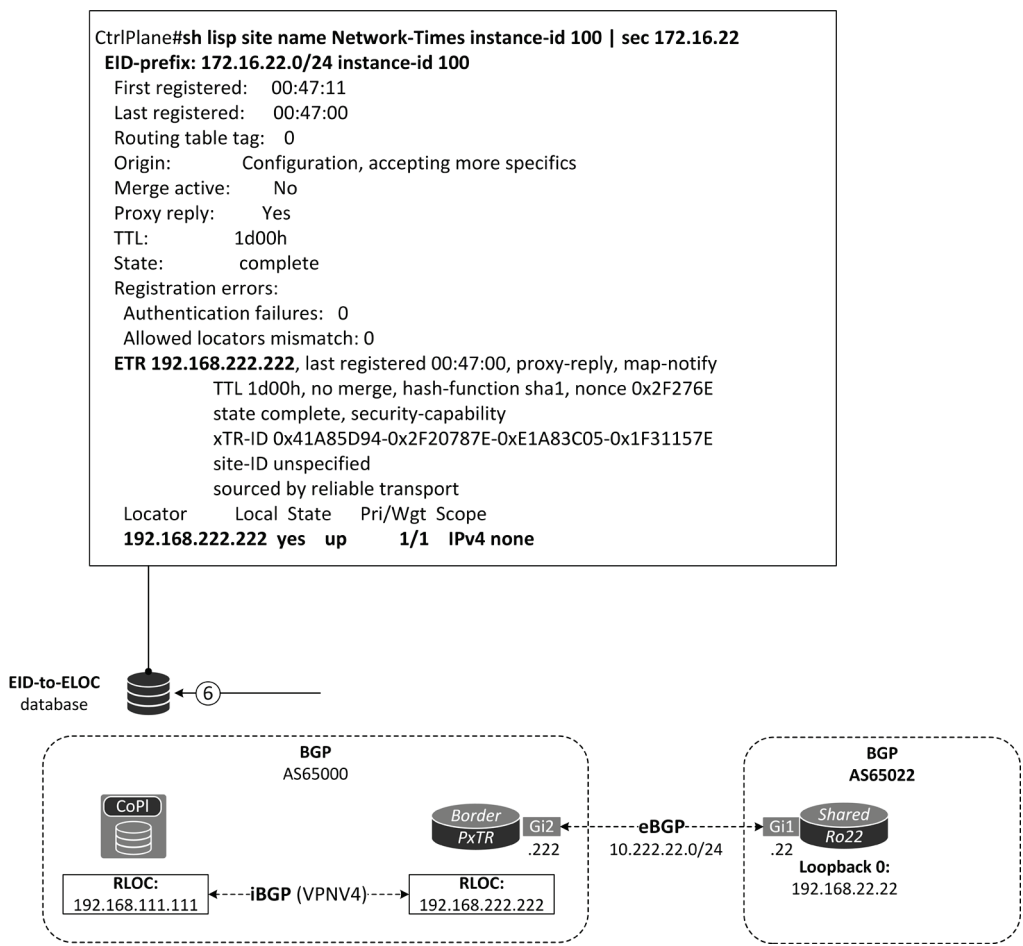


Figure 7-15: non-LISP Site Subnet Processing on Control Plane.

This section describes the EID registration process in detail. The next section discusses Data-Plane operation.

Phase 3. Map-Resolver receives the Encapsulated Control Plane message.

Phase 4. Map-Resolver consults the EID-to-RLOC Mapping database published by Map-Server. It finds the entry registered by Border-PxTR for 172.16.22.0/24 and sends a Map-Reply message back to E1xTR.

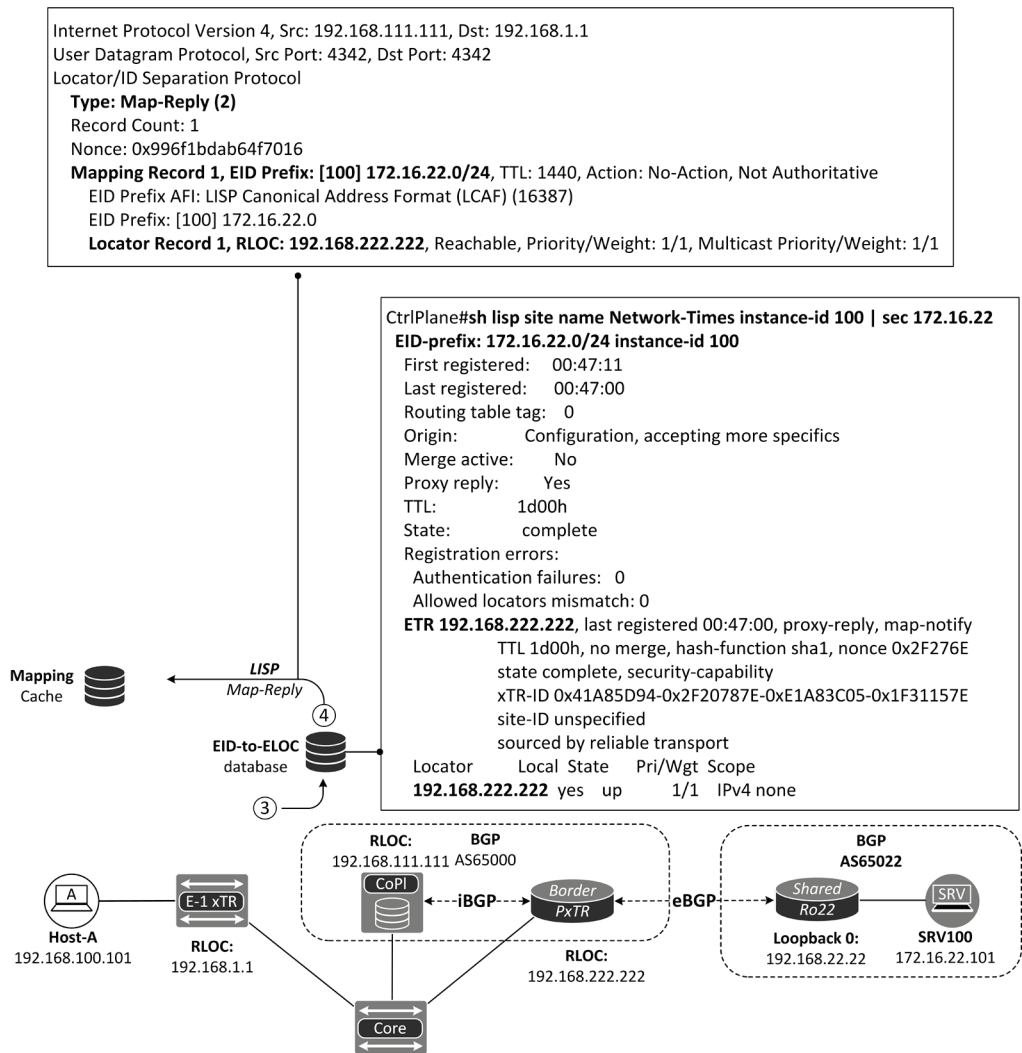


Figure 7-17: Data-Plane operation.

- Phase 5.** E1xTR receives the Map-Reply from Map-Resolver. It installs the information into Mapping-Cache.
- Phase 6.** When Host-A sends the next ICMP-Request, E1xTR first does routing lookup, and because there is no route to the destination, it checks the LISP Mapping-Cache, which now is a hit.
- Phase 7.** E1xTR encapsulates the original ICMP-Request inside VXLAN header using RLOC address found in Mapping-Cache as a destination IP address and using its RLOC address as a source IP address. Then it forwards the packet towards Border-PxTR.

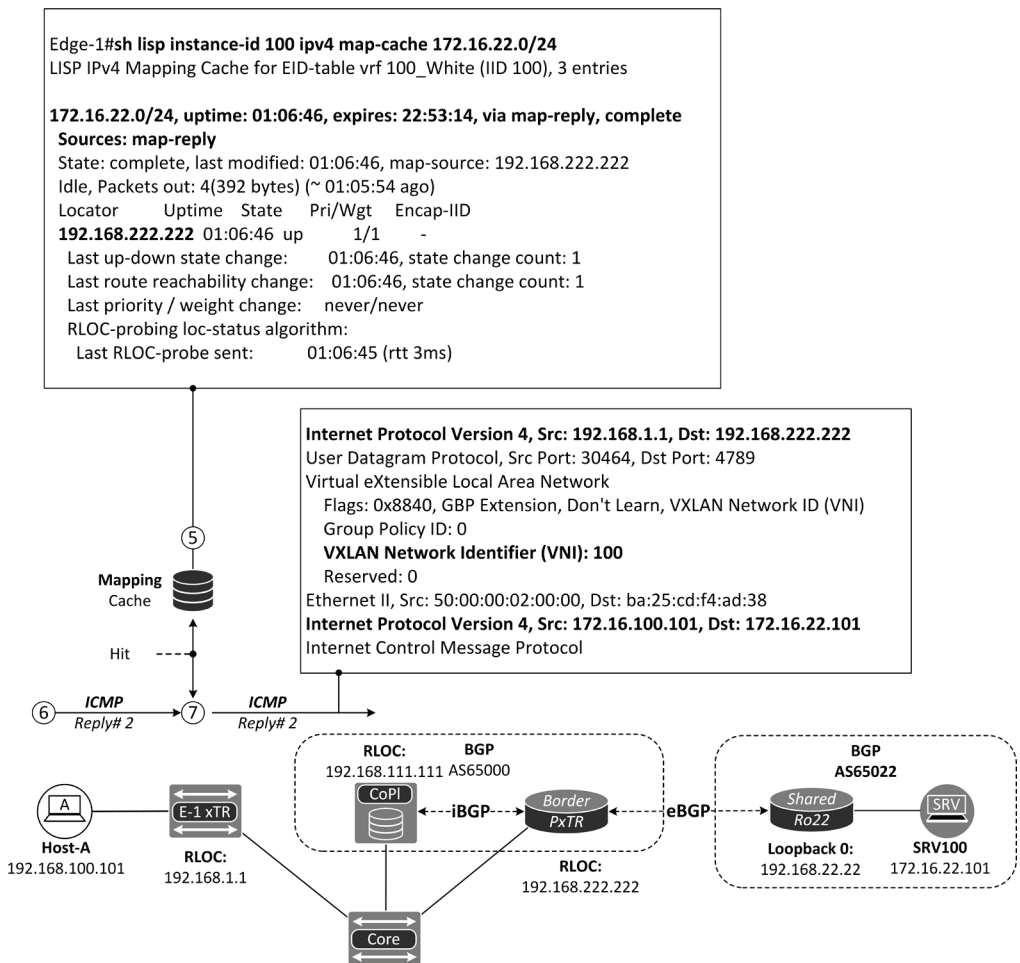


Figure 7-18: Data-Plane operation.

Eventually, the ICMP-Request is forwarded to SRV100, which sends an ICMP-reply. The process is the same in Campus Fabric's perspective, so there is no need to describe that process in detail.

Control and Data-Plane Operation: Internet

This section explains the process where the allowed EID-prefix list of site/Instance-Id in Mapping-Server does not have a statically configured entry for the external network. This is the case of Internet Border-PxTR. This process is demonstrated by removing the command `eid-record instance-id 100 172.16.22.0/24 accept-more-specifics` from the Map-Servers site Network-Times. We are going to use the same setup as in the previous example in a difference that now the Shared-Ro22 is Internet-Ro22 and the network 172.16.22.0/24 is some public www-site.

Figure 7-19 shortly describes the process. (1) Internet-Ro22 creates a BGP-Update message from its Loc-RIB and (2) send it to its eBGP peer Border-PxTR. (3) Border-PxTR receives the BGP update and installs it from Adj-RIB-In to VRF 100_White Loc-RIB where it is installed into VRF 100_White RIB. (4) The route is then exported to the LISP process and (5) advertised to Map-Server using the LISP Map-Register message. (6) Because this EID information is not permitted on Map-Server configuration, the Map-register message is rejected without installing it into the EID-to-RLOC Mapping database.

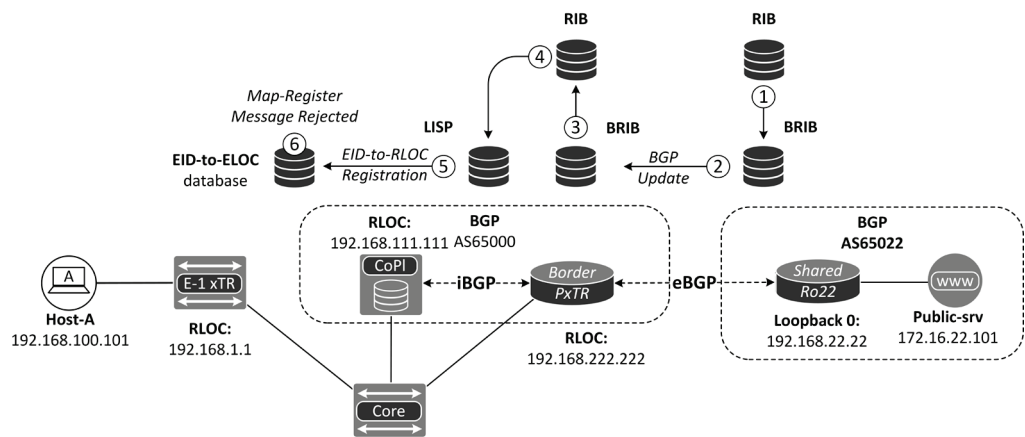


Figure 7-19: Data-Plane operation.

Figure 7-20 illustrates the process where Host-A starts pinging to Public-SRV 172.16.22.101. The ITR Edge-1 sends a LISP Map-Request message to Map-Resolver. This time Map-Server does not have EID-to-RLOC mapping information about requested EID, so Map-Resolver sends a *Negative Map-Reply* back to E1xTR. It sets the Action-bit to one (Natively-Forward). By doing this it instructs the requester to send packets to requested EID straight to configured Proxy-ETR. Note that there is no RLOC record specific Locator Record in negative Map-Reply message.

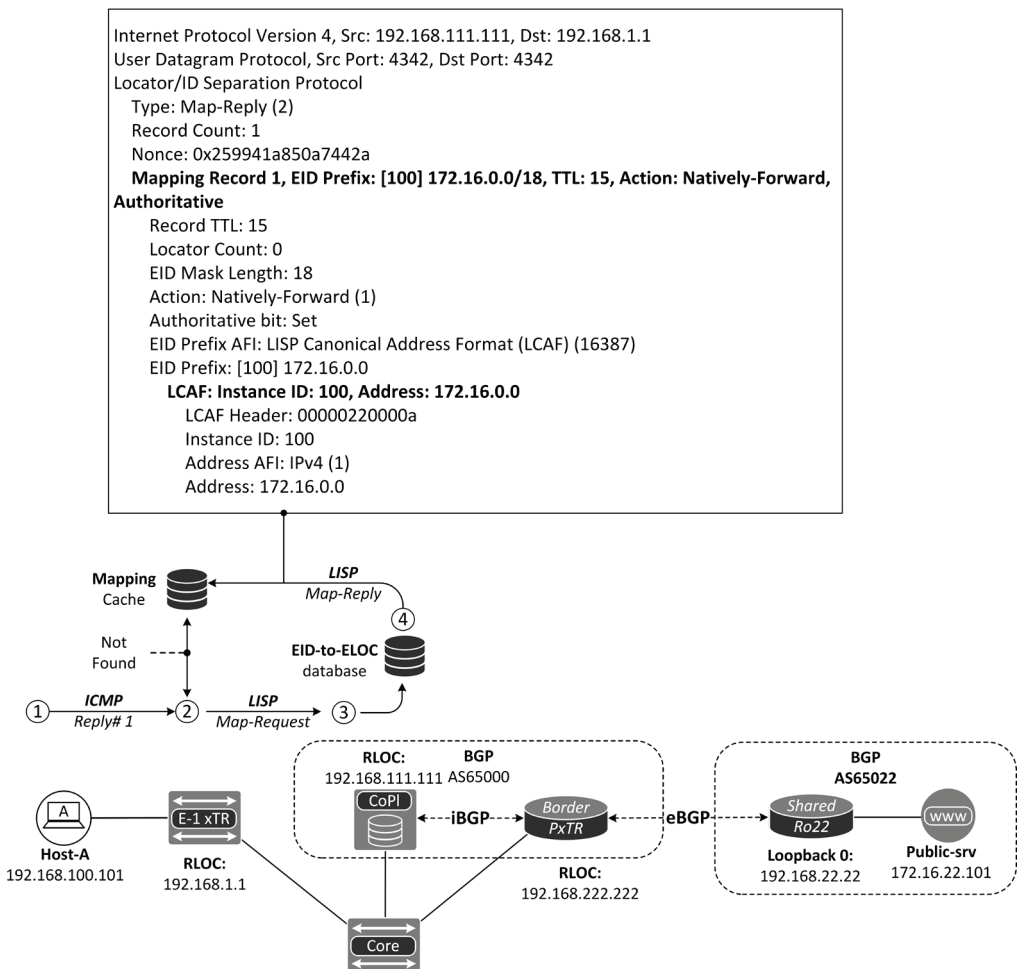


Figure 7-20: Data-Plane operation.

When E1xTR receives the next ICMP-request towards 172.16.22.101, it forwards it based on Proxy-ETR defined in its LISP configuration. This solution makes it possible to use a dedicated shared service Border router segregated from internet Border routers.

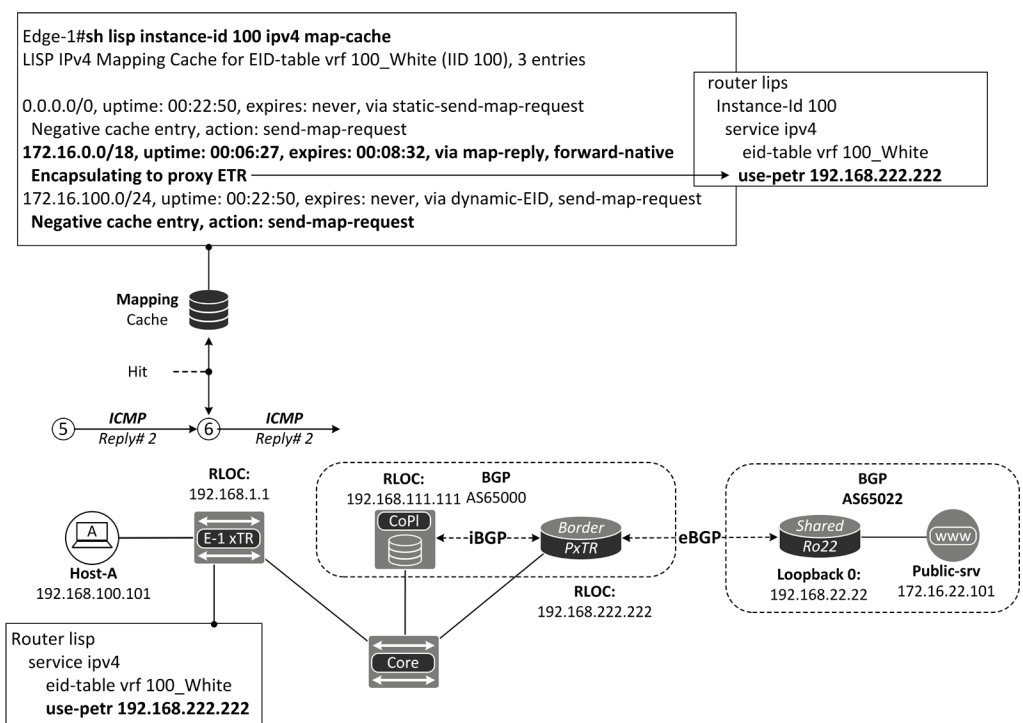


Figure 7-21: Data-Plane Operation.

Chapter 8: Campus Fabric Inter-VN Connections

Introduction

This chapter discusses the Inter-VN IP communication in Campus Fabric. First, it explains the basic operation model and then it explains the configuration needed for Inter-VN communication. The last part of this chapter explains the Control-Plane and Data-Plane operation in detail.

Inter-VN EID-Prefix advertisement model

Figure 8-1 shows the basic processes of how two endpoints belonging to different Layer 3 VRFs can establish IP connectivity between themselves in Campus Fabric.

- Phase 1.** Host-A joins the network and generates a GARP message. E1xTR receives the GARP message and generates the LISP Map-Register message where it describes the EID-to-RLOC mapping information. It sends this message to Map-Server.
- Phase 2.** Map-Server installs the received mapping information into the site Network-Times EID-to-RLOC mapping database with Instance-Id 100, from where it exports the information into VRF 100_White RIB using EID as a target IP address and RLOC as a next-hop-router.
- Phase 3.** The route is exported to BGP Loc-RIB from the VRF 100_White RIB using Route-Target 1:100 which is used in VRF 100_White.
- Phase 4.** Map-Server advertises the routing information as an aggregate-route through the BGP Adj-RIB-Out to Border-PxTR. Border-PxTR imports the NLRI carried in BGP-Update message based on the BGP Extended Community Path-Attribute Route-Target 1:100 from the Adj-RIB-In to BGP Loc-RIB.

- Phase 5.** Border-PxTR installs the route, after route validation, into the RIB of VRF 100_White.
- Phase 6.** Border-PxTR then sends a BGP-Update message to Shared-Ro22 located in the non-LISP site over VRF 100_White eBGP peering.
- Phase 7.** Shared-Ro22 receives the BGP-Update message from its eBGP peer 10.222.22.22 and installs the route from BGP Adj-RIB-In into BGP Loc-RIB and from there it is installed into the global RIB.
- Phase 8.** Shared-Ro22 sends a BGP-Update message to its eBGP peer Border-PxTR 10.222.33.222 through the Adj-RIB-Out.
- Phase 9.** Border-PxTR receives the BGP-Update from its eBGP peer Shared-Ro22 but via different BGP peering than what it is using for sending updates to Shared-Ro22 (VRF 100_White). Border-PxTR has to accept the BGP-Update message that carries its local AS number in the AS_Path Path Attribute list, otherwise, the BGP loop prevention policy discards the update. PxTR installs the route into VRF 200_Grey BGP Loc-RIB and from there into the same RIB of the same VRF.
- Phase 10.** Border-PxTR imports the routing information into the LISP process using the host address as an EID and its own RLOC address 192.168.222.222 as Locator Identifier.
- Phase 11.** Border-PxTR sends a LISP Map-Register message to Map-Server using Instance-Id 200 that is assigned to VRF 200_Grey. Map-Server receives the message and installs it into the Site Network-Times EID-to-RLOC database using the Instance-Id 200.

The same operation happens from the Host-D perspective in the opposite direction. When that is done, Map-Responder is able to reply to Map-Request messages concerning Host-A and Host-D sent by either E1xTR or E2xTR.

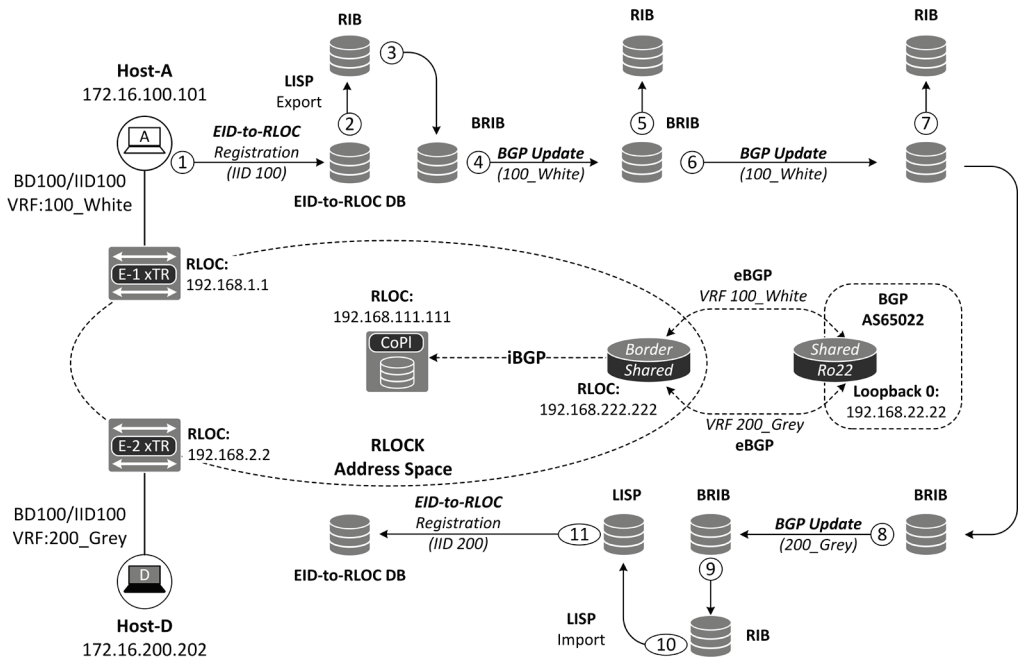


Figure 8-1: EID advertisement between VRFs.

Configuration

This section introduces the configuration required in order to establish an Inter-VN connection in Campus Fabric.

Control Plane Node configuration: VRF 100_White

The command **eid-record instance-id 100 172.16.200.0/24 accept-more-specifics** is needed in order to Map-Server can accept the received Map-Register message into site Network-Times EID-to-RLOC database. If this command is not used, the mapping messages concerning EID-Prefixes 172.16.100.xx/xx are rejected by Map-Server.

Routes are redistributed from LISP into the VRF 100_White BGP process through the BGP policy. We are using route-maps in this example but it could be any other suitable BGP filtering method. The redistribution policy has to be configured on the Control Plane Node, otherwise, there will be a routing loop. The loop will be formed without filter because all LISP routes are exported to BGP by default. On the lower part in figure 8-1, we can see how Control Plane Node receives the LISP Map-Register message carried information about Host-A IP and Location. It is installed into the EID-to-RLOC database using IID 200. Because Control Plane Node, without redistribution policy, export all IID 200 information form mapping EID-to-RLOC mapping database to BGP, also aggregated route 172.16.100.0/24 is redistributed. When this happens, the Border-PxTR Border receives BGP-Update messages about network 172.16.100.0/24 from both Shared router via VRF 100_White eBGP peering and via VPNv4 eBGP peering from Control Plane Node. This is why redistribution policy needs to be in place.

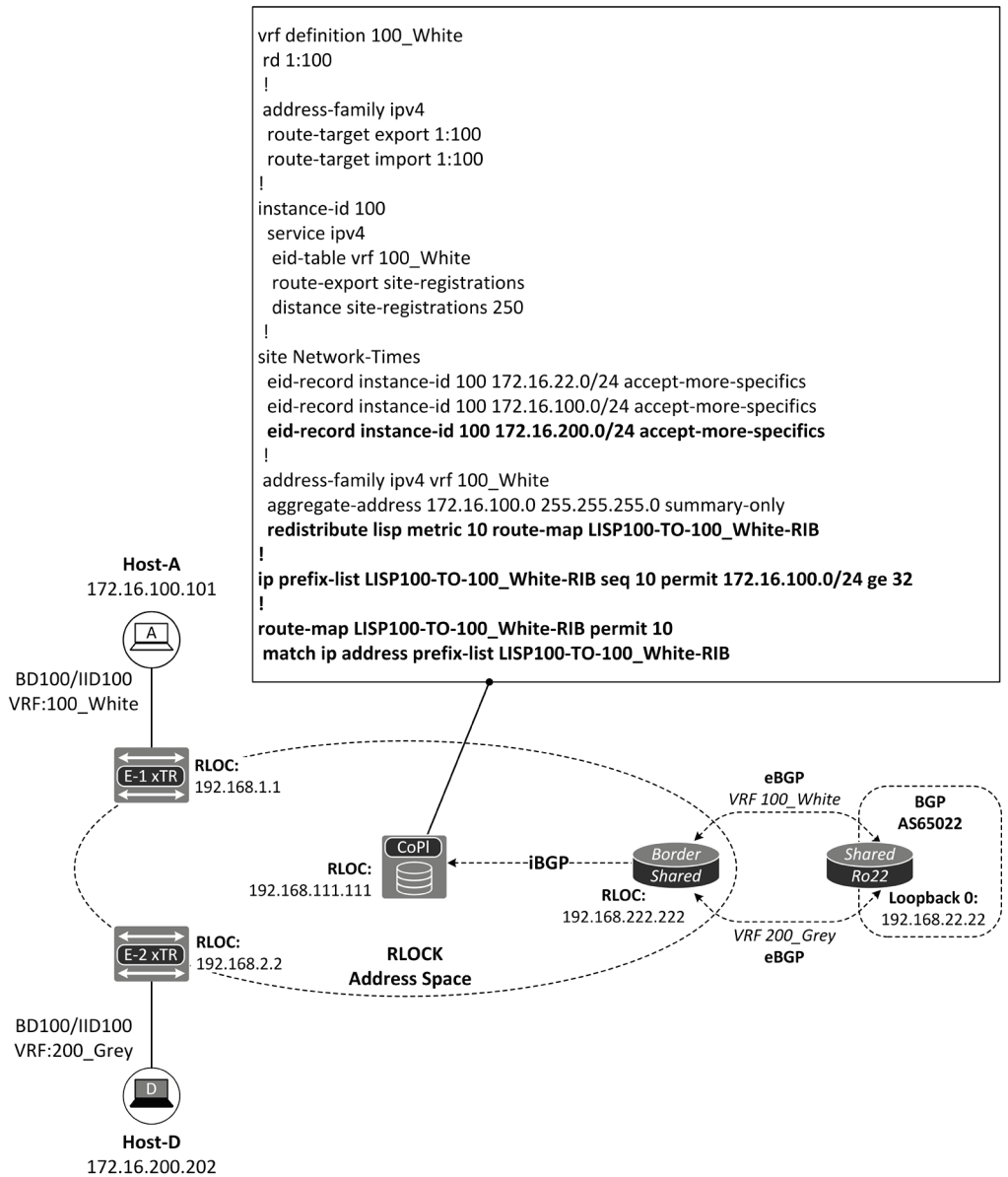


Figure 8-2: Control Plane Node Configuration, VRF 100_White.

Control Plane Node configuration: VRF 200_Grey

The same kind of configuration is also added to VRF 200_Grey. First, we allow the Inter-VN EID 172.16.100.0/24 under the site Network-Times IPv4 service to be installed using IID 200. Next, the LISP to BGP redistribution is done using a redistribution filter that permits only the original information.

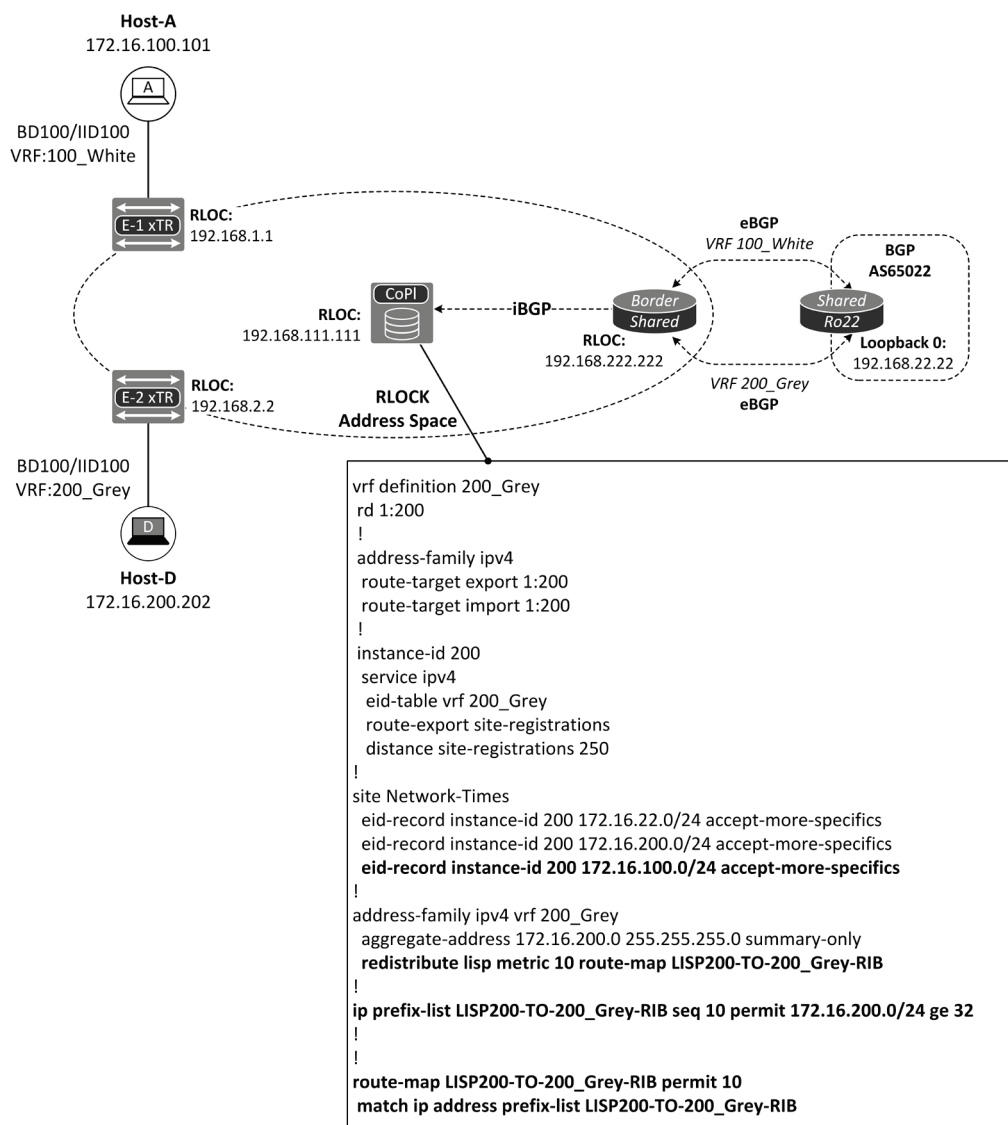


Figure 8-3: Control Plane Node Configuration, VRF 200_Grey.

PxTR configuration: VRF 100_Grey and VRF 200_Grey

Figure 8-4 shows the configuration needed in Border-PxTR. There are no logical differences between VRF 100_White and VRF 200_Grey. This section focuses on VRF 200_Grey.

Under the VRF 200_Grey configuration, there is a Route-Distinguisher 1:200 that is used as a VPN identifier in the BGP table. Route-Target 1:200, in turn, is used for BGP import/export policy. As an example, when Border-PxTR receives VPNv4 BGP-Update from Control Plane Node, the NLRI is first installed into the Adj-RIB-In table as is, without any modification. Then, if the BGP-Update message includes the BGP Extended Community Path-Attribute Route-Target with value 1:200, it is imported into the BGP Loc_RIB. During the import process, Border-PxTR adds Route-Distinguisher used in this VRF in front of the advertised IP-Prefix. This way it can differentiate BGP entries in its BGP Loc-RIB and it is able to install routes from Loc-RIB into correct VRF RIB.

BGP peering with Shared-Ro22 is done over the interface Gi3 that is attached to VRF 200_Grey. This way Border-PxTR knows which updates received from Shared-Ro22 belong to VRF 200_Grey.

External BGP peering with Shared-Ro22 is done in the same way that was introduced in the previous chapter. However, there is one additional command **neighbor 10.222.22.22 allowas-in**. This command bypasses the AS_Path Path-Attribute verification by allowing also BGP-Updates that includes the receiver BGP AS-Path. The other way to do this is to configure Shared-Ro22 to modify the AS-Path information when sending BGP-Updates.

Instance-specific configuration under the IPv4 service is the same as what was shown in the previous chapter. The command **eid-table 200_Grey** attaches the instance into VRF 100_Grey. Commands **map-cache 0.0.0.0/0 map-request** and **map-cache site-registration** generate the Map-Request for the destination EIDs for traffic coming from the non-LISP site. EID Received as BGP NLRI is imported into the LISP process and advertised to Map-Server using RLOC of Border-PxTR defined in the RLOC-SET1. The command **distance site-registration 250** is optional and it sets the AD for imported routes to 250.

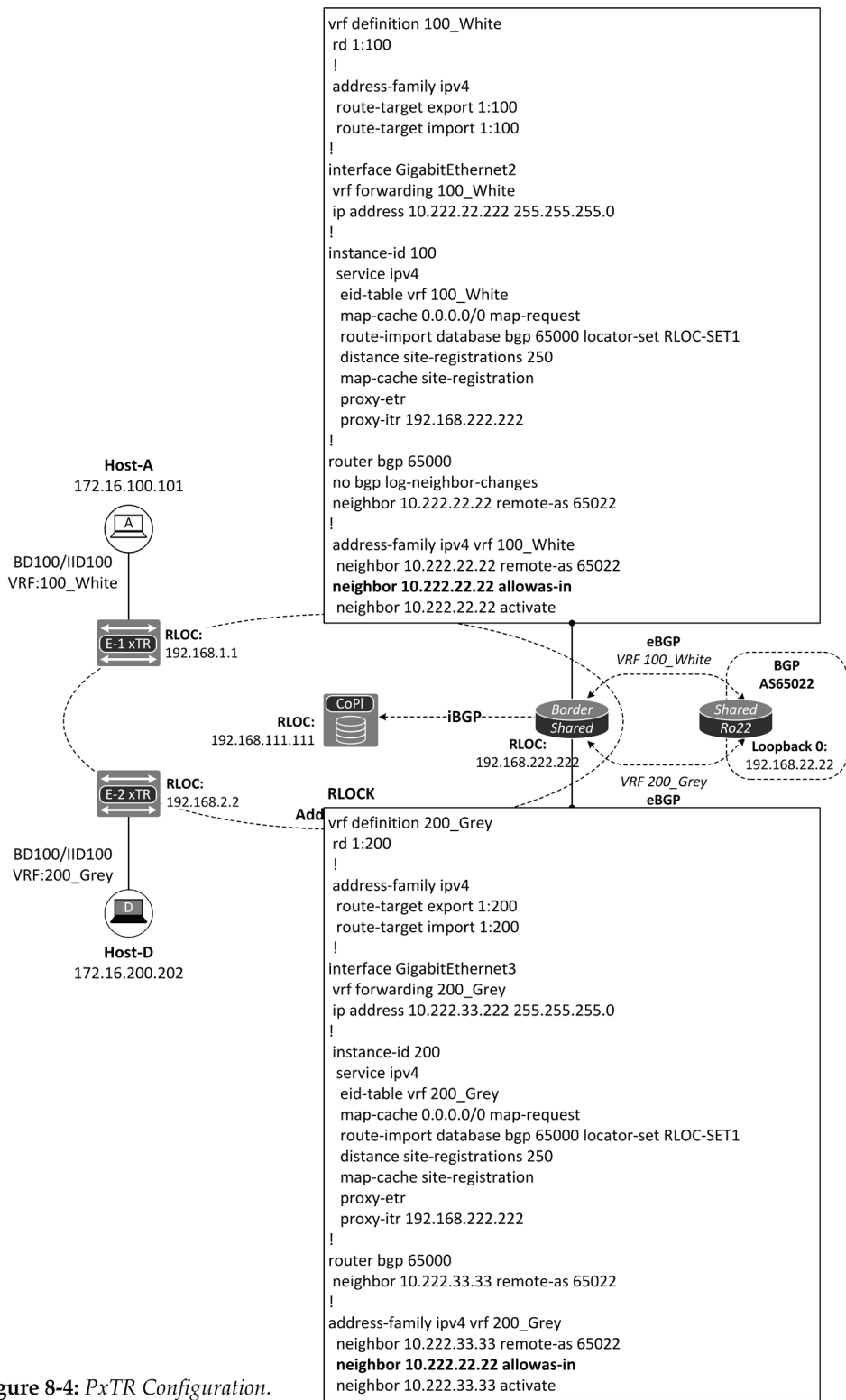


Figure 8-4: PxTR Configuration.

C-P Egress Operation: LISP Registration

Figure 8-5 on the next page shows the EID-to-RLOC mapping database of Map-Server after both E1xTR and E2xTR are registered their connected EIDs to it. EID-Prefix 172.16.200.202/32 has IID 200 and E2xTR as RLOC. EID-Prefix 172.16.100.101/32 has IID 100 and E1xTR as RLOC. EIDs are allowed to install into Mapping-Table since they fit in the dynamic EID range defined under Instances. The lifetime of entries is 24 hours.

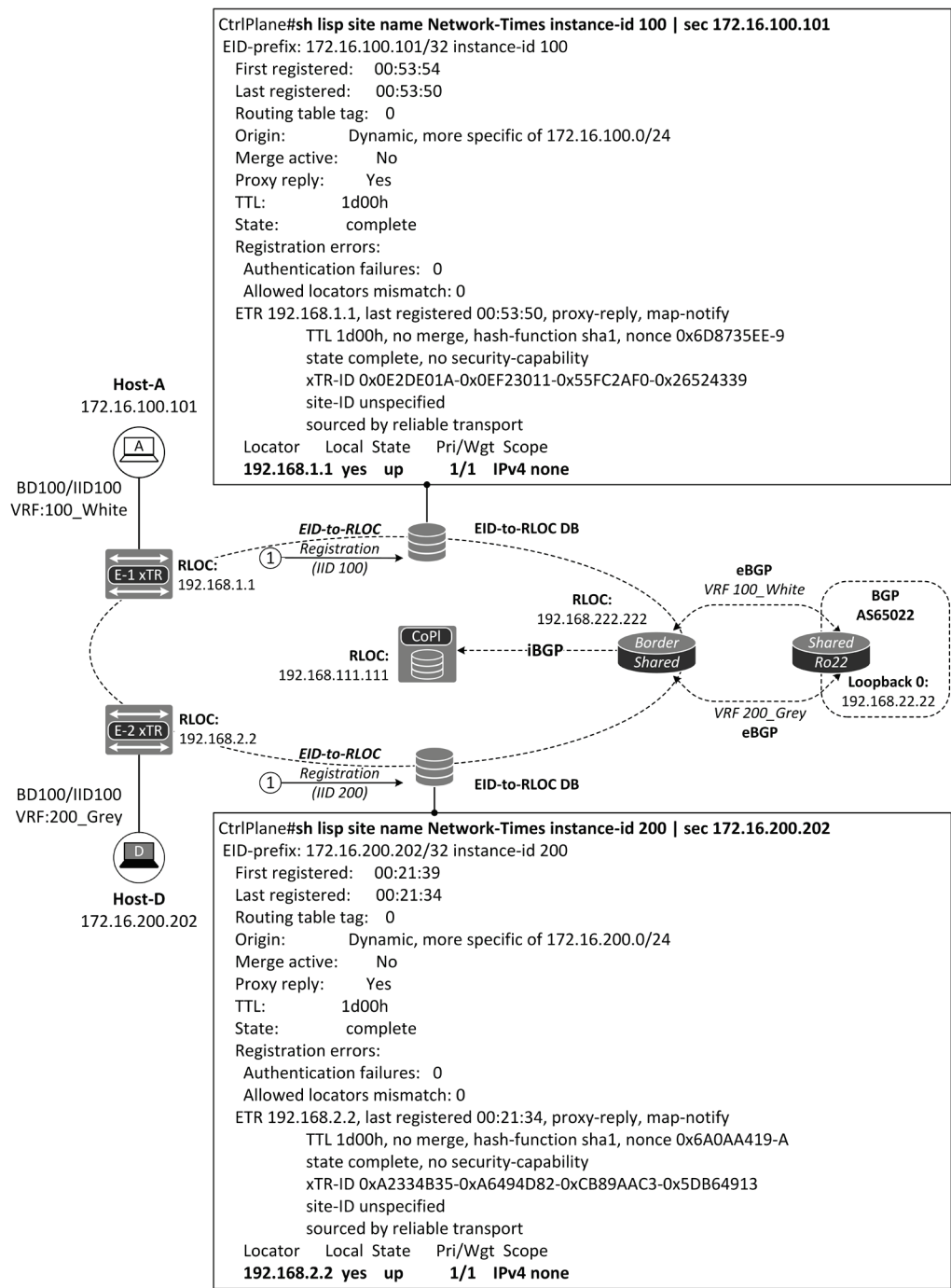


Figure 8-5: LISP Map-Cache on Map-Server.

C-P Egress Operation: BGP-Update process

Figure 8-6 on the next page explains the process where Map-Server installs the EID reachability information from the EID-to-RLOC mapping-table into VRF specific RIB.

- Phase 2.** EID 172.16.100.101/32 is exported from the EID-to-RLOC mapping database to VRF 100_White RIB as the LISP route (l). The next-hop is Null0. In addition, there is a route to subnet 172.16.100.0/24 that is also routed to Null0. This entry is created from the BGP aggregation. The Null routes prevent the forwarding loops in case that Control Plane Node is used as next-hop-router for destination hosts in 172.16.100.0/24
- Phase 3.** The route is exported from the RIB to the BGP table. Host routes are suppressed (s>) from BGP-Updates due to command **aggregate-address summary-only** BGP peer specific configuration command.
- Phase 4.** Control Plane Node sends a BGP-Update message to Border-PxTR.

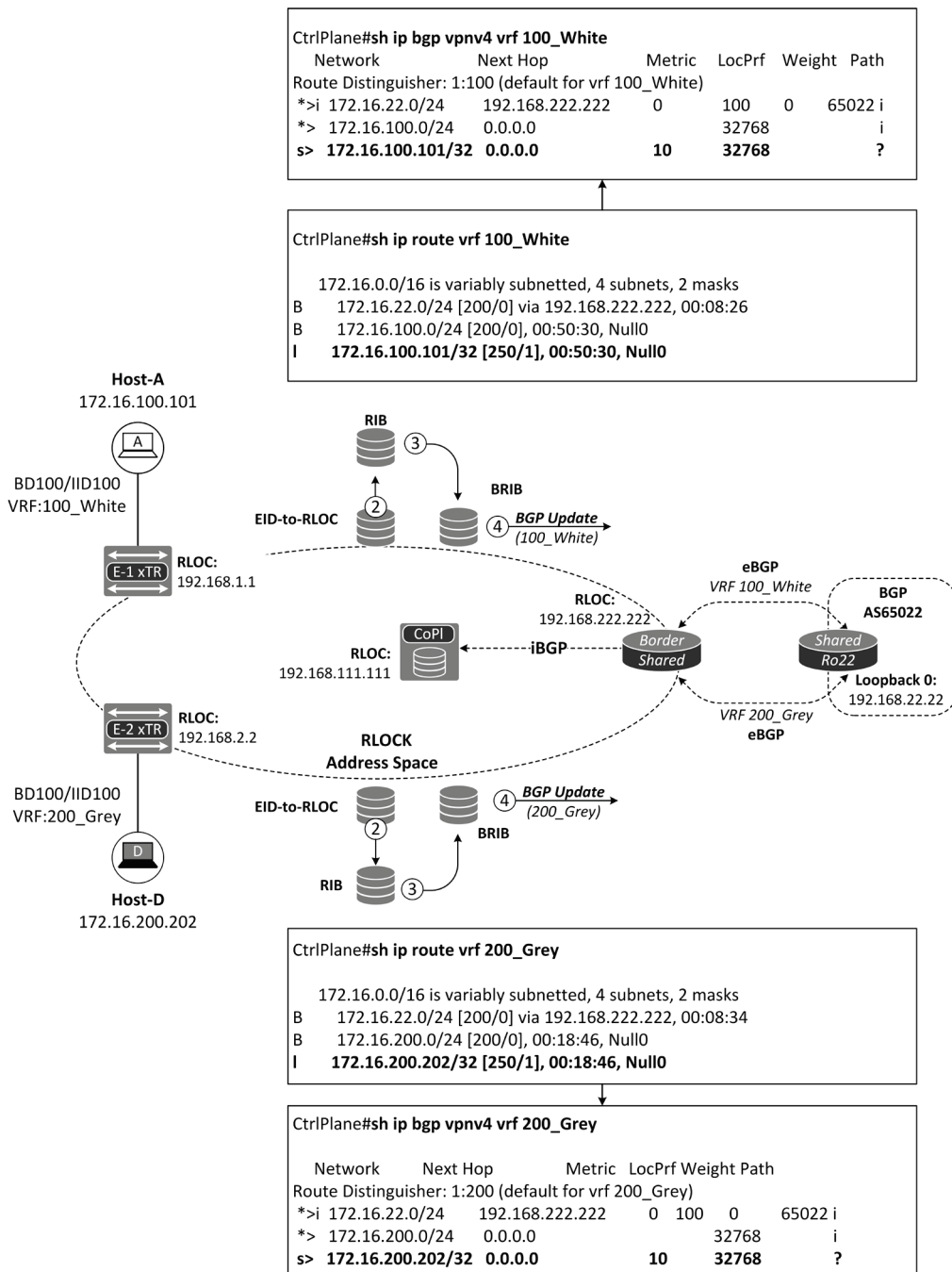


Figure 8-6: *BGP and RIB tables on Control Plane Node.*

C-P Egress Operation: BGP-Updates: PxTR

Figure 8-7 shows the BGP table on Border-PxTR. (Note that the prompt is Border-Shared). Both routes are installed into VRF specific BGP tables with Control Plane Node as a next-hop router. Border-PxTR advertises these routes to Shared-Ro22 over the VRF specific eBGP peering.

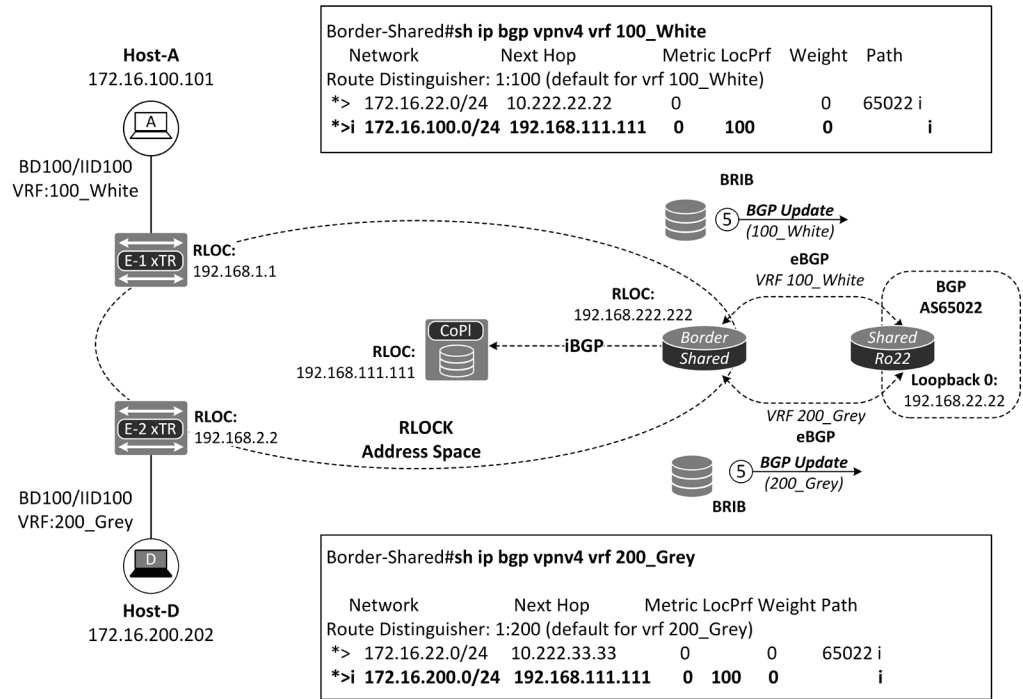


Figure 8-7: BGP tables on Border-PxTR.

C-P Egress Operation: BGP-Updates: non-LISP router

Figure 8-8 shows that Shared-Ro22 has received BGP-Updates messages from Border-PxTR and installed them into the global BGP Loc-RIB. When BGP-Updates are sent to an external BGP peer, the next-hop Path-Attribute is changed to egress interface IP address of sending BGP speakers., so the IP address 10.222.22.222 is used as a next-hop for 172.16.100.0/24 and the IP address 10.222.33.222 is used as a next-hop for 172.17.200.0/24.

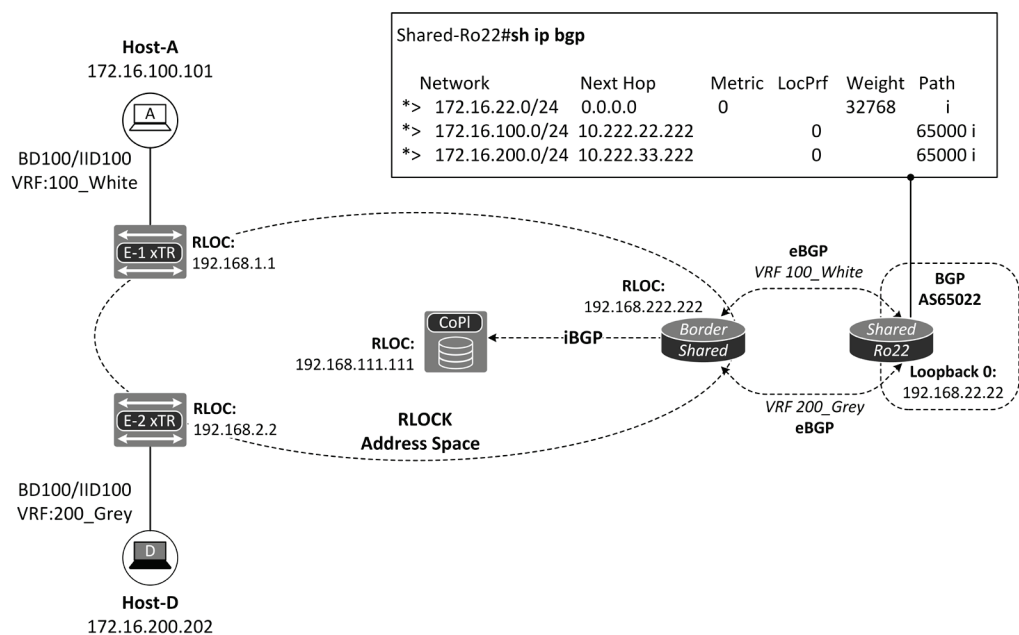


Figure 8-8: BGP table on Shared-Ro22.

C-P Ingress Operation: BGP-Updates: PxTR

Figure 8-9 explains the route propagation when Shared-Ro22 advertises routes back to Campus Fabric. The process is explained by using VRF 200_Grey route 172.16.200.0/24.

- Phase 6.** Border-PxTR receives the BGP-Update message about 172.16.200.0/24 from the non-LISP router Shared-Ro22. The BGP-Update message is received over VRF 100_White peering and the route is installed into VRF 100_White BGP Loc-RIB.
- Phase 7.** The route is installed into VRF 100_White RIB.
- Phase 8.** The route is then imported from VRF 100_White RIB to the LISP process.
- Phase 9.** Border-PxTR sends a LISP Map-Register messenger to Map-Server.

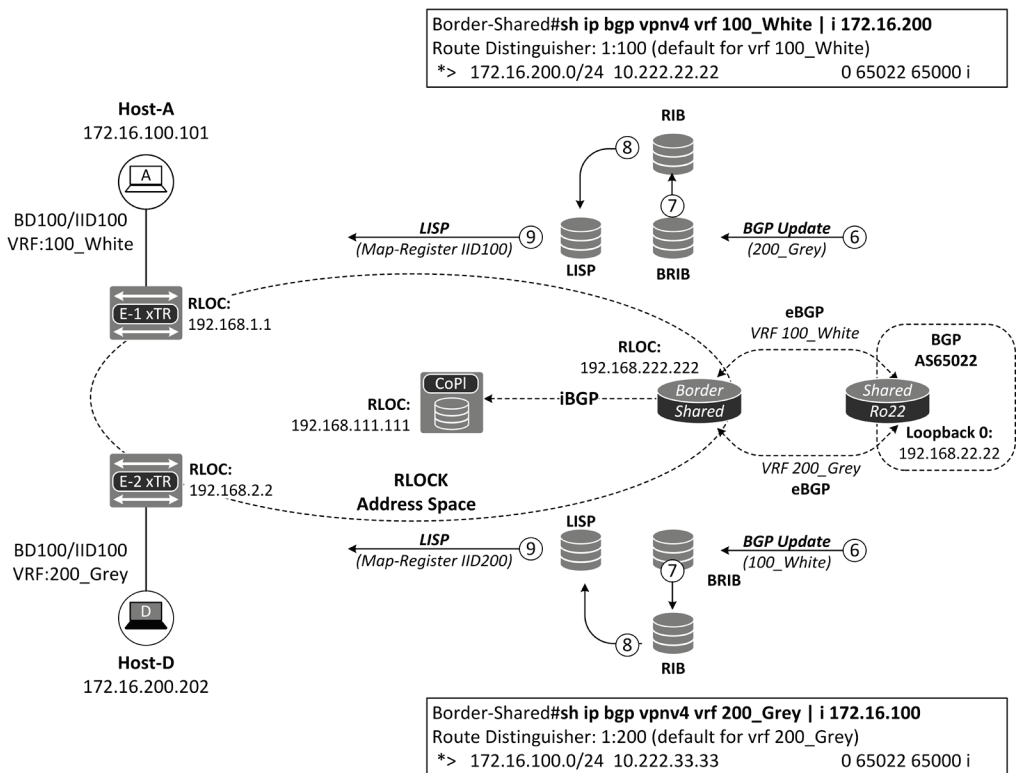
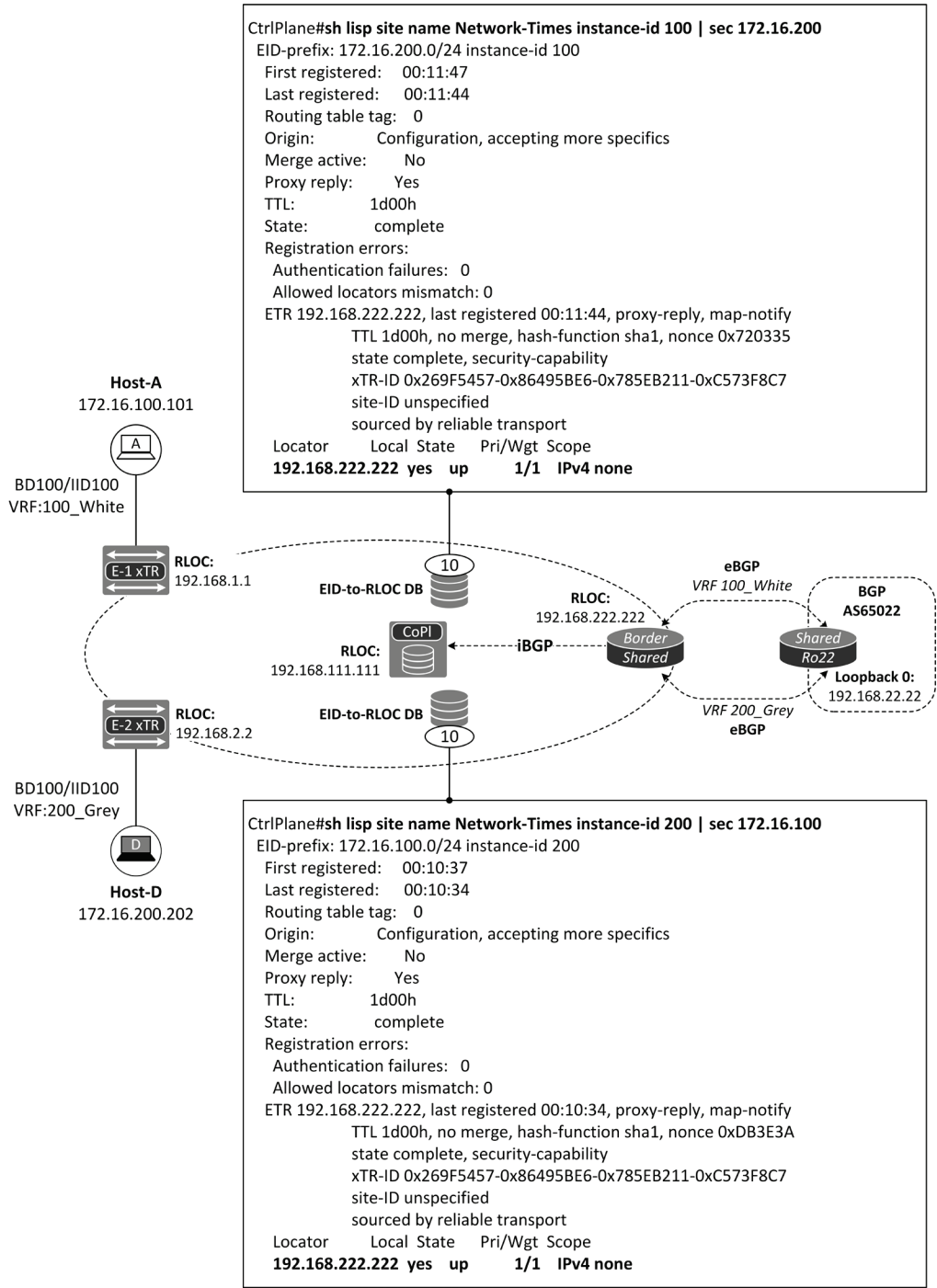


Figure 8-9: BGP-to-LISP Import in Border-PxTR.

C-P Ingress Operation: BGP-Updates: Border-PxTR

Figure 8-10 shows the EID-to-RLOC mapping-database. EID-Prefix 172.16.200.0/24 is installed into instance-Id 100 table under site IPv4 service in site Network-Times. The Locator IP address is the RLOC of advertising LISP router Border-PxTR. From the Instance-Id 100 perspective, network 172.16.200.0/24 is Campus Fabric external network.

This solution can be used not only for Inter-VN connection inside Campus Fabric but also to connect two geographically dispersed sites. The other use case is Hub-and-Spoke topology where one site is the central site and traffic between remote sites is routed via the central site.



Data-Plane Operation

Figure 8-11 describes the Map-Request process when Host-A (172.16.100.101/VRF 100_White) starts pinging Host-D (172.16.200.202/VRF 200_Grey). Note that figure 8-11 also shows the same process used by Host-D

- Phase 1.** Host-A, send the first ICMP-Request to E1xTR.
- Phase 2.** First, E1xTR tries to resolve the next-hop router from the VRF 100_White RIB. Because there is no routing information in the RIB, Edge-1 makes a LISP Mapping-Cache lookup to resolve the RLOC for the destination. At this phase, the Mapping-Cache does not have EID entry.
- Phase 3.** E1xTR tries to figure out the EID-to-RLOC mapping information from Map-Resolver by sending a Map-Request message wrapped inside the LISP Encapsulated Control Plane message.
- Phase 4.** Map-Resolver consults the EID-to-RLOC mapping database published by the Map-Server. Map-Server has received the mapping information from Border-PxTR and the requested EID and its authoritative RLOC are known. Map-Resolver sends a LISP Map-Reply message to ITR E1xTR. E1xTR installs the mapping information into its LISP mapping-Cache with RLOC pointing to Border PxTR. At this stage, E1xTR is able to forward IP packets to EID 172.16.200.202/32.

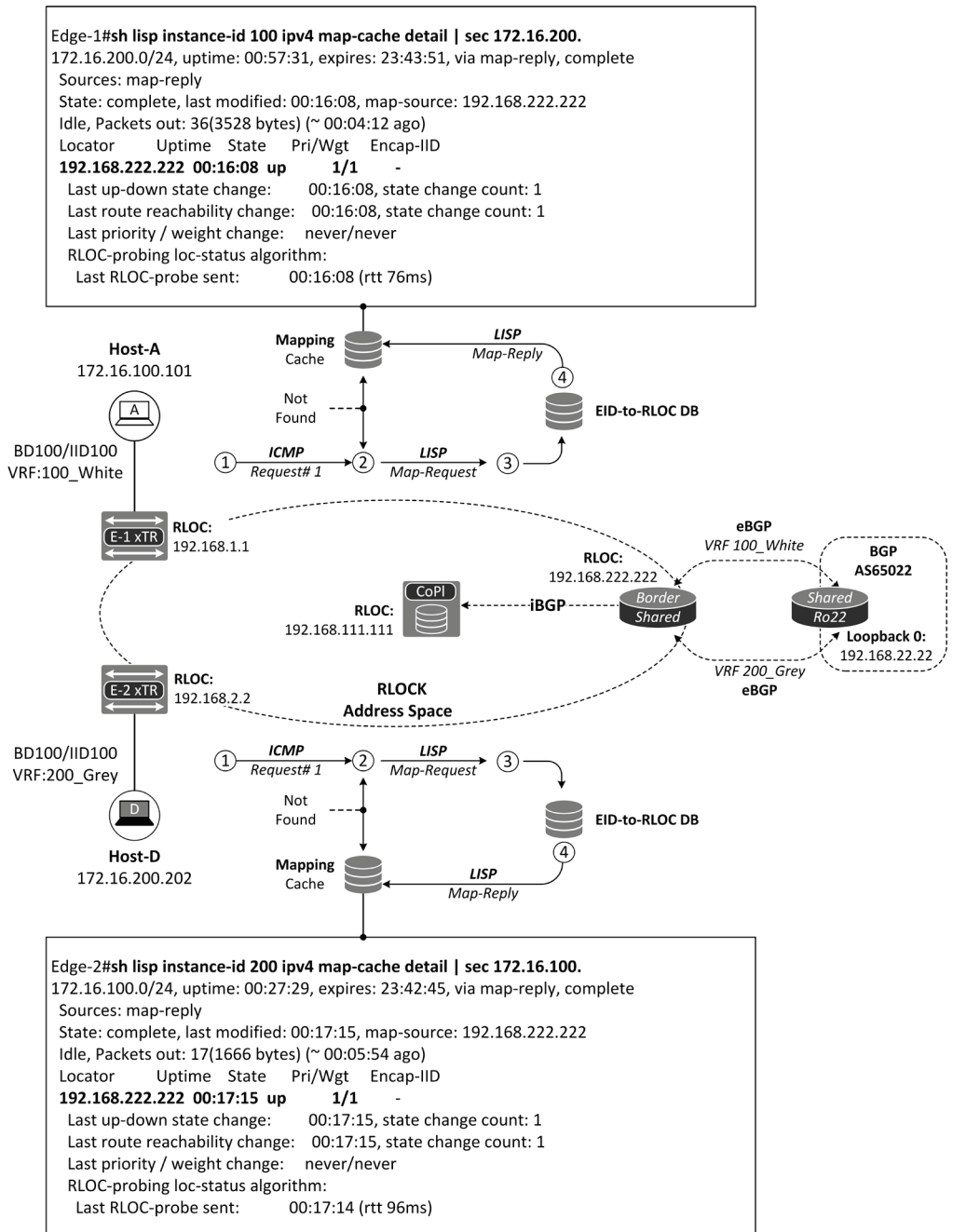


Figure 8-11: Map-Requests by ITRs.

Figure 8-12 shows the second ICMP-Request processing of Host-A.

- Phase 1.** E1xTR receives the ICMP-request from Host-A. This time the LISP Mapping-Cache lookup is hit.
- Phase 2.** Campus Fabric data-plane uses VXLAN encapsulation. E1xTR copies the Instance-Id 100 from LISP Mapping-Cache to the VXLAN header VN field. It sets the outer destination IP address to RLOC address 192.168.222.222 found from the Mapping-Cache. It uses its own RLOC IP address as a source IP address in the outer IP header. Core-Ro11 makes it routing decisions based on the outer IP header and forwards the packet to Border-PxTR.
- Phase 3.** Border-PxTR receives the encapsulated ICMP-request message. Based on the destination IP address in the tunnel header, it knows that it should process the packet. It verifies the VN-Id from the VXLAN header and then removes the VXLAN encapsulation. Based on VN-Id 100, Border-PxTR knows that the packet has to be routed based on VRF 100_White RIB. It sends the original ICMP-Request message to Shared-Ro22.
- Phase 4.** Shared-Ro22 receives the packet from Border-PxTR via interface Gi2 (VRF 100_White eBGP peering) and does routing lookup from the global RIB.
- Phase 5.** Shared-Ro22 routes packet back to Border-PxTR but now using the interface Gi3 (VRF 200_Grey eBGP peering).
- Phase 6.** Border-PxTR receives the ICMP-Request message from the interface attached to VRF 200_Grey. This process generates the LISP Map-Request for destination EID. This process follows the same model as what was explained when E1xTR receives the first ICMP-Request from Host-A.

Figure 8-12: *Data-Plane Operation.*

Phase 9. E2xTR receives the encapsulated IP-Packet. Based on the destination IP address in the outer IP header, it knows that it should process the IP packet. It removes the tunnel encapsulation and makes a forwarding decision based on the inner IP address and the VXLAN header VN-Id value 200.

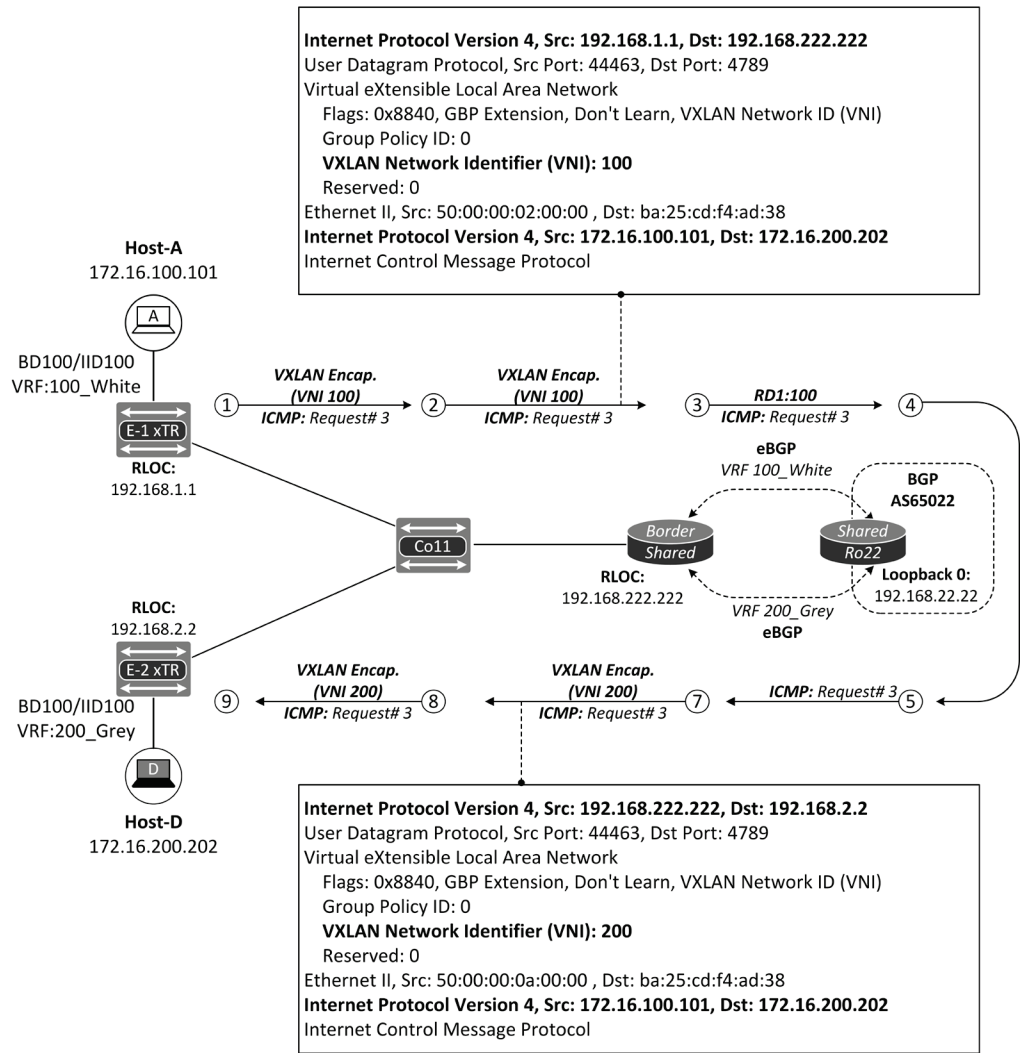


Figure 8-13: Data-Plane Operation.

Summary

Figure 8-14 summarizes the external connection use case. Shared-PxTR can be used for 1) connection to shared service. 2) Connectin two or more sites over non-LISP infrastructure. 3) The connection between different VRFs/Instances inside Campus Fabric. 4) Connection to untrusted networks such, as the Internet.

The EIDs behind Shared-Ro22 PxTR can be resolved via LISP Map-Request messages while the EIDs behind Inet-PxTR are used by sending unresolved destination EIDs to Proxy-ETR.

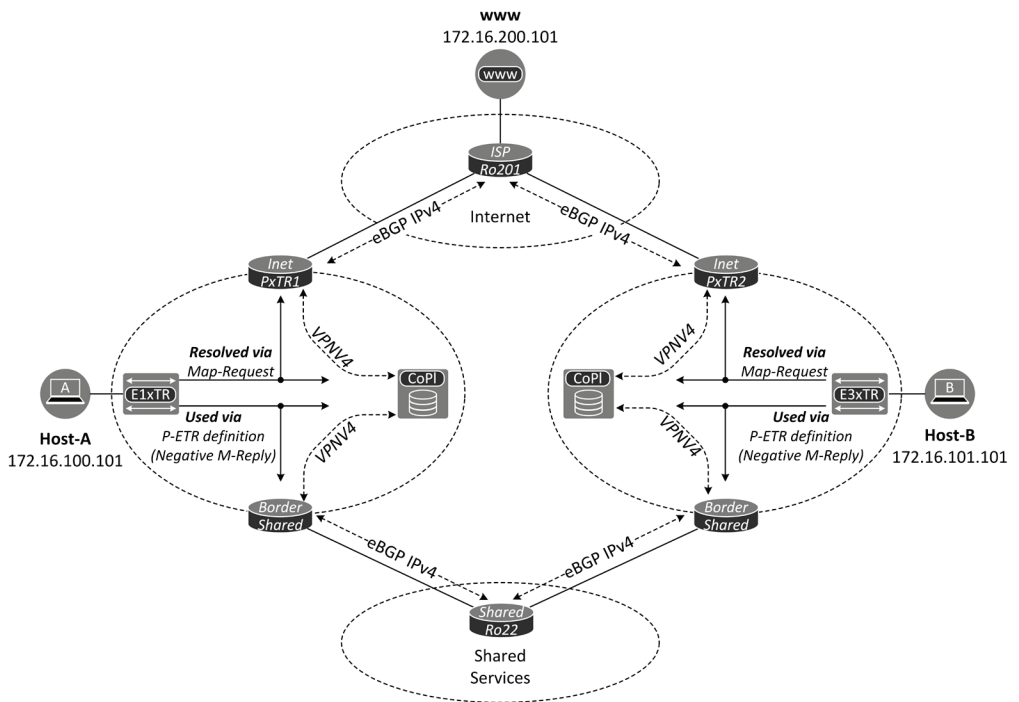


Figure 8-14: External connection summary.

References

- RFC 4364 E. Rosen and Y. Rekhter HMAC: “BGP/MPLS IP Virtual Private Networks (VPNs)”, RFC 4364, February 2006.

Chapter 9: Campus Fabric Redundancy

Introduction

This chapter discusses of LISP redundancy solution in Campus Fabric. The first section explains the LISP Mapping-System redundancy solution, where there are two Control Plane nodes. The second section introduces the dual-homed Proxy-xTR model.

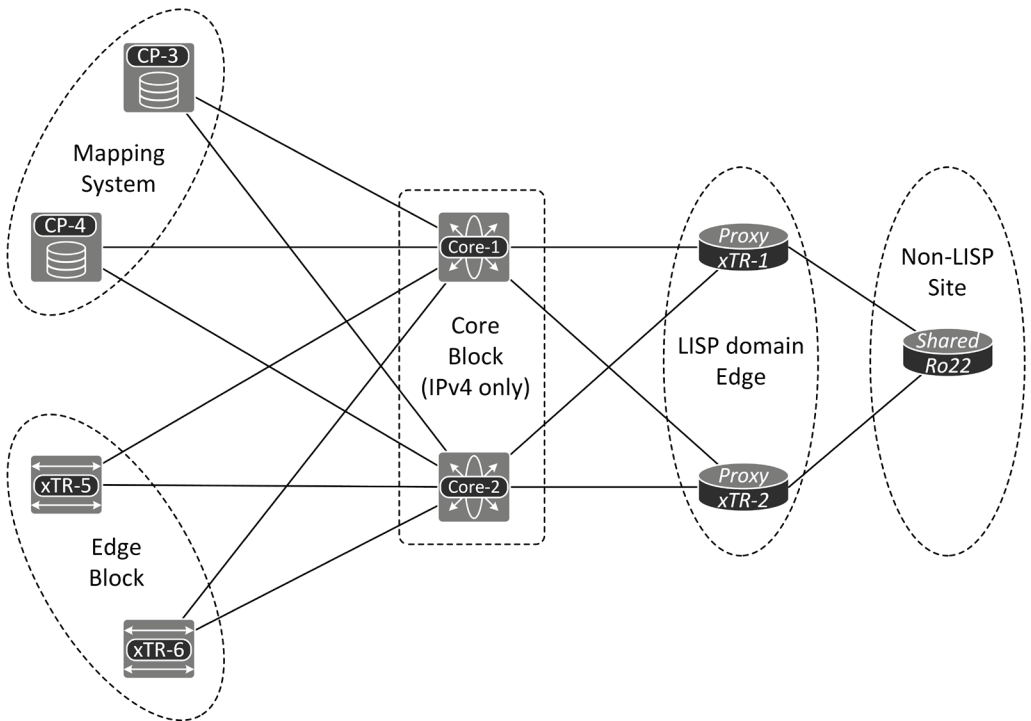


Figure 9-1: Building Blocks.

CP Node Redundancy: Anycast-IP RLOC

Figure 9-2 shows the Mapping System model with two Control Plane nodes (CP-3 and CP-4). Both nodes share the same RLOC Anycast-IP address 192.168.1.34. However, CP-3 uses mask /31 while CP-4 uses mask /30. This actually means that they, even though sharing the same Anycast-IP address, advertise the different subnet. Subnet advertised by CP-3 is 192.168.1.34/31 while the subnet advertised by CP-4 is 192.168.1.32/30. In real life, it might be a good idea to choose the Anycast-IP address that fits into the same subnet even though the mask length is different. In addition to node redundancy solution, both CP-nodes are connected to both Core-1 and Core-2 switches.

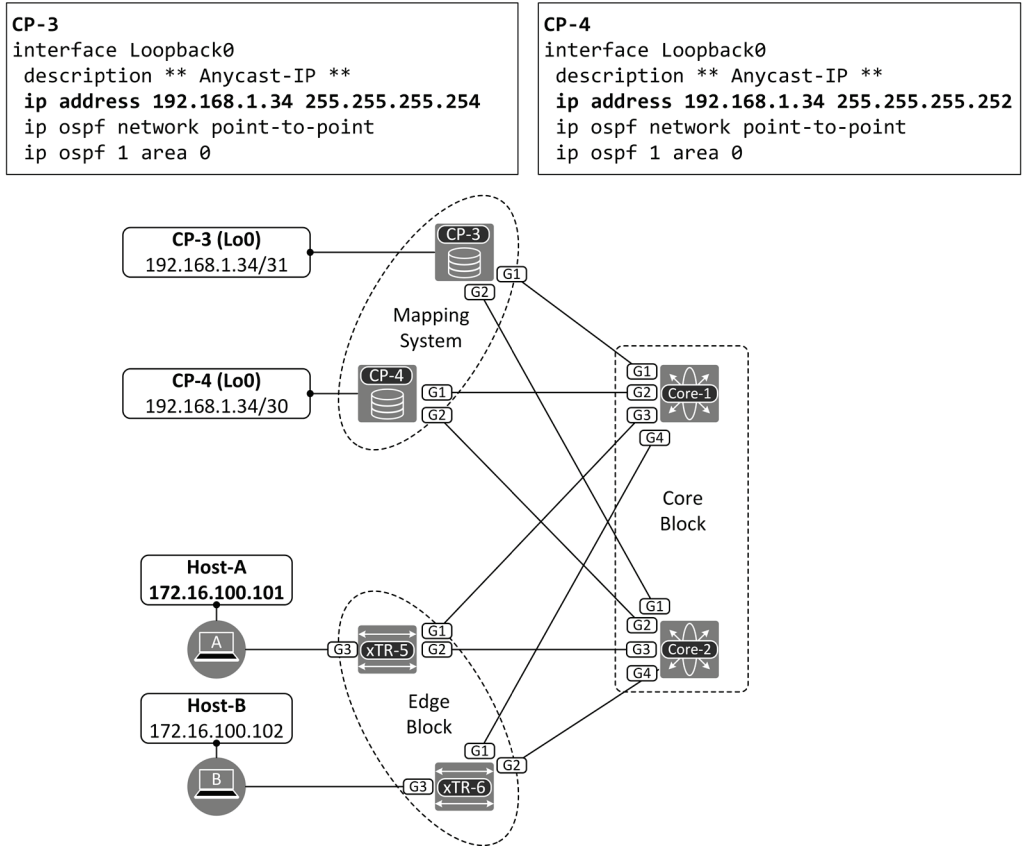


Figure 9-2: Anycast-IP address on CP-3 and CP-4.

Both advertised subnets 192.168.1.32/30 and 192.168.1.34/31 eventually end up to RIBs of xTR switches. However, only 192.168.1.34/31 is used for routing packets to 192.168.1.34. Figure 9-3 shows the RIB of xTR-5 concerning subnets and also the individual route to 192.168.1.34. The destination 192.168.1.34 is load-balanced over both Core switches (advertised by 192.168.3.3 – CP-3).

```
R5-xTR1#sh ip route 192.168.1.32 longer-prefixes
```

```
<snipped>
```

```
0          192.168.1.32/30 [110/3] via 10.2.5.2, 00:01:56, GigabitEthernet2
                                [110/3] via 10.1.5.1, 00:01:52, GigabitEthernet1
0          192.168.1.34/31 [110/3] via 10.2.5.2, 00:01:56, GigabitEthernet2
                                [110/3] via 10.1.5.1, 00:01:52, GigabitEthernet1
```

```
R5-xTR1#sh ip route 192.168.1.34
```

```
Routing entry for 192.168.1.34/31
```

```
Known via "ospf 1", distance 110, metric 3, type intra area
```

```
Last update from 10.1.5.1 on GigabitEthernet1, 00:04:23 ago
```

```
Routing Descriptor Blocks:
```

```
* 10.2.5.2, from 192.168.3.3, 00:04:23 ago, via GigabitEthernet2
```

```
Route metric is 3, traffic share count is 1
```

```
10.1.5.1, from 192.168.3.3, 00:04:23 ago, via GigabitEthernet1
Route metric is 3, traffic share count is 1
```

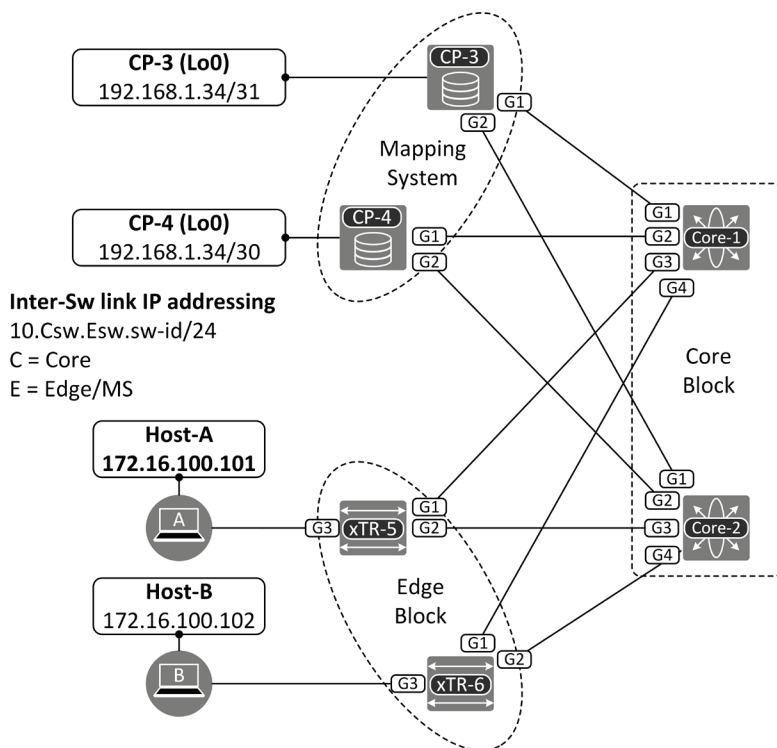


Figure 9-3: The RIB entries of xTR-5.

When xTR switches register their local EIDs by sending Map-Register messages to Map-Server 192.168.1.34, the message is forwarded only to CP-3. Figure 9-4 shows that the EIDs of Host-A (172.16.100.101) and Host-B (172.16.100.102) are registered and installed only into CP-3 EID-to-RLOC Mapping database. There is no Mapping information in the Mapping Database of CP-4.

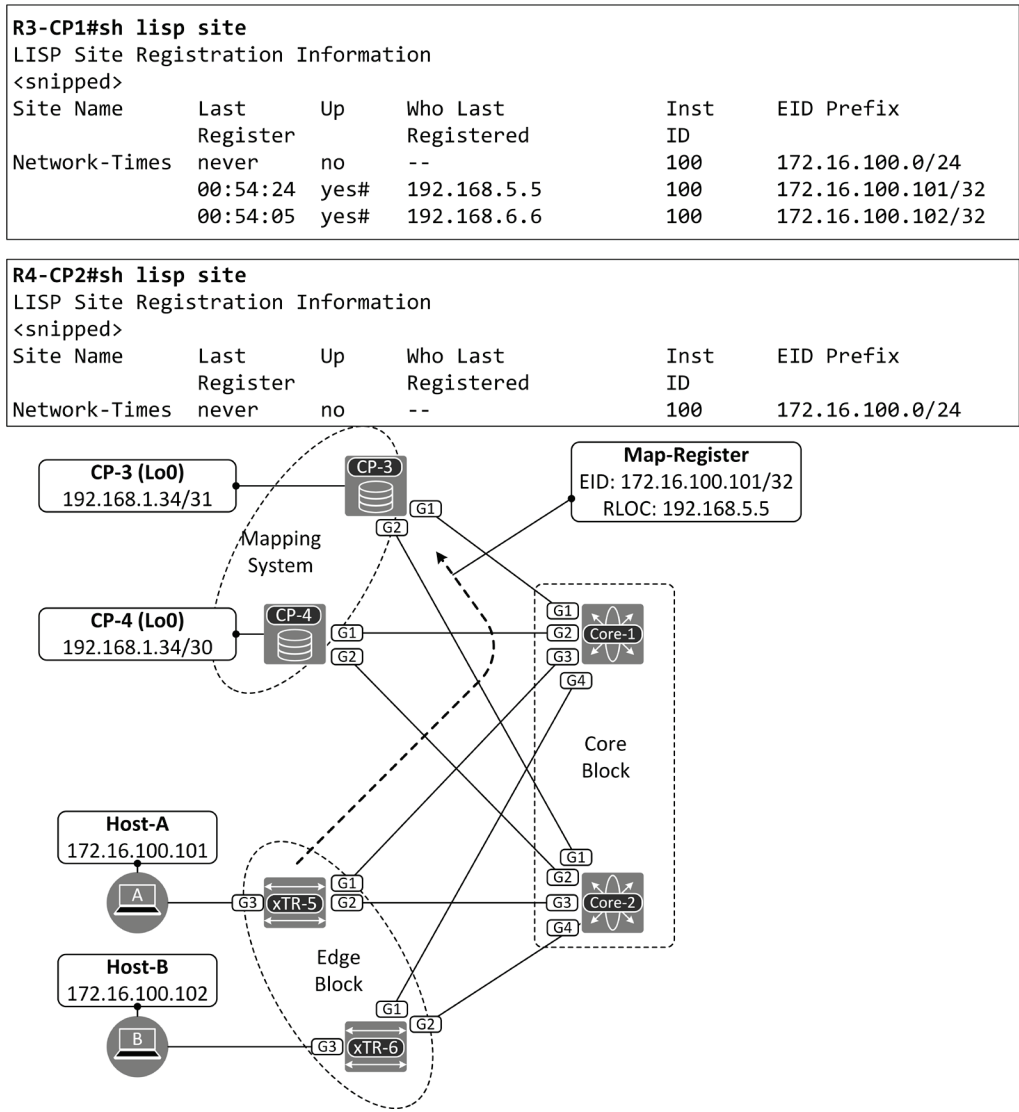


Figure 9-4: The Mapping Databases of CP-3 and CP-4.

CP Node Redundancy: Core switch out of service

The dual-homed CP nodes make it possible to take one of the Core switches out of the service in a controlled manner without disturbing data traffic. Figure 9-5 illustrates the situation where Core-1 is taken out of the data-path by shutting down the OSPF process 1. As a reaction to this event, Core-1 flushes all of its Router LSAs by using LSA age 3600 seconds. Then it shuts down its Loopback 0 interface used as OSPF Router-ID. This way it removes itself from the data-path and its OS can be updated, as an example.

```

R1-Core1(config)#router ospf 1
R1-Core1(config-router)#shutdown
OSPF-1 LSGEN: Flushing External LSAs
OSPF-1 LSGEN: Flushing Opaque AS LSAs
OSPF-1 LSGEN: Flushing Link states in area 0
OSPF-1 ADJ   Lo0: Interface going Down
OSPF-1 ADJ   Lo0: 192.168.1.1 address 192.168.1.1 is dead, state DOWN
OSPF-1 ADJ   Gi3: Interface going Down
OSPF-1 ADJ   Gi3: 192.168.5.5 address 10.1.5.5 is dead, state DOWN
%OSPF-5-ADJCHG: Process 1,
      Nbr 192.168.5.5 on GigabitEthernet3 from FULL to DOWN,
      Neighbor Down: Interface down or detd
OSPF-1 ADJ   Gi3: 192.168.1.1 address 10.1.5.1 is dead, state DOWN
  
```

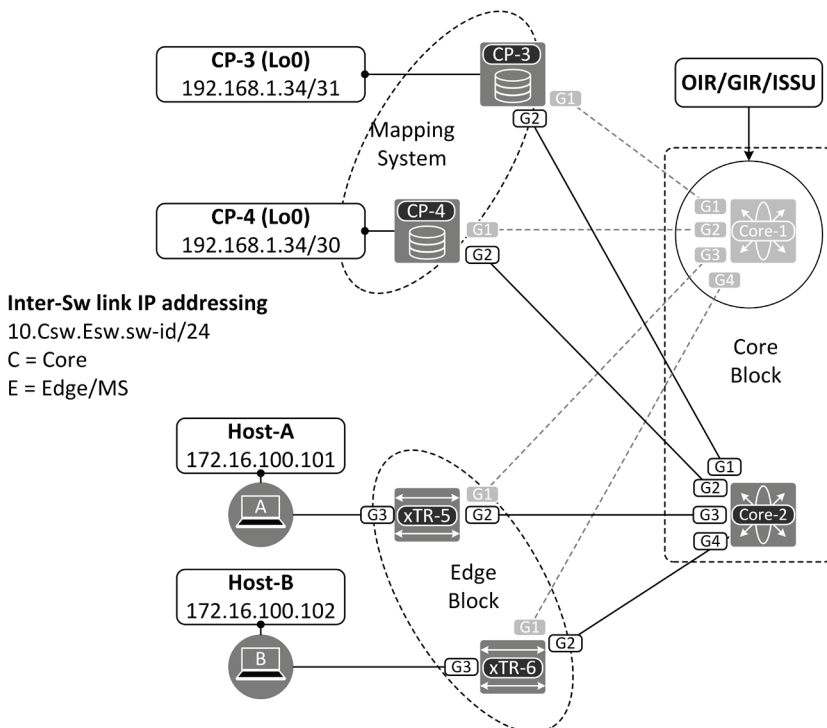


Figure 9-5: Removing Core-1 from the Data-Path.

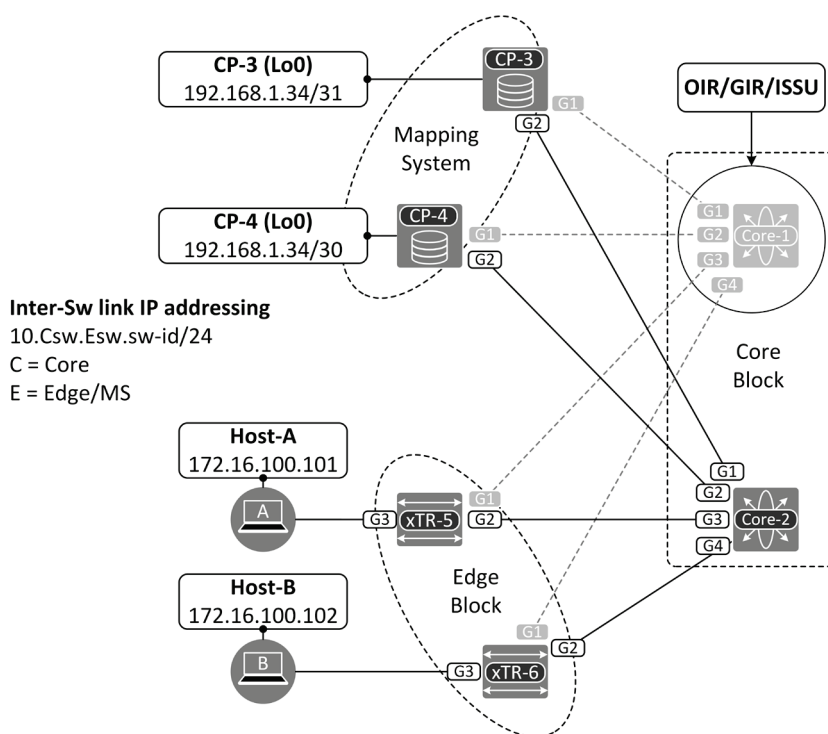
Capture 9-1 shows the Router-LSA sent out of the interface G3 (link to xTR-5) by Core-1 after shutting down the OSPF process 1.

Note, there is no OSPF optimization used in the example network used in this chapter, that is why also all transport networks are included in.

```
Internet Protocol Version 4, Src: 10.1.5.1, Dst: 224.0.0.5
Open Shortest Path First
  OSPF Header
  LS Update Packet
    Number of LSAs: 1
    LSA-type 1 (Router-LSA), len 132
      .000 1110 0001 0000 = LS Age (seconds): 3600
      0... .. = Do Not Age Flag: 0
      Options: 0x22, (DC) Demand Circuits, (E) External Routing
      LS Type: Router-LSA (1)
      Link State ID: 192.168.1.1
      Advertising Router: 192.168.1.1
      Sequence Number: 0x80000008
      Checksum: 0xc35a
      Length: 132
      Flags: 0x00
      Number of Links: 9
        Type: Stub      ID: 192.168.1.1      Data: 255.255.255.255 Metric: 1
        Type: PTP       ID: 192.168.6.6      Data: 10.1.6.1        Metric: 1
        Type: Stub      ID: 10.1.6.0        Data: 255.255.255.0   Metric: 1
        Type: PTP       ID: 192.168.5.5      Data: 10.1.5.1        Metric: 1
        Type: Stub      ID: 10.1.5.0        Data: 255.255.255.0   Metric: 1
        Type: PTP       ID: 192.168.4.4      Data: 10.1.4.1        Metric: 1
        Type: Stub      ID: 10.1.4.0        Data: 255.255.255.0   Metric: 1
        Type: PTP       ID: 192.168.3.3      Data: 10.1.3.1        Metric: 1
        Type: Stub      ID: 10.1.3.0        Data: 255.255.255.0   Metric: 1
```

Capture 9-1: Router-LSA sent by Core-1.

Figure 9-6 shows that after shutting down the OSPF process 1 in Core-1, xTR-5 still uses CP-3 as Map-Server and Map-Resolver but now there is only one path via Core-2.



```
R5-xTR1#sh ip route 192.168.1.34
Routing entry for 192.168.1.34/31
  Known via "ospf 1", distance 110, metric 3, type intra area
  Last update from 10.2.5.2 on GigabitEthernet2, 00:50:29 ago
Routing Descriptor Blocks:
  * 10.2.5.2, from 192.168.3.3, 00:50:29 ago, via GigabitEthernet2
    Route metric is 3, traffic share count is 1
```

Figure 9-6: *xTR-5 RIB After Shutting Down the OSPF Process 1 in Core-1.*

CP-Node Redundancy: Primary CP Node Out of Service

Next, Core-1 is taken back to service. Figure 9-7 shows the effect on data forwarding when the primary Control Plane node CP-3 is not reachable. Failure event is simulated by shutting down the interface Loopback 0 used as RLOC address while pinging the IP 192.168.1.34 from the xTR-5 at the same time. We can see that the ping success present is still 100% so we did not lose any of the 2000 ICMP-packets sent during the switch-over process. Figure 9-7 also verifies that the Anycast-IP address is now advertised by CP-4 (192.168.4.4).

```
R5-xTR1#sh ip route 192.168.1.34
Routing entry for 192.168.1.34/31
<snipped>
* 10.2.5.2, from 192.168.3.3, 00:00:29 ago, via GigabitEthernet2
  Route metric is 3, traffic share count is 1
  10.1.5.1, from 192.168.3.3, 00:00:29 ago, via GigabitEthernet1
    Route metric is 3, traffic share count is 1
```

```
R5-xTR1#ping 192.168.1.34 repe 2000
Type escape sequence to abort.
Sending 2000, 100-byte ICMP Echos to 192.168.1.34, timeout is 2 seconds:
!!!!!!!!!!<snipped>
Success rate is 100 percent (2000/2000), round-trip min/avg/max = 2/3/17 ms
```

```
R5-xTR1#sh ip route 192.168.1.34
Routing entry for 192.168.1.32/30
<snipped>
* 10.2.5.2, from 192.168.4.4, 03:25:50 ago, via GigabitEthernet2
  Route metric is 3, traffic share count is 1
  10.1.5.1, from 192.168.4.4, 00:03:55 ago, via GigabitEthernet1
    Route metric is 3, traffic share count is 1
```

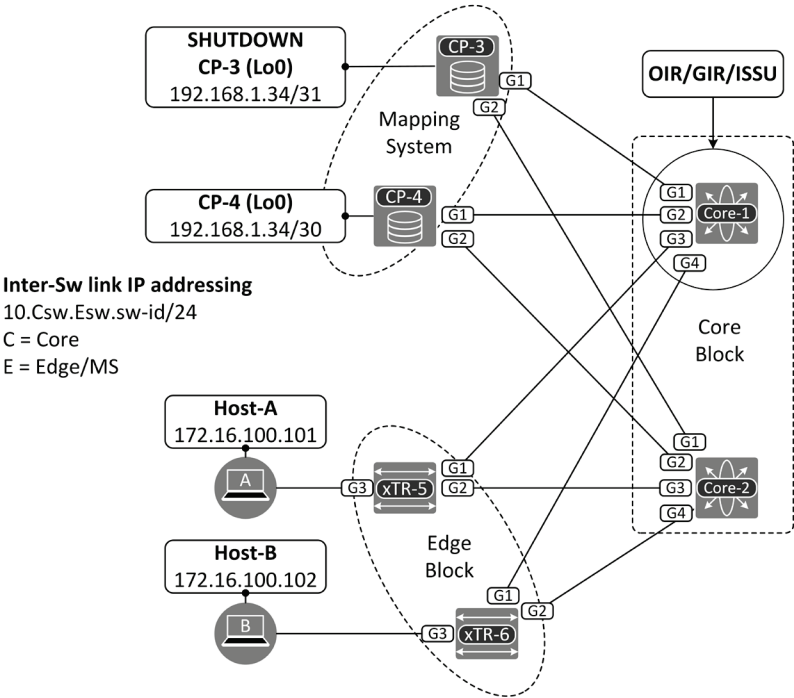


Figure 9-7: Primary Control Plane Node CP-3 Out of Service.

Figure 9-8 gives an overview of events related to Control Plane node switch-over. The detailed process is explained after the figure. (1) The Loopback 0 used as an RLOC of Map-Server is manually shut down on CP-3. (2) CP-3 continues the TCP-Probing process by sending a TCP-Probe to xTR-5. (3) xTR-5 replies with TCP-Probe, which however ends up to CP-4 due to routing convergence. (4) CP-4 sends TCP Reset to xTR-5 because it does not have an open TCP connection with xTR-5. (5) As a reaction, xTR-5 sends the LISP Map-Register message to CP-4. This way CP-4 EID-to-RLOC Mapping database is updated.

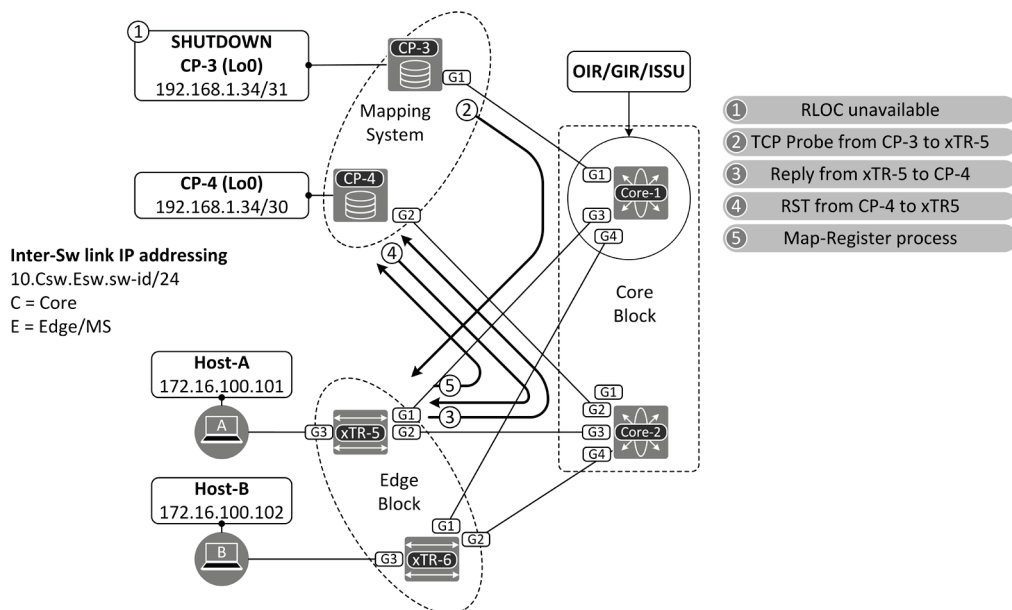


Figure 9-8: CP-4 EID-to-RLOC Database Update Process.

Phase 1: Interface Loopback 0 down

Interface Loopback 0 used as RLOC is shut down manually on CP-3.

```
R3-CP1(config)# interface loopback 0
R3-CP1(config-if)# shutdown
*Jan 27 20:38:28.907: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Loopback0, changed state to down
*Jan 27 20:38:28.908: %LINK-5-CHANGED: Interface Loopback0, changed
state to administratively down
```

Example 9-1: CP-3 Loopback 0 Down

Phase 2: CP-3 TCP Probe from CP-3 to ETRs

CP-3 continues TCP-Probing every 60 seconds by default. It sends TCP-Probe with ACK-flag set to xTR-5 using sequence number 1712500512.

```
Internet Protocol Version 4, Src: 192.168.1.34, Dst: 192.168.5.5
Transmission Control Protocol, Src Port: 4342, Dst Port: 41216, Seq:
1, Ack: 1, Len: 0
    Source Port: 4342
    Destination Port: 41216
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 1      (relative sequence number)
    Sequence number (raw): 1712500512
    [Next sequence number: 1      (relative sequence number)]
    Acknowledgment number: 1      (relative ack number)
    Acknowledgment number (raw): 3402943904
    0101 .... = Header Length: 20 bytes (5)
    Flags: 0x010 (ACK)
    Window size value: 16254
<snipped>
```

Capture 9-2: CP-3 TCP-Probe to xTR-5

Phase 3: xTR-5 send TCP-ACK

The router xTR-5 replies with TCP-Probe with ACK-Flag set and with acknowledge number increased by one (1712500513). The IP address 192.168.1.34 is now routed to CP-4, which means that the TCP-ACK ends up to CP-4.

```
Internet Protocol Version 4, Src: 192.168.5.5, Dst: 192.168.1.34
Transmission Control Protocol, Src Port: 41216, Dst Port: 4342, Seq:
1, Ack: 1, Len: 0
    Source Port: 41216
    Destination Port: 4342
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 1      (relative sequence number)
    Sequence number (raw): 3402943903
    [Next sequence number: 1      (relative sequence number)]
    Acknowledgment number: 1      (relative ack number)
    Acknowledgment number (raw): 1712500513
    0101 .... = Header Length: 20 bytes (5)
    Flags: 0x010 (ACK)
    Window size value: 16262
<snipped>
```

Capture 9-3: xTR-5 replies with TCP-ACK

Phase 4: CP-4 send TCP Reset to xTR-5

CP-4 does not have an open TCP connection with xTR-5 at this moment, so it resets the connection by sending a TCP-Probe with RST bit set back to xTR-5.

```

Internet Protocol Version 4, Src: 192.168.1.34, Dst: 192.168.5.5
Transmission Control Protocol, Src Port: 4342, Dst Port: 41216, Seq:
1, Len: 0
  Source Port: 4342
  Destination Port: 41216
  [Stream index: 0]
  [TCP Segment Len: 0]
  Sequence number: 1      (relative sequence number)
  Sequence number (raw): 1712500513
  [Next sequence number: 1      (relative sequence number)]
  Acknowledgment number: 0
  Acknowledgment number (raw): 0
  0101 .... = Header Length: 20 bytes (5)
  Flags: 0x004 (RST)
  Window size value: 0
  [Calculated window size: 0]
  [Window size scaling factor: -1 (unknown)]
  Checksum: 0x613e [unverified]
  [Checksum Status: Unverified]
  Urgent pointer: 0
  [Timestamps]

```

Capture 9-4: *TCP-RST send by CP-4 to xTR-5.*

Phase 5: xTR-5 re-register its EIDs to Map-Server.

As a reaction to TCP RST, xTR-5 sends a Map-Register message to Map-Server 192.168.1.34, which now is CP-4, about its EIDs. The same process is done by xTR-6. The same process applies in reversed order when CP-3 is brought back to service.

```

Internet Protocol Version 4, Src: 192.168.5.5, Dst: 192.168.1.34
User Datagram Protocol, Src Port: 4342, Dst Port: 4342
Locator/ID Separation Protocol
  0011 .... = Type: Map-Register (3)
  .... 1... = P bit (Proxy-Map-Reply): Set
<snipped>
  .... .... 1 = M bit (Want-Map-Notify): Set
  Record Count: 1
  Nonce: 0x1b5812d523bf5afa
  Key ID: 0x0001
  Authentication Data Length: 20
  Authentication Data: 878aea274c0e58375e5124b113976655f3951cfa
  Mapping Record 1, EID Prefix: [100] 172.16.100.101/32, TTL: 1440,
  Action: No-Action, Authoritative
  xTR-ID: e74f7a6c354f29f9f4c2a5ac8bab7151

```

Capture 9-5: *LISP Map-Register message from xTR-5.*

PxTR Redundancy: Priority

LISP solution supports redundant Tunnel Routers (xTR) where traffic engineering is based on Priority/Weight values. Figure 9-9 illustrates the example topology used in this section. There are two Proxy-xTR routers which both have eBGP IPv4 peering with non-LISP router Shared-Ro22. In addition, they have iBGP VPNv4 peering with Map-Server CP-3. The BGP solution is the same that what was used in chapter 7 and chapter 8. For simplicity, there is only one Control Plane node and Core-switch. The RLOC of PxTR-1 advertises Priority 3 and Weight 1 in Map-Register messages sent to CP-3 while PxTR-2 uses Priority 2 and Wight 1. The router Shared-Ro22 uses the BGP egress policy that permits only local routes (empty AS-Path list) and BGP ingress policy that prefers routes received from PxTR-1 with weight 200. This section first shortly explains the configuration and then explains the theory part. The basic configuration is the same that is shown in Appendix B at the end of the book.

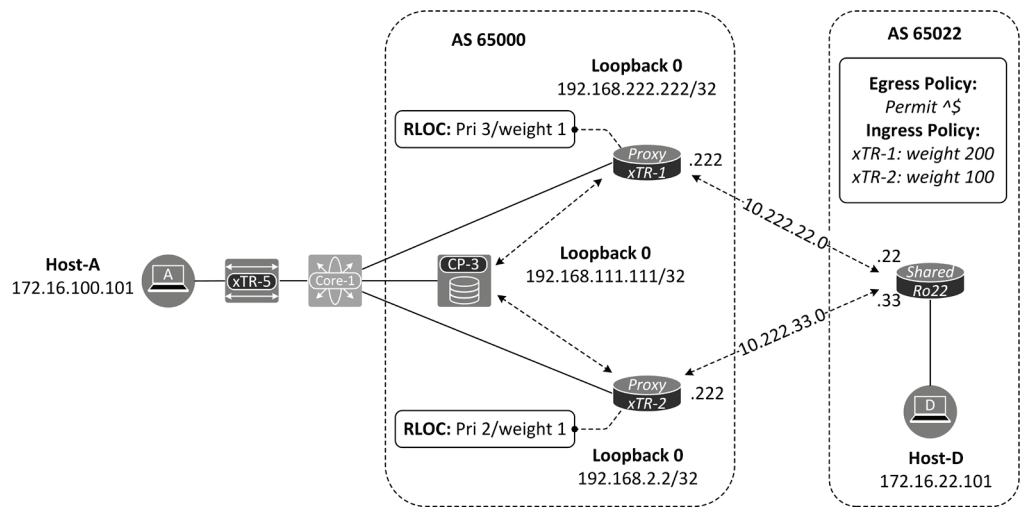


Figure 9-9: PxTR Redundancy.

PxTR Redundancy: Priority Configuration

Figure 9-10 recaps the device configurations. From the PxTR redundancy perspective, there is priority 3 added into PxTR-1 LISP configuration and priority 2 into PxTR-2 LISP configuration.

CP-3

```
Router lisp
instance-id 100
service ipv4
eid-table vrf 100_White
route-export site-registrations
distance site-registrations 250
!
router bgp 65000
address-family ipv4 vrf 100_White
aggregate-address 172.16.100.0 255.255.255.0 summary-only
redistribute lisp metric 10 route-map LISP100-TO-100_White-RIB
!
ip prefix-list LISP100-TO-100_White-RIB seq 10 permit 172.16.100.0/24 ge 32
route-map LISP100-TO-100_White-RIB permit 10
match ip address prefix-list LISP100-TO-100_White-RIB
```

PxTR-1

```
interface Loopback0
ip address 192.168.222.222 255.255.255.255
!
router lisp
locator-set RLOC-SET1
IPv4-interface Loopback0 priority 3 weight 1
!
instance-id 100
service ipv4
eid-table vrf 100_White
map-cache 0.0.0.0/0 map-request
route-import database bgp 65000 locator-set RLOC-SET1
map-cache site-registration
proxy-etr
proxy-itr 192.168.222.222
```

PxTR-2

```
interface Loopback0
ip address 192.168.2.2 255.255.255.255
!
router lisp
locator-set RLOC-SET1
IPv4-interface Loopback0 priority 2 weight 1
!
instance-id 100
service ipv4
eid-table vrf 100_White
map-cache 0.0.0.0/0 map-request
route-import database bgp 65000 locator-set RLOC-SET1
map-cache site-registration
proxy-etr
proxy-itr 192.168.2.2
```

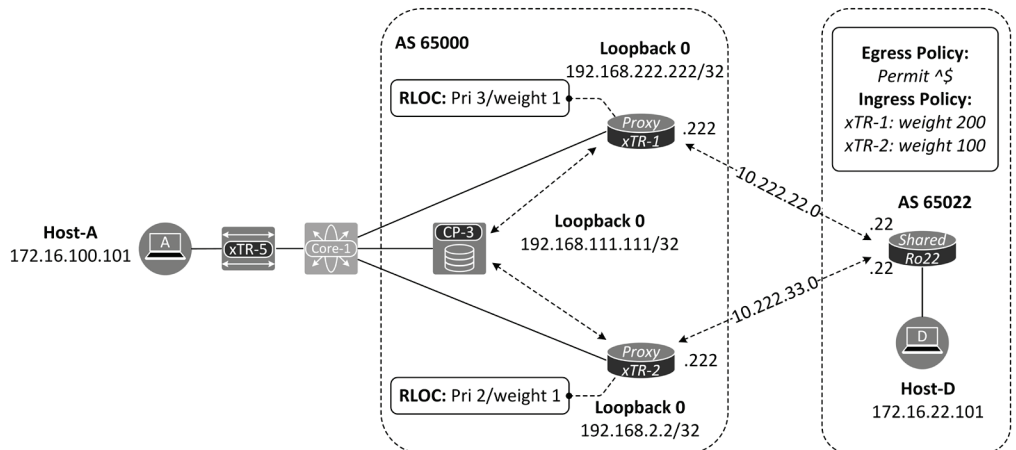


Figure 9-10: Basic configurations.

PxTR Redundancy: Priority Verification

Figure 9-11 shows the EID-to-RLOC table where mapping information about subnet 172.16.22.0/24 is advertised with RLOC Priority 2 by PxTR-2 and with RLOC Priority 3 by PxTR-1.

CP-3#sh lisp site name Network-Times instance-id 100

Site name: Network-Times

Allowed configured locators: any

Allowed EID-prefixes:

EID-prefix: 172.16.22.0/24 instance-id 100

<snipped>

ETR 192.168.2.2, last registered 00:20:46, proxy-reply, map-notify

TTL 1d00h, no merge, hash-function sha1, nonce 0xE3168E3F-0x8A5A8F62

state complete, no security-capability

xTR-ID 0xF092057C-0x652642F4-0xA73EB5D9-0x34502FCC

site-ID unspecified

sourced by reliable transport

Locator	Local	State	Pri/Wgt	Scope
192.168.2.2	yes	up	2/1	IPv4 none

ETR 192.168.222.222, last registered 00:21:38, proxy-reply, map-notify

TTL 1d00h, no merge, hash-function sha1, nonce 0xBE0755EF-0xA8D5C84C

state complete, security-capability

xTR-ID 0x062BA01A-0xE6B84DCE-0x16D21A4E-0x3AC1BEB7

site-ID unspecified

sourced by reliable transport

Locator	Local	State	Pri/Wgt	Scope
192.168.222.222	yes	up	3/1	IPv4 none

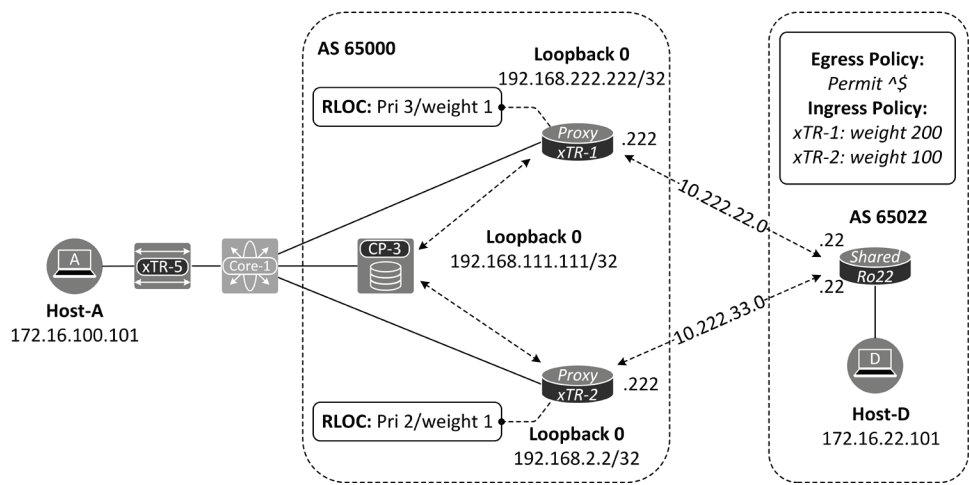


Figure 9-11: EID-to-RLOC Mapping Information in the Map-Server CP-3 Database.

Figure 9-12 shows that xTR-5 is introduced by Map-Resolver CP-3 to use PxTR-2 (192.168.2.2), due to its lower RLOC Priority value, as a destination IP address for traffic to network 172.16.22.0/24 in its outer tunnel header.

This can be verified also by tracing from Host-A to Host-D. The fourth hop 10.222.33.22 is the IP address of the link to PxTR-2 in the router Shared-Ro22.

```
xTR-5#sh lisp instance-id 100 ipv4 map-cache
LISP IPv4 Mapping Cache for EID-table vrf 100_White (IID 100), 3 entries

0.0.0.0/0, uptime: 00:05:28, expires: never, via static-send-map-request
Negative cache entry, action: send-map-request
172.16.22.0/24, uptime: 00:05:23, expires: 23:59:05, via map-reply, complete
Locator      Uptime    State     Pri/Wgt    Encap-IID
192.168.2.2  00:00:54  up        2/1        -
172.16.100.0/24, uptime: 00:05:28, expires: never, via dynamic-EID, send-map-request
Negative cache entry, action: send-map-request
```

```
HostA> trace 172.16.22.101
trace to 172.16.22.101, 8 hops max, press Ctrl+C to stop
 1  172.16.100.1   1.399 ms  0.931 ms  1.253 ms
 2  10.1.11.11    3.627 ms  3.713 ms  2.220 ms
 3  * * *
 4  10.222.33.22  5.617 ms  5.974 ms  4.576 ms
 5  *172.16.22.101 4.931 ms (ICMP type:3, code:3, Destination port unreachable)
```

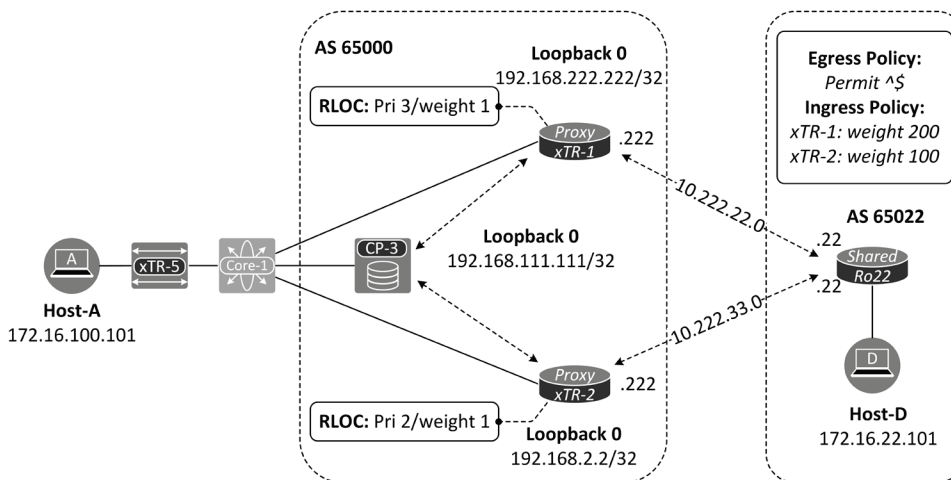


Figure 9-12: LISP Mapping-Cache in xTR-5.

Figure 9-13 illustrate the EID-to-RLOC Mapping database in CP-3 when the RLOC Priority is changed from 2 to 5 in PxTR-2.

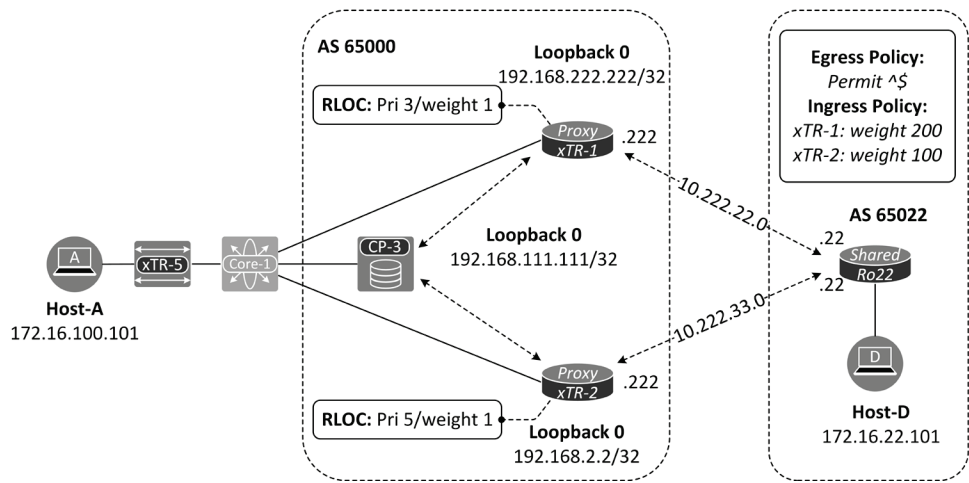
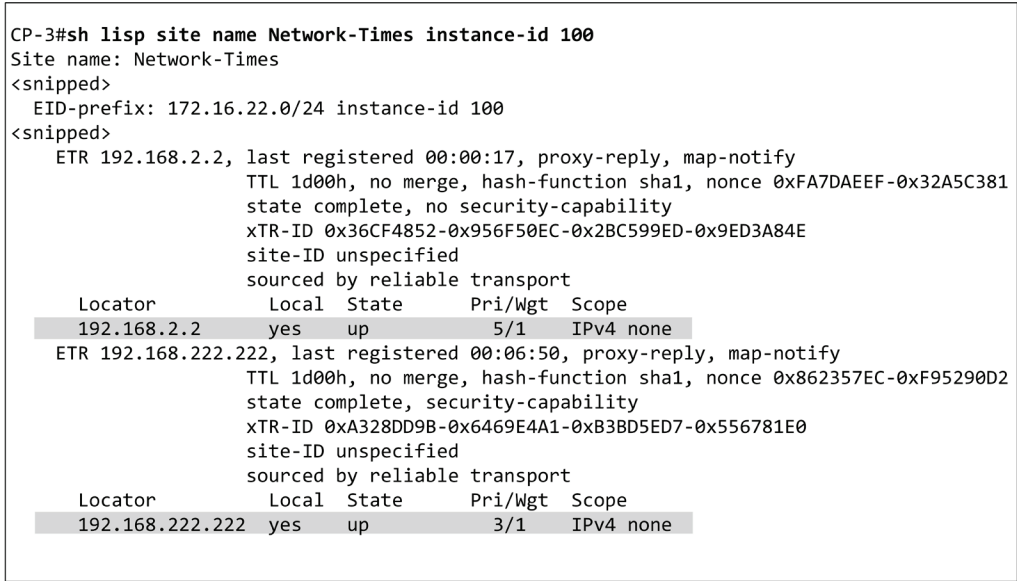


Figure 9-13: EID-to-RLOC Mapping Information in the Map-Server CP-3 Database.

Figure 9-14 shows that xTR-5 is introduced by Map-Resolver CP-3 to now use PxTR-1 (192.168.222.222) as a destination IP address for traffic to network 172.16.22.0/24 in its outer tunnel header.

This can be verified also by tracing from Host-A to Host-D. The fourth hop 10.222.33.22 is the IP address of the link to PxTR-1 in the router Shared-Ro22.

```
xTR-5#sh lisp instance-id 100 ipv4 map-cache 172.16.22.101
LISP IPv4 Mapping Cache for EID-table vrf 100_White (IID 100), 3 entries

172.16.22.0/24, uptime: 00:02:14, expires: 23:57:46, via map-reply, complete
Sources: map-reply
State: complete, last modified: 00:02:14, map-source: 192.168.222.222
Active, Packets out: 19(1940 bytes) (~ 00:01:17 ago)
Locator      Uptime      State      Pri/Wgt      Encap-IID
192.168.222.222 00:02:14 up          3/1          -
  Last up-down state change: 00:02:14, state change count: 1
  Last route reachability change: 00:02:14, state change count: 1
  Last priority / weight change: never/never
  RLOC-probing loc-status algorithm:
  Last RLOC-probe sent: 00:02:14 (rtt 4ms)
```

```
HostA> trace 172.16.22.101
trace to 172.16.22.101, 8 hops max, press Ctrl+C to stop
 1  172.16.100.1  11.620 ms  1.449 ms  1.279 ms
 2  10.1.11.11   3.129 ms  2.936 ms  2.870 ms
 3  * * *
 4  10.222.22.22 12.105 ms  4.772 ms  5.166 ms
 5  *172.16.22.101 4.747 ms (ICMP type:3, code:3, Destination port unreachable)
```

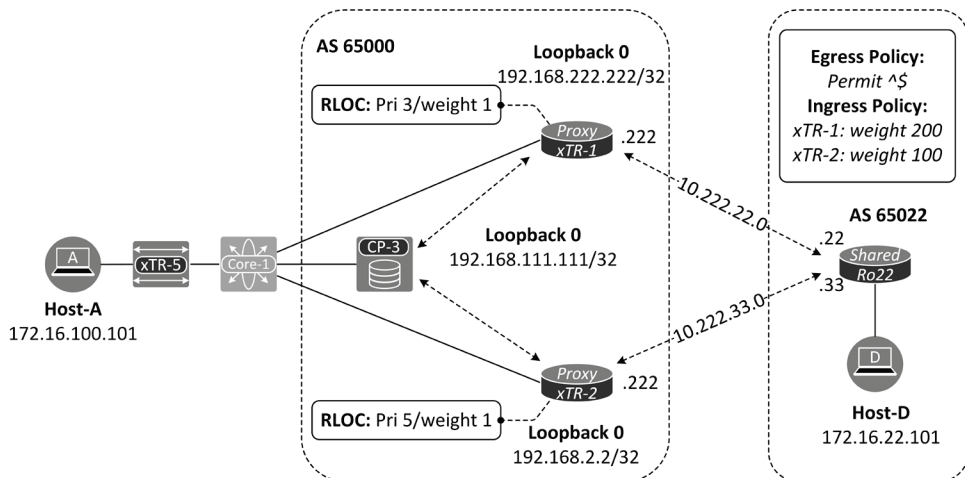


Figure 9-14: LISP Mapping-Cache in xTR-5.

PxTR Redundancy: PxTR Switch-over - Max. Priority

For the PxTR redundancy test, the RLOC Priority in PxTR-1 is changed from 3 to 255 (figure 9-15). The RLOC Priority value 255 means that PxTR-1 is not eligible as an RLOC for any of its advertised EIDs, which can be seen as an “admin-down” state in Map-Server CP-3 EID-to-RLOC database.

```
CP-3#sh lisp site name Network-Times instance-id 100
Site name: Network-Times
<snipped>
EID-prefix: 172.16.22.0/24 instance-id 100
<snipped>
ETR 192.168.2.2, last registered 00:01:55, proxy-reply, map-notify
TTL 1d00h, no merge, hash-function sha1, nonce 0x544A95BA-0x29CFB9C4
state complete, no security-capability
xTR-ID 0x9966E1AC-0x6C122F69-0x45E97DF5-0x71A695F3
site-ID unspecified
sourced by reliable transport
Locator      Local State  Pri/Wgt  Scope
192.168.2.2  yes    up       5/1     IPv4 none
ETR 192.168.222.222, last registered 00:08:34, proxy-reply, map-notify
TTL 1d00h, no merge, hash-function sha1, nonce 0x16F705EB-0xD72F4B5C
state complete, security-capability
xTR-ID 0xE8EDBCA7-0x7B1BE4ED-0x3B75C5E2-0x0ADCD85B
site-ID unspecified
sourced by reliable transport
Locator      Local State  Pri/Wgt  Scope
192.168.222.222  yes    admin-down 255/1   IPv4 none
```

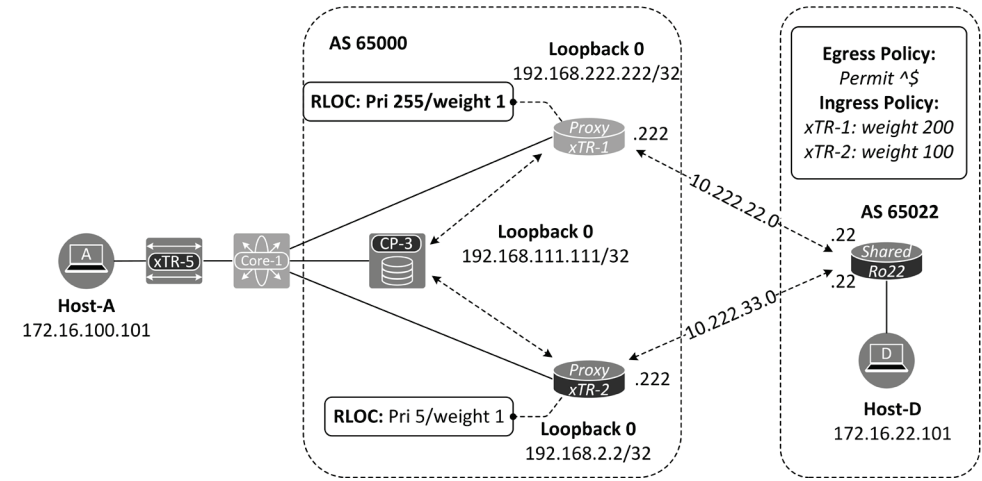


Figure 9-15: Taking Active PxTR-1 Out of Service from the CP-3 perspective.

CP-3 now introduces xTR-5 to switch from PxTR-1 to PxTR-2 due to lower RLOC Priority value (and it is also the only available PxTR at this moment).

```
xTR-5#sh lisp instance-id 100 ipv4 map-cache 172.16.22.101
LISP IPv4 Mapping Cache for EID-table vrf 100_White (IID 100), 3 entries

172.16.22.0/24, uptime: 00:08:04, expires: 23:56:17, via map-reply, complete
Sources: map-reply
State: complete, last modified: 00:03:43, map-source: 192.168.2.2
Active, Packets out: 19(1940 bytes) (~ 00:02:20 ago)
Locator      Uptime      State      Pri/Wgt      Encap-IID
192.168.2.2  00:03:43  up         5/1          -
  Last up-down state change:      00:03:43, state change count: 1
  Last route reachability change: 00:03:43, state change count: 1
  Last priority / weight change:  never/never
RLOC-probing loc-status algorithm:
  Last RLOC-probe sent:          00:03:42 (rtt 3ms)
```

```
HostA> trace 172.16.22.101
trace to 172.16.22.101, 8 hops max, press Ctrl+C to stop
 1  172.16.100.1   1.830 ms  1.775 ms  1.364 ms
 2  10.1.11.11    3.416 ms  2.348 ms  3.348 ms
 3  * * *
 4  10.222.33.22   5.976 ms  4.576 ms  4.883 ms
 5  *172.16.22.101 6.148 ms (ICMP type:3, code:3, Destination port unreachable)
```

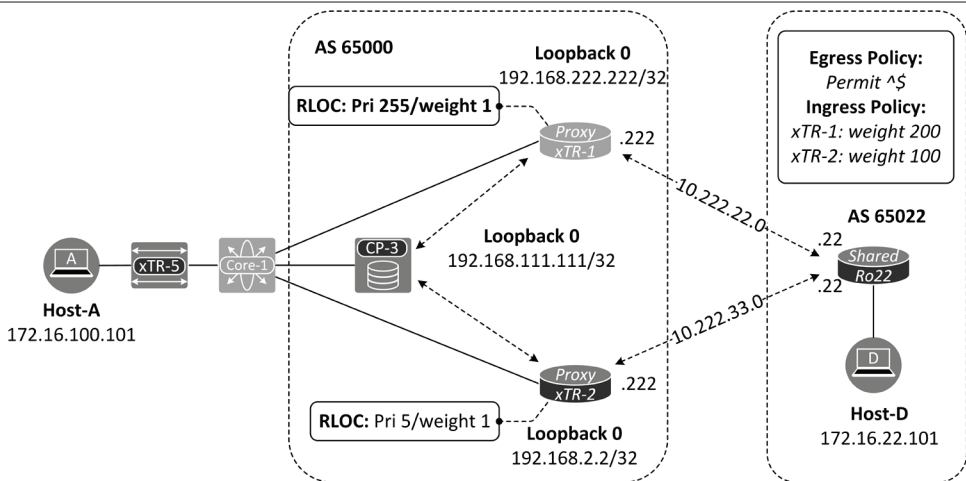


Figure 9-16: Taking Active PxTR-1 Out of Service from the xTR-5 perspective.

Figure 9-17 shows the EID-to-RLOC mapping database of CP-3 when the RLOC Priority is changed back to 3 in PxTR-1.

CP-3#sh lisp site name Network-Times instance-id 100

Site name: Network-Times

<snipped>

EID-prefix: 172.16.22.0/24 instance-id 100

<snipped>

ETR 192.168.222.222, last registered 00:00:10, proxy-reply, map-notify

TTL 1d00h, no merge, hash-function sha1, nonce 0x16F705EB-0xD72F4B5C

state complete, security-capability

xTR-ID 0xE8EDBCA7-0x7B1BE4ED-0x3B75C5E2-0x0ADCD85B

site-ID unspecified

sourced by reliable transport

Locator	Local	State	Pri/Wgt	Scope
192.168.222.222	yes	up	3/1	IPv4 none

ETR 192.168.2.2, last registered 00:08:31, proxy-reply, map-notify

TTL 1d00h, no merge, hash-function sha1, nonce 0x544A95BA-0x29CFB9C4

state complete, no security-capability

xTR-ID 0x9966E1AC-0x6C122F69-0x45E97DF5-0x71A695F3

site-ID unspecified

sourced by reliable transport

Locator	Local	State	Pri/Wgt	Scope
192.168.2.2	yes	up	5/1	IPv4 none

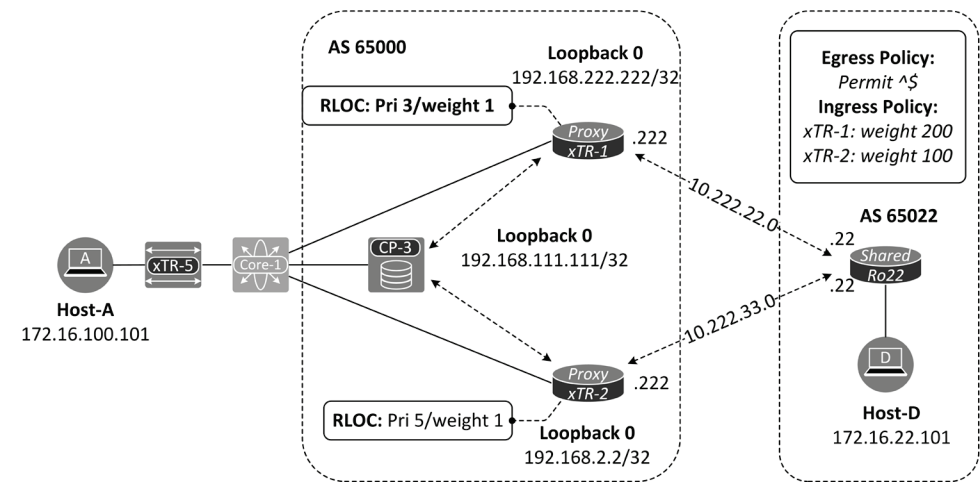


Figure 9-17: Changing the RLOC Priority back to 3 in PxTR-1.

Figure 9-18 verifies that the data path is switched back to PxTR-1.

```
xTR-5#sh lisp instance-id 100 ipv4 map-cache 172.16.22.101
LISP IPv4 Mapping Cache for EID-table vrf 100_White (IID 100), 3 entries

172.16.22.0/24, uptime: 00:00:04, expires: 23:59:55, via map-reply, complete
Sources: map-reply
State: complete, last modified: 00:00:04, map-source: 192.168.222.222
Active, Packets out: 0(0 bytes)
Locator      Uptime    State    Pri/Wgt    Encap-IID
192.168.222.222 00:00:04 up        3/1        -
  Last up-down state change: 00:00:04, state change count: 1
  Last route reachability change: 00:00:04, state change count: 1
  Last priority / weight change: never/never
  RLOC-probing loc-status algorithm:
    Last RLOC-probe sent: 00:00:04 (rtt 3ms)
```

```
HostA> trace 172.16.22.101
trace to 172.16.22.101, 8 hops max, press Ctrl+C to stop
 1 172.16.100.1 1.759 ms 1.211 ms 1.477 ms
 2 10.1.11.11 2.776 ms 2.856 ms 2.447 ms
 3 * * *
 4 10.222.22.22 4.480 ms 3.818 ms 4.500 ms
 5 *172.16.22.101 3.804 ms (ICMP type:3, code:3, Destination port unreachable)
```

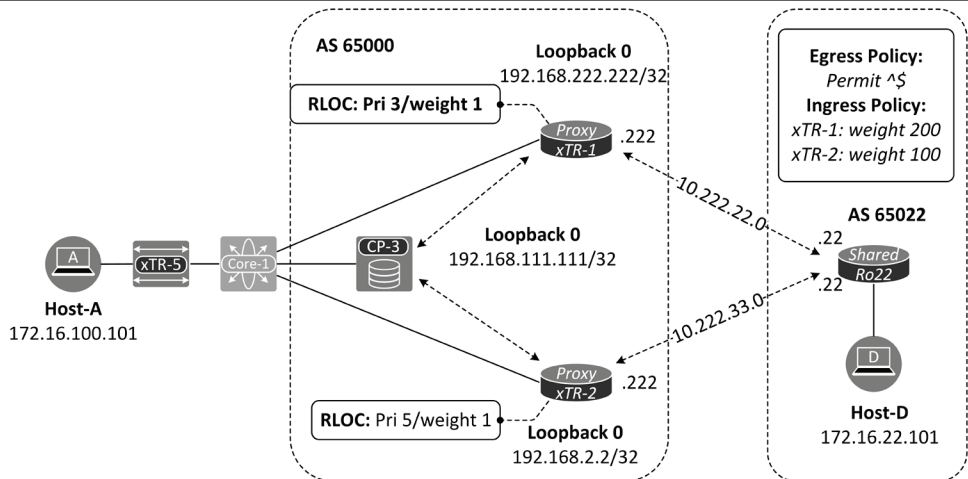


Figure 9-18: Data Path verification.

Chapter 10: Inter-Site and Internet connections

This chapter summarizes the LISP solutions introduced in this book by using the network shown in figure 10-1. The example network consists of two Campus Fabric sites where LISP is used as Control-Plane protocol and VXLAN as Data-Plane encapsulation. Inter-Site communication is done only over the Transit Service Provider network using a dedicated circuit. Both sites have dedicated Internet connection. The left-hand site Networktimes-1 is not using the Internet connection of the right-hand site Networktimes-2 as back-up Internet connection and vice versa. Host-A in site Networktimes-1 is attached to network 172.16.100.0/24 and Host-B in site Networktimes-2 is attached to network 172.16.101.0/24. The Instance-Id/VRF/L3SVI for both networks is 100/100_White/BD100. Complete device configuration can be found in Appendix-3 at the end of the book.

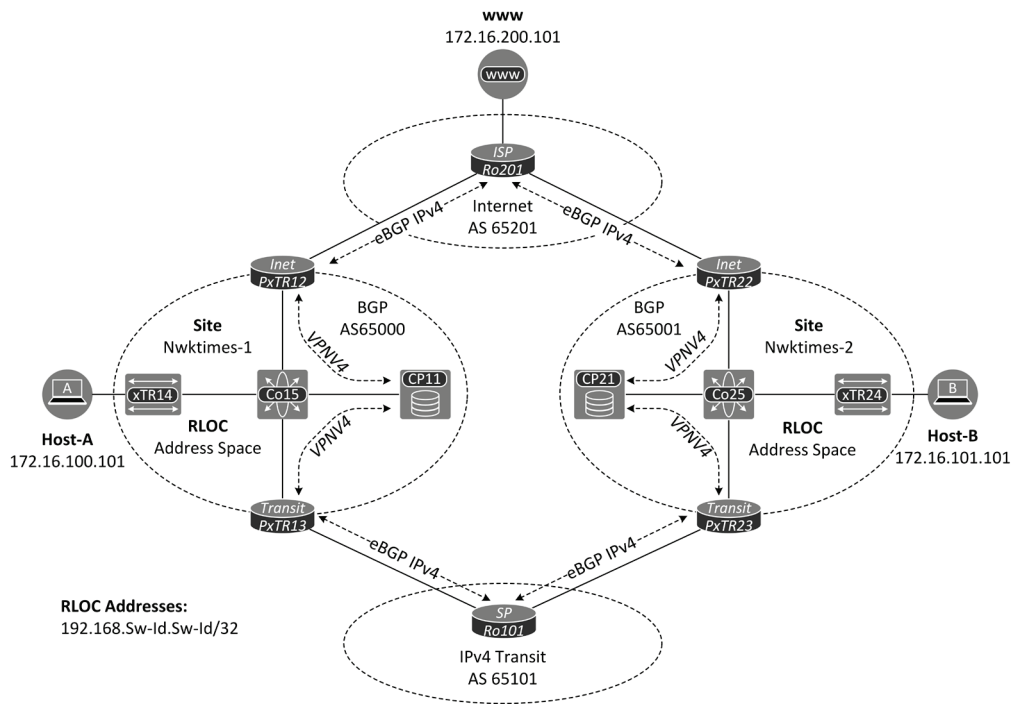


Figure 10-1: Example Network Topology.

LISP Registration Process by iTR

This section recaps the LISP Control-Plane processes of how the EID information about Host-A is registered to CP11 by xTR14 and how the information eventual ends up to LISP Map-Cache on xTR24.

Figure 10-2 on the next page illustrates the process where Host-A joins the network and the ingress tunnel router xTR14 learns its IP address 172.16.100.101. The IP address belongs to subnet 172.16.100.0/24 that in turn belongs to the dynamic-eid 172_16_100_0-100_White of LISP Instance 100. The command **`lisp mobility 172_16_100_0-100_White`** under the interface BD100 configuration starts the LISP mapping process when an IP address belonging to subnet 172.16.100.0 is learned by xTR. The mapping information sent by xTR is stored into the EID-to-RLOC Mapping Database of the LISP Map-Server (CP11). The mapping information is allowed to be installed into CP11 database with command **`eid-record instance-id 100 172.16.100.0/24 accept-more-specific`** under the site-specific configuration. The RLOC address attached to the subnet is the RLOC address used by xTR14 (192.168.14.14).

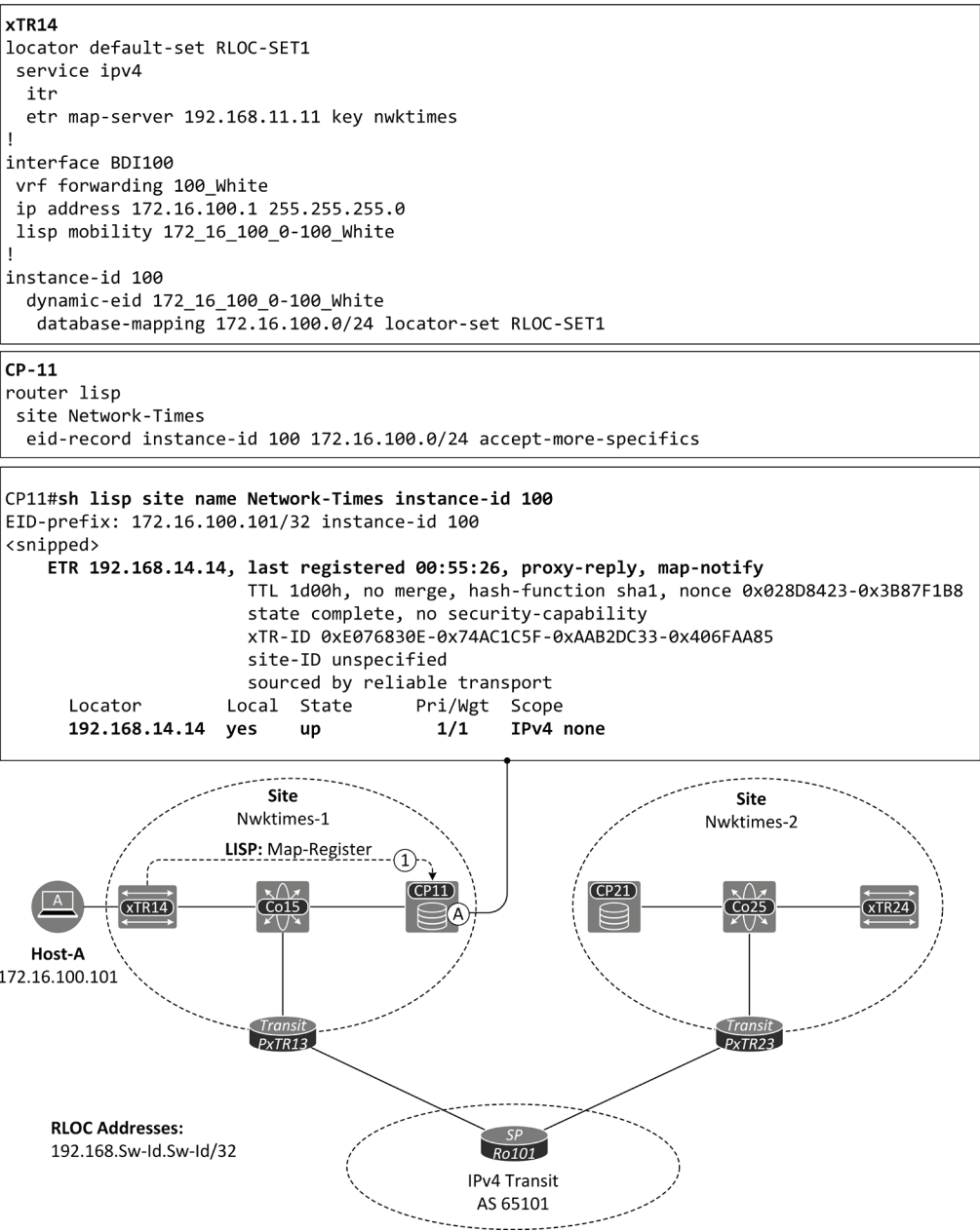


Figure 10-2: EID-to-RLOC Mapping Process.

LISP-to-BGP redistribution by Map-Server

Figure 10-3 shows the configuration needed in order to Map-Server CP11 advertise its LISP EID information to Transit-PxTR14. The Map-Server CP11 redistributes routes, permitted in route-map, from the LISP process to BGP and advertises the route as an aggregate route to its iBGP VPNv4 peer 192.168.13.13. The redistribution process is done in a controlled manner because CP11 also receives EID information from Transit-PxTR, which imports BGP routes to LISP. If redistribution is done without any control, CP11 also starts advertising these routes causing a routing loop.

Figure 10-3 also shows that Transit-PxTR14 has received the BGP-Update message from CP11 about subnet 172.16.100.0/24 and installed it into Adj-RIB-In from where it is imported into Loc-RIB of VRF 100_White based on the Route-Target 1:100 carried as BGP Extended Community Path Attribute. The next-hop-route IP address is 192.168.11.11 that is the Loopback 0 of CP11.

```
CP11
router bgp 65000
neighbor 192.168.13.13 remote-as 65000
neighbor 192.168.13.13 update-source Loopback0
!
address-family vpnv4
neighbor 192.168.13.13 activate
neighbor 192.168.13.13 send-community extended
exit-address-family
!
address-family ipv4 vrf 100_White
aggregate-address 172.16.100.0 255.255.255.0 summary-only
redistribute lisp metric 10 route-map LISP100-TO-100_White-RIB
exit-address-family
!
ip prefix-list LISP100-TO-100_White-RIB seq 10 permit 172.16.100.0/24 ge 32

Transit-PxTR13#sh ip bgp vpnv4 vrf 100_White
BGP table version is 6, local router ID is 192.168.13.13
<snipped>
      Network          Next Hop          Metric LocPrf Weight Path
Route Distinguisher: 1:100 (default for vrf 100_White)
 *>i 172.16.100.0/24 192.168.11.11          0    100      0 i
<snipped>
```

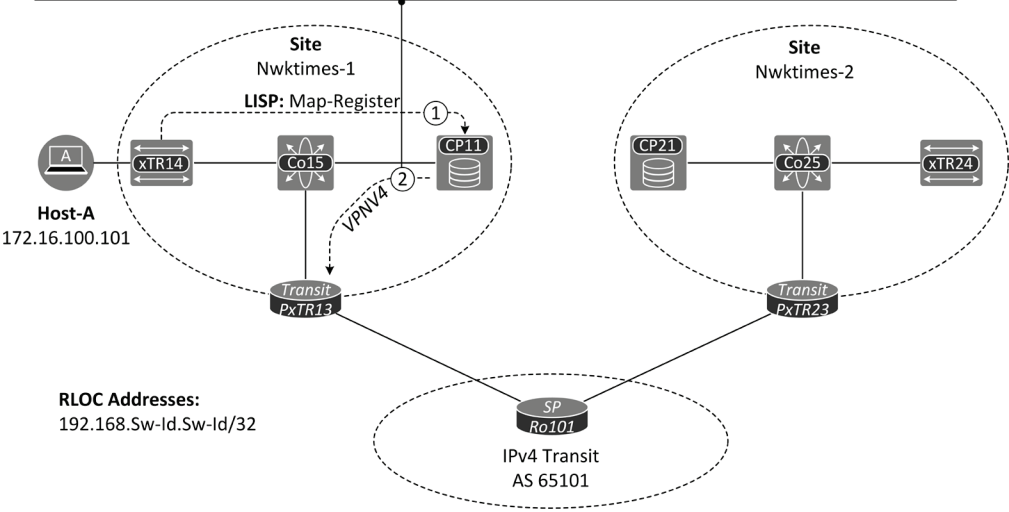


Figure 10-3: LISP to BGP redistribution by CP11.

BGP Update from PxTR to the non-LISP router

Figure 10-4 shows the Transit-PxTR14 BGP configuration. The peering with Transit-SP101 is done by using VRF specific eBGP afi IPv4 peering. The figure also shows the partial BGP Loc-RIB information of the router Transit-SP101.

```

Transit-xTR
router bgp 65000
no bgp log-neighbor-changes
neighbor 10.13.101.101 remote-as 65101
neighbor 192.168.11.11 remote-as 65000
neighbor 192.168.11.11 update-source Loopback0
!
address-family ipv4
neighbor 10.13.101.101 activate
neighbor 192.168.11.11 activate
exit-address-family
!
address-family vpnv4
neighbor 192.168.11.11 activate
neighbor 192.168.11.11 send-community extended
exit-address-family
!
address-family ipv4 vrf 100_White
neighbor 10.13.101.101 remote-as 65101
neighbor 10.13.101.101 activate
exit-address-family

```

```

Transit-SP101#sh ip bgp 172.16.100.0
BGP routing table entry for 172.16.100.0/24, version 2
Paths: (1 available, best #1, table default)
  Advertised to update-groups:
    1
  Refresh Epoch 1
  65000, (aggregated by 65000 192.168.11.11)
    10.13.101.13 from 10.13.101.13 (192.168.13.13)
      Origin IGP, localpref 100, valid, external, atomic-aggregate, best
      rx pathid: 0, tx pathid: 0x0

```

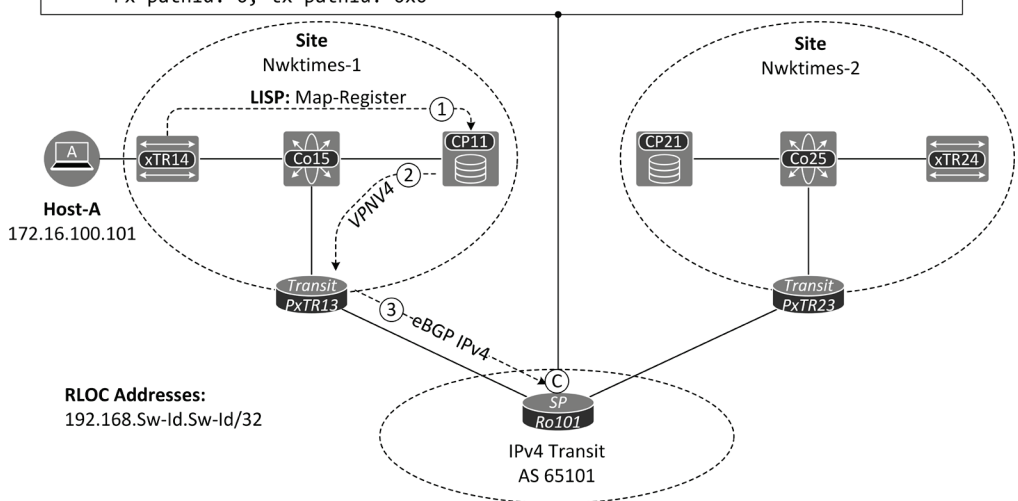


Figure 10-4: BGP-Update from PxTR14 to a non-LISP BGP peer.

BGP-to-LISP redistribution by Transit-PxTR23

Figure 10-5 shows that Transit-PxTR23 has received the BGP_Update message from Transit-SP101 and installed the routing information into BGP Loc-RIB. Transit-PxTR24 imports the routing information from the BGP process to the LISP process. The command **route-import database bgp 65001 locator-set RLOC-SET1** under the IPv4 service of Instance 100 defines that the routing information about the subnet 172.16.100.0/24 is registered to LISP Map-Server using RLOC address specified under the **locator-set RLOC-SET1**. The Map-Server where the Map-Register message will be sent is defined in section **IPv4 service** under the **locator default-set RLOC-SET1** using the command **etr map-server 192.168.21.21 key nwktimes**.

```

Transit-PxTR23
locator-set RLOC-SET1
  IPv4-interface Loopback0 priority 3 weight 1
!
locator default-set RLOC-SET1
service ipv4
  etr map-server 192.168.21.21 key nwktimes
!
router bgp 65001
  neighbor 10.23.101.101 remote-as 65101
!
address-family ipv4 vrf 100_White
  neighbor 10.23.101.101 remote-as 65101
  neighbor 10.23.101.101 activate
  exit-address-family
!
instance-id 100
  service ipv4
    eid-table vrf 100_White
    map-cache 0.0.0.0/0 map-request
    route-import database bgp 65001 locator-set RLOC-SET1

```

```

Transit-PxTR23#sh ip bgp vpnv4 vrf 100_White 172.16.100.0
BGP routing table entry for 1:100:172.16.100.0/24, version 2
Paths: (1 available, best #1, table 100_White)
  Advertised to update-groups:
    1
  Refresh Epoch 2
  65101 65000, (aggregated by 65000 192.168.11.11)
    10.23.101.101 (via vrf 100_White) from 10.23.101.101 (10.23.101.101)
      Origin IGP, localpref 100, valid, external, atomic-aggregate, best
      Extended Community: RT:1:100
      mpls labels in/out 16/nolabel
      rx pathid: 0, tx pathid: 0x0

```

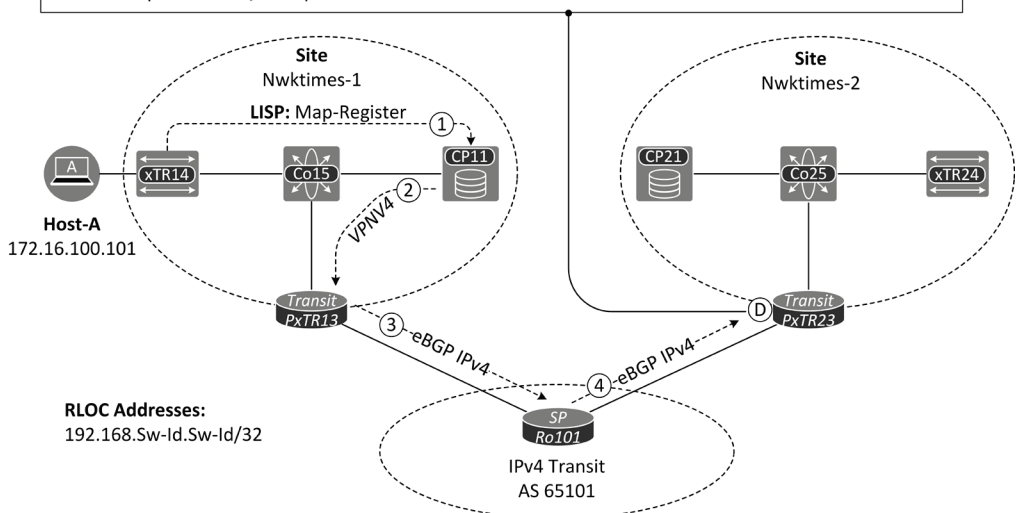


Figure 10-5: The BGP Loc-RIB of Transit-PxTR.

LISP Map-Register from Transit-PxTR23 to Map-Server CP21.

CP21 receives the LIPS Map-Register message from Transit-PxTR23 and due to the command **eid-record instance-id 100 172.16.100.0/24 accept-more-specifics** the information is installed into the EID-to-RLOC mapping database. The Egress Tunnel Router RLOC is now Transit-PxTR24 with IP address 192.168.23.23.

CP21

site Network-Times-2
authentication-key nwktimes
eid-record instance-id 100 172.16.100.0/24 accept-more-specifics

CP21#sh lisp site name Network-Times-2 instance-id 100

EID-prefix: 172.16.100.0/24 instance-id 100

<snipped>

ETR 192.168.23.23, last registered 00:11:24, proxy-reply, map-notify

TTL 1d00h, no merge, hash-function sha1, nonce 0xD981E75E-0x9AE9B500

state complete, security-capability

xTR-ID 0x3DEE3B7E-0x9C5D1E48-0xC2FDA9CE-0x3B41440F

site-ID unspecified

sourced by reliable transport

Locator	Local	State	Pri/Wgt	Scope
192.168.23.23	yes	up	3/1	IPv4 none

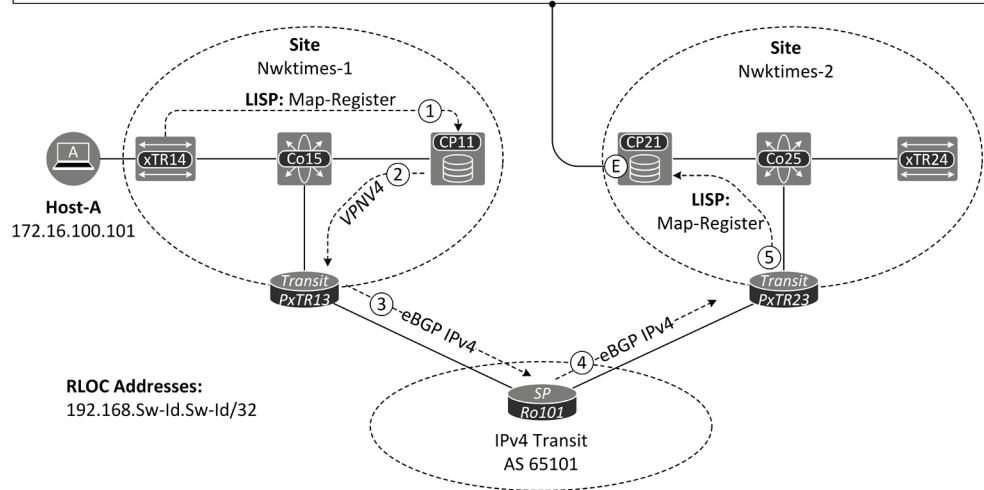


Figure 10-6: EID-to-RLOC Mapping Database on CP21.

Now the reachability information about Host-A is propagated from CP11 in site Networktimes-1 to CP21 in site Networktimes-2. The next section explains the Data-Plane operation.

Map-Request and Map-Reply process

Host-B starts pinging Host-A IP. Because the destination IP address is in a different subnet than Host-B, it sends the packet to its gateway xTR24. The Edge xTR24 receives the IP packet and it first makes RIB lookup. There is no routing information about network 172.16.100.0/24 in the RIB so as a next step xTR checks the LISP Mapping-Cache. Because the lookup result is miss, xTR24 sends a Map-Request to Map-Resolver CP21. CP21 has received EID-to-RLOC Mapping from PxTR23 so it replies with LISP Map-Reply message to xTR24. Figure 10-7 shows the LISP Mapping-Cache of xTR24 after receiving the mapping information from CP21. RLOC for the network 172.16.100.0/24 is Transit-PxTR23.

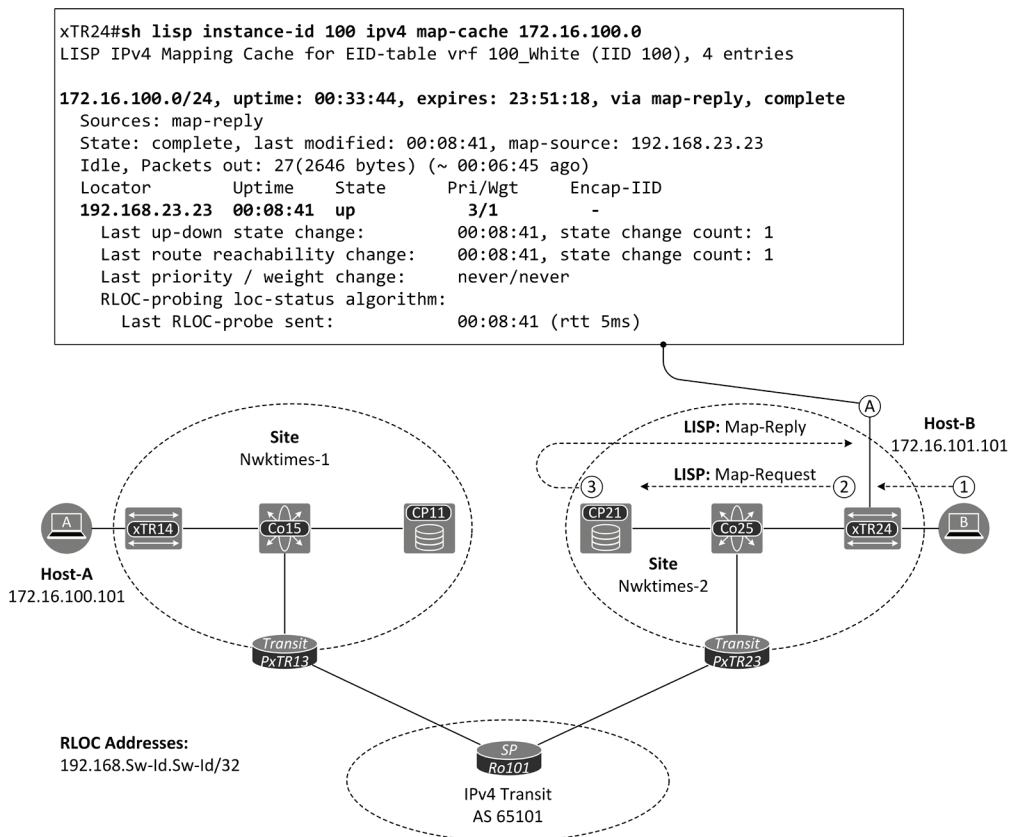


Figure 10-7: The Map-Request and Map-Reply process.

Non-LIPS router routing process

The router Transit-SP101 receives the ICMP-Request from Transit-PxTR23 and forwards it based on the global RIB. It has received the routing information previously from its eBGP peer Transit-PxTR13.

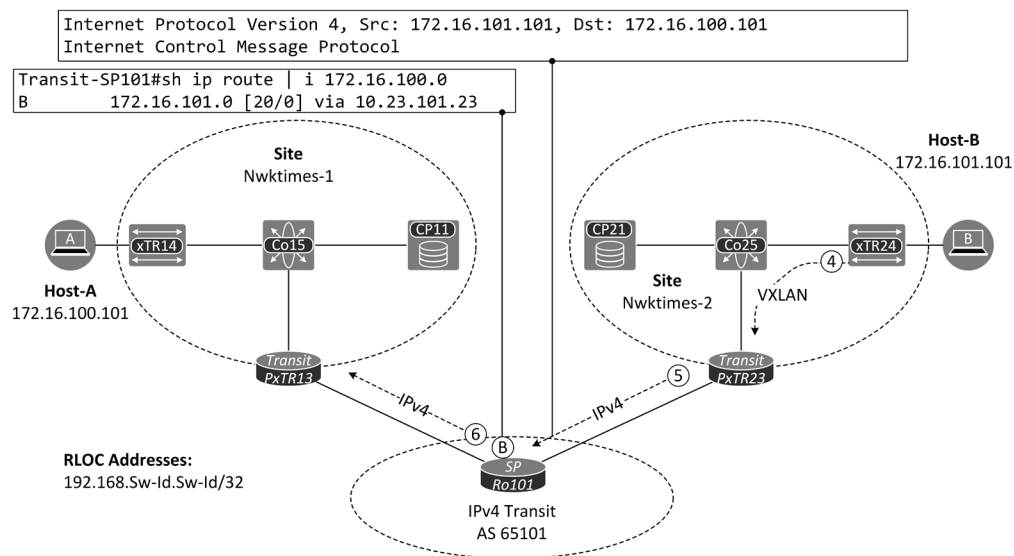


Figure 10-9: ICMP-Request forwarding.

Map-Request caused by incoming IP packet from non-LISP site

Transit-PxTR receives the ICMP-Request from its eBGP peer Transit-SP101. The IP packet ingress interface belongs to VRF 100_White. Proxy-PxTR13 already has an EID-to-RLOC mapping information about EID 172.17.100.101 with the RLOC address 192.168.14.14 (xTR14). Transit-PxTR13 encapsulates the received ICMP-Request with VXLAN tunnel headers. It uses the RLOC IP address 192.168.14.14 as a destination IP in the outer IP header. In addition, it sets the Instance-Id 100 to the VNI field into the VXLAN header. The core router Co15 makes the routing decision based on the IP address in the outer IP header. When xTR14 receives the packet, it notices that this VXLAN encapsulated packet belongs to it. It then removes the tunnel headers and forwards the original ICMP-Request to Host-A.

```
Transit-PxTR13#sh lisp instance-id 100 ipv4 map-cache
LISP IPv4 Mapping Cache for EID-table vrf 100_White (IID 100), 2 entries

0.0.0.0/0, uptime: 02:02:22, expires: never, via static-send-map-request
Negative cache entry, action: send-map-request
172.16.100.101/32, uptime: 00:53:54, expires: 23:55:53, via map-reply, self, complete
Locator      Uptime    State      Pri/Wgt    Encap-IID
192.168.14.14 00:53:54 up          1/1        -
```

```
Internet Protocol Version 4, Src: 192.168.13.13, Dst: 192.168.14.14
User Datagram Protocol, Src Port: 1024, Dst Port: 4789
Virtual eXtensible Local Area Network
  Flags: 0x8840, GBP Extension, Don't Learn, VXLAN Network ID (VNI)
  Group Policy ID: 0
  VXLAN Network Identifier (VNI): 100
  Reserved: 0
Ethernet II, Src: 50:00:00:0a:00:00, Dst: ba:25:cd:f4:ad:38
Internet Protocol Version 4, Src: 172.16.101.101, Dst: 172.16.100.101
Internet Control Message Protocol
```

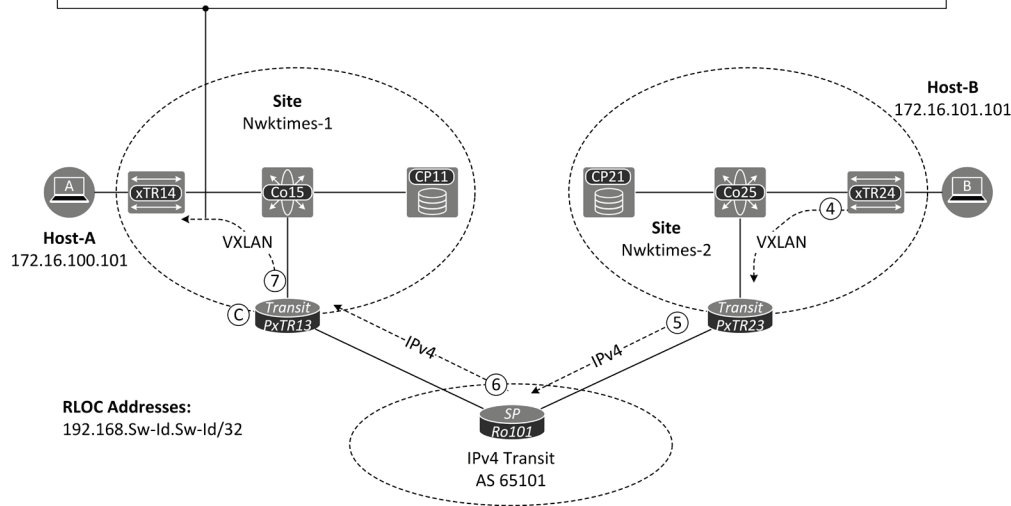


Figure 10-10: ICMP-Request forwarding.

Verification

Figure 10-11 verifies that the ping works and that the ICMP-Request packets are routed over the transit network

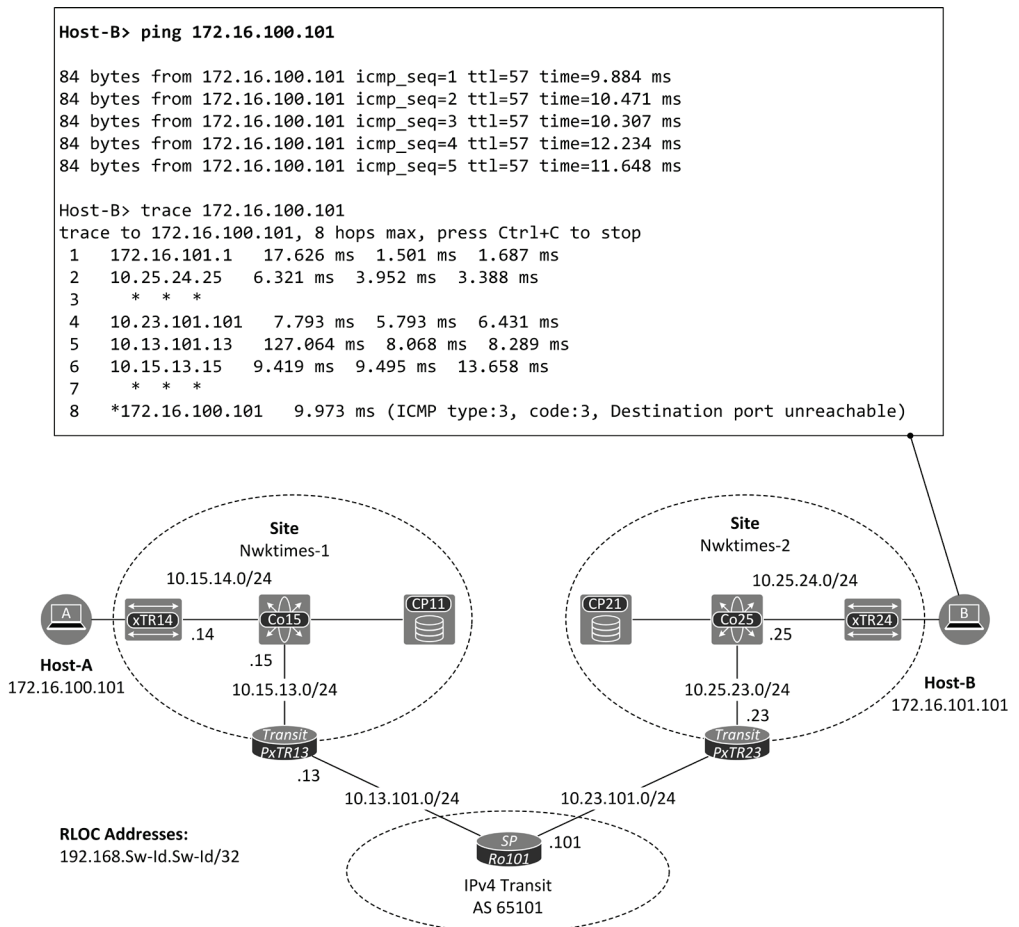


Figure 10-11: Ping and Trace from Host-B to Host-A.

Internet Connection: EID to non-LISP Router

Figure 10-12 illustrates the process of how the EID reachability information is propagated from the LISP site to untrusted non-LISP sites. The process follows the same principals that what was explained in previous sections. (1) Edge xTRs register their connected hosts to Map-Servers. (2) Map-Servers, in turn, send a BGP-Update message as aggregated NLRI to Inet-PxTR routers. (3) Inet-PxTR sends a BGP-Update message to the non-LISP router ISP-Ro201. There is a BGP policy configured on the ISP-Router, which denies routes learned from site Networktimes-1 to be advertised to site Networktimes-2 and the other way around. This prevents site-to-site communication over the Internet connection. Note, site-to-site communication can be established over the Internet using reliable and secure tunneling solutions. However, in this chapter, the Internet connection is used only for Internet access.

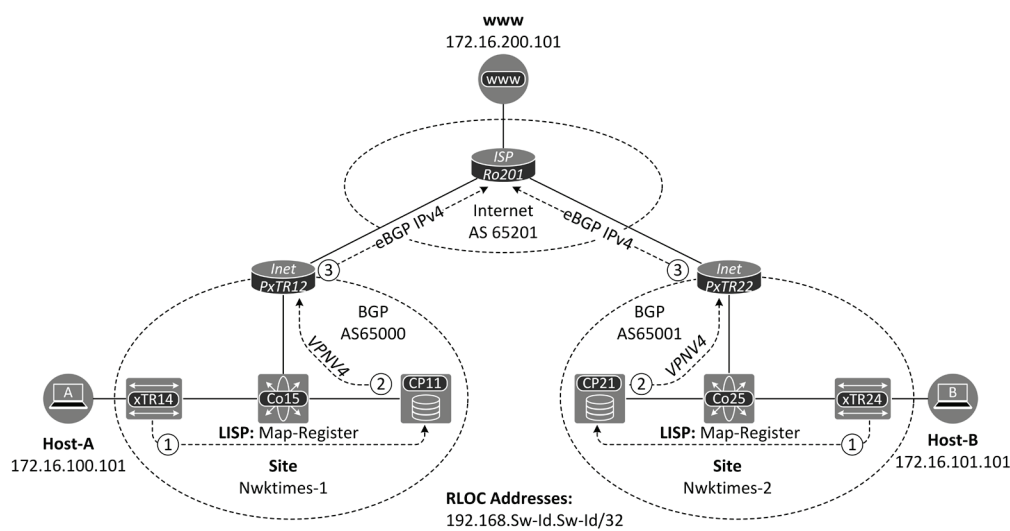


Figure 10-12: EID propagation from the LISP site to the untrusted non-LISP site.

Internet Connection: non-LISP Router network to LISP Site

The policy on Map-Servers CP11 and CP21 only allow the internal networks to be installed in the EID-to-RLOC Mapping database. So the mapping information received from Inet-PxTRs is rejected by both Map-Servers. Figure 10-13 shows that when xTR24 sends the Map-Request about EID-to-RLOC mapping information about the www-server 172.16.200.101, it gets a Negative Map-Reply message from the Map-Server CP21. Instead of describing the target RLOC address, CP11 instructs the requesting xTR24 to use its configured Proxy-ETR router, which is in the case of xTR24 is Inet-PxTR 192.168.22.22. Note that the proxy-etr role has to be assigned to Inet-PxTR routers.

```
Inet-PxTR22
Router lisp
instance-id 100
service ipv4
eid-table vrf 100_White
proxy-etr
proxy-itr 192.168.22.22

xTR24
router lisp
locator default-set RLOC-SET1
service ipv4
encapsulation vxlan
use-petr 192.168.22.22

xTR24#sh lisp instance-id 100 ipv4 map-cache 172.16.200.0
LISP IPv4 Mapping Cache for EID-table vrf 100_White (IID 100), 4 entries

172.16.128.0/17, uptime: 00:32:17, expires: 00:12:27, via map-reply, forward-
native
Sources: map-reply
State: forward-native, last modified: 00:32:17, map-source: 192.168.21.21
Idle, Packets out: 44(4312 bytes) (~ 00:08:40 ago)
Encapsulating to proxy ETR
```

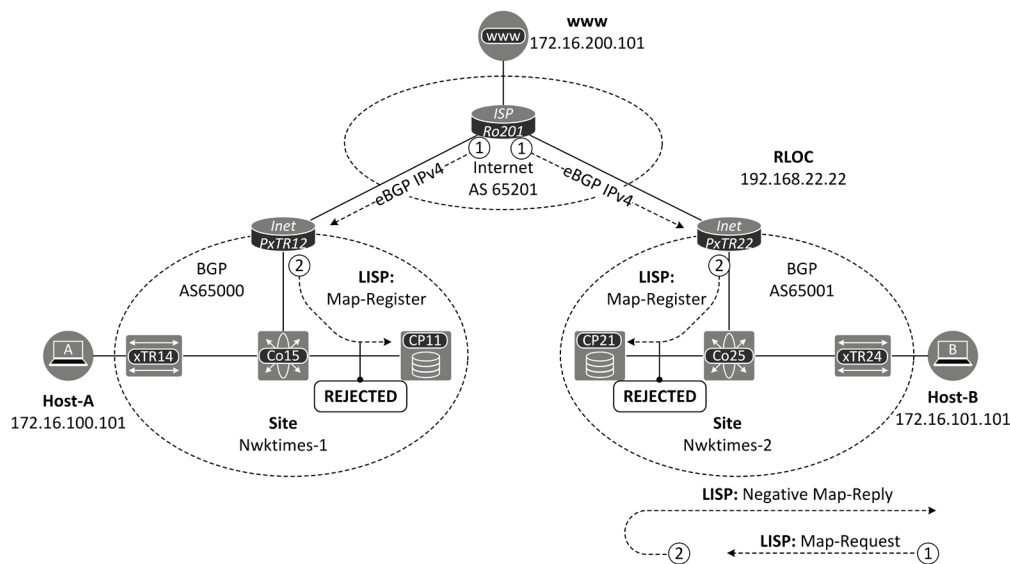


Figure 10-13: Negative Map-Reply.

Internet Connection verification

Figure 10-14 verifies that the IP connectivity from Host-B (172.16.101.101) to Host www (172.16.200.101) is routed correctly.

```
ISP-201>sh ip bgp
BGP table version is 10, local router ID is 172.16.200.1
<snipped>
      Network          Next Hop          Metric LocPrf Weight Path
*>  172.16.100.0/24    10.12.201.12          0 65000 i
*>  172.16.101.0/24    10.22.201.22          0 65001 i

Host-B> ping 172.16.200.101

84 bytes from 172.16.200.101 icmp_seq=1 ttl=60 time=6.714 ms
84 bytes from 172.16.200.101 icmp_seq=2 ttl=60 time=5.994 ms
84 bytes from 172.16.200.101 icmp_seq=3 ttl=60 time=6.765 ms
84 bytes from 172.16.200.101 icmp_seq=4 ttl=60 time=5.429 ms
84 bytes from 172.16.200.101 icmp_seq=5 ttl=60 time=6.144 ms

Host-B> trace 172.16.200.101
trace to 172.16.200.101, 8 hops max, press Ctrl+C to stop
 1  172.16.101.1      2.568 ms  2.248 ms  2.180 ms
 2  10.25.24.25       4.103 ms  3.720 ms  3.717 ms
 3      * * *
 4  10.22.201.201     6.204 ms  6.756 ms  6.205 ms
 5  *172.16.200.101   6.217 ms (ICMP type:3, code:3, Destination port unreachable)
```

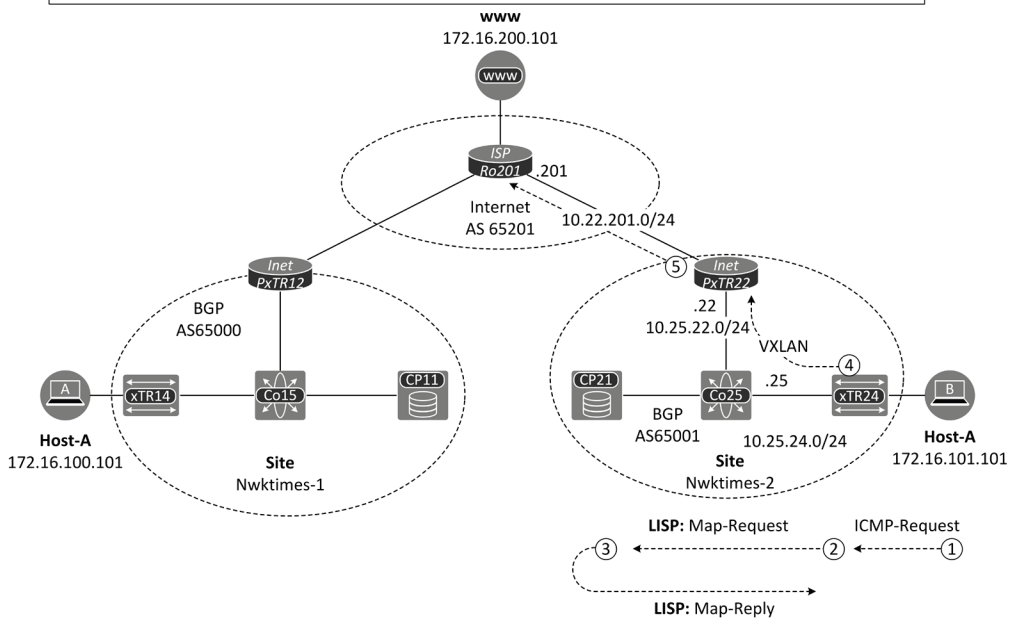


Figure 10-14: Verification.

References

- [RFC 6830] D. Farinacci et al., "The Locator/ID Separation Protocol (LISP)", RFC 6830, January 2013.
- [RFC 6833] V. Fuller and D. Farinacci., "Locator/ID Separation Protocol (LISP) Map-Server Interface", RFC 6833, January 2013.
- [VXLAN-GPE] F. Maino et al., "Generic Protocol Extension for VXLAN", draft-ietf-nvo3-vxlan-gpe-09, December 2019.
- [LISP Control Plane] D. Farinacci et al., "The Locator/ID Separation Protocol (LISP) Control Plane", draft-ietf-lisp-rtc6833bis-25, June 16, 2019.
- [VXLAN-GPE] J. Lemon et al., "Group Policy Encoding with VXLAN GPE", draft-ietf-vxlan-gpe-gbp-02, March 2, 2019.
- [LISP Rel-Tran] C. Cassar et al., "LISP Reliable Transport", draft-kouvelas-lisp-reliable-transport-02, March 26, 2015.

LISP Network Deployment and Troubleshooting: The Complete Guide to LISP Implementation on IOS-XE, IOS-XR, and NX-OS
ISBN-10: 1-58714-506-5 – Tarique Shakil, Vinit Jain, and Yves Louis

https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst9500/software/release/16-6/configuration_guide/cf/b_166_cf_9500_cg/b_166_cf_9500_cg_chapter_00.pdf

Chapter 11: The Underlay Network-SPT Algorithm

Introduction

The Campus Fabric Underlay Network is not aware of customer subnets, it forwards VXLAN encapsulated packets between LISP capable devices based on IP addresses on the outer IP header without checking the inner headers or payload.

The common routing protocol choice for the Campus Fabric Underlay Network is either IS-IS or OSPF, which both are Link-State Protocols. The basic operation of these two protocols is the same, each OSPF/IS-IS speaker advertises its link states information to adjacent OSPF/IS-IS peers. In addition, they store the information received from the peer into the Link-State Database and then floods the information to their own peer(s). This way each OSPF/IS-IS speaker has a common view of network topology. The routes that are installed into RIB are calculated independently by each device using Dijkstra/Shortest Path First (SPF) algorithm.

This chapter explains the operation of the Dijkstra/Shortest Path First (SPF) algorithm that is used by Link State Protocols for calculating the Shortest-Path Tree using OSPF as an example protocol. This chapter also discusses how the Shortest-Path Tree calculation process can be optimized by disabling the advertisement of unnecessary LSU information. In addition, this chapter illustrates how devices exchange the Link State information and based on that how they calculate the Shortest-Path Tree.

Note! Chapters 11 and 12 are also included, with BGP EVPN VXLAN Fabric perspective, in my book “Virtual Extensible LAN – VXLAN: A Practical Guide to Understand the Operation of VXLAN”

OSPF

OSPF speakers exchange Link State Update (LSU) packets with adjacent routers. LSU packets carry the Link State Advertisement (LSA) where the router describes its links. After successful validation of the received LSU, the router stores the Link descriptions into OSPF Link-State Database (LSDB) and floods it to adjacent routers. This way each router has a common topology view of an OSPF Area. After the LSDB synchronization process, routers run the Dijkstra algorithm to find the shortest/best path to each destination. The terms Dijkstra/SPF are used interchangeably in this chapter.

Figure 11-1 introduces the example topology used in this chapter. Switches from 101-105 represent LISP Capable devices (Edge/xTR and Control Plane nodes), which each have an Interface Loopback 50 (192.168.50.x) used as an RLOC IP address. Switches 11 and 12 represent Inter Mediate Node (IMN).

All Inter-Switch links and Loopback addresses belong to OSPF Area 0 in the example network. OSPF metric values shown in the figure are set statically by configuring an OSPF cost per interface. The IP addressing scheme of physical interfaces is based on switch numbering. As an example, the IP subnet 10.101.11.0/24 is assigned to the Inter-Switch link between switches 101 and 11. Obviously and the IP address of switch 101 is 10.101.11.101/24. This is not the ideal solution and in real life, using mask /30, /31 or Unnumbered interface is a much elegant solution. OSPF Router-Ids are configured statically (192.168.0.sw-id). Note that there are no Loopback Interfaces for RID in this example network. This is not the recommended solution but it reduces the LSAs in LSDB and makes it easier to explain the Dijkstra algorithm. In addition, there is a hostname-to-IP address mapping information and OSPF name-lookup enabled in each switch. This way the advertising OSPF routers are identified by its name instead of its IP address.

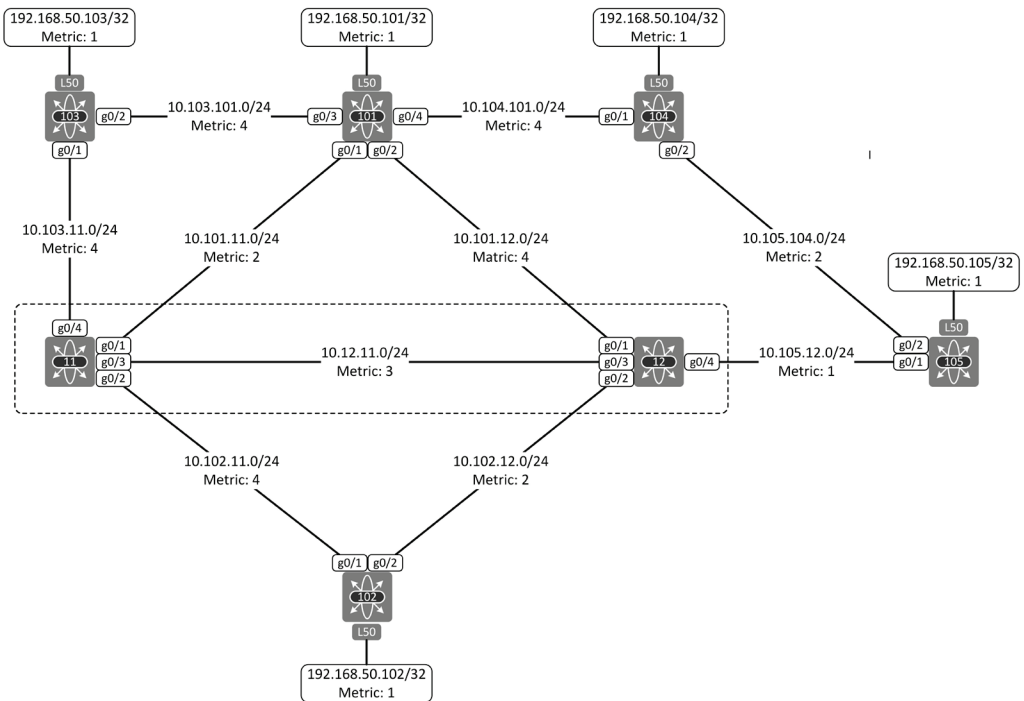


Figure 11-1: Underlay Network Topology and Interface Addressing.

Link-State Database (LSDB) optimization

The default OSPF network type in the Ethernet link is Broadcast. This means that routers connected to link select the Designated Router (DR) and Backup-designated Router (BDR) among themselves. Non-Designated routers send Router LSAs only to DR, which in turn floods them to other routers in the segment as a Network LSA (Type-2).

Figure 11-2 shows the LSA generation from the Switch 102 perspective, which is the DR for segments connected to switches 11 and 12. Switch 102 originates the LSU packet that carries both Router LSAs and Network LSAs and sends it to both adjacent switches.

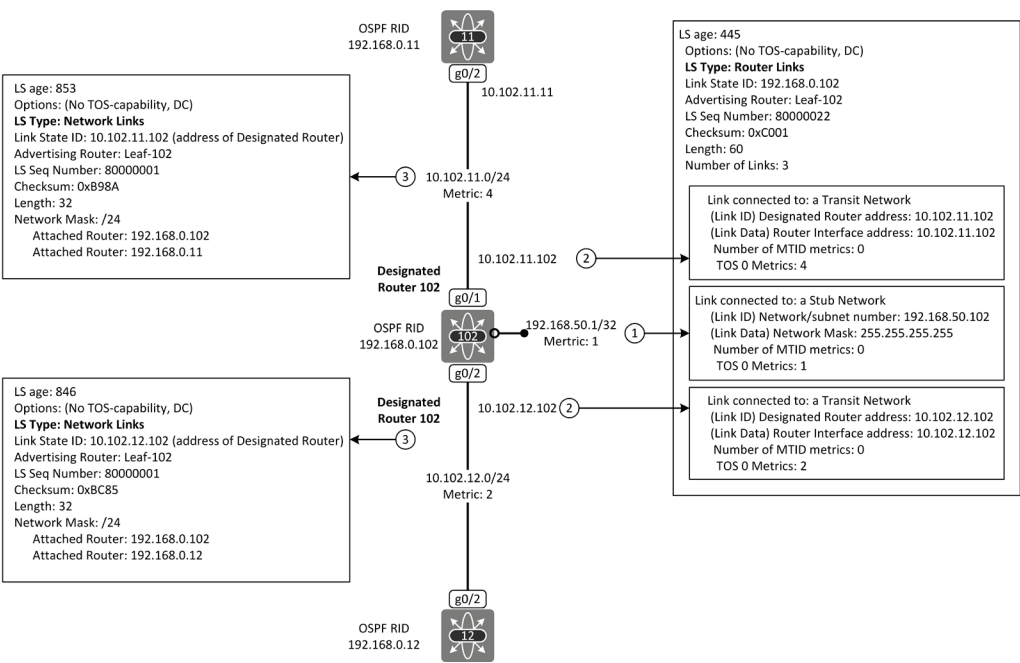


Figure 11-2: Router LSA and Network LSAs originated by Leaf-102 (DR).

The process of the DR/BDR election is useless in a segment where there are only two OSPF speakers. It also slightly increases the recovery times and generates an unnecessary Network LSAs. The first step in the LSDB optimization process is to change the OSPF Network type from Broadcast to Point-to-Point, where there is no DR/BDR election per segment and adjacent routers exchange only Router LSAs.

Figure 11-3 shows the Router LSAs generated by switch 102 after the OSPF network type is changed from Broadcast to Point-to-Point. Switch 102 describes its neighbor routers and the local interfaces by using Link Type-1 Router LSA (shown as “Link connected: to another router”). This interface information is used for routing. In addition, switch 102 describes the subnet to which the routers are connected to by using the Link Type-3 Router LSA (shown as “connected to Stub Network”). Both of these LSAs are flooded to every switch inside an OSPF area. The sub-networks are Transit networks and are not used for data traffic, which means that they consume unnecessary hardware resources and decreases the convergence time. These unnecessary Transit Network LSAs can be hidden by using Prefix-Suppression.

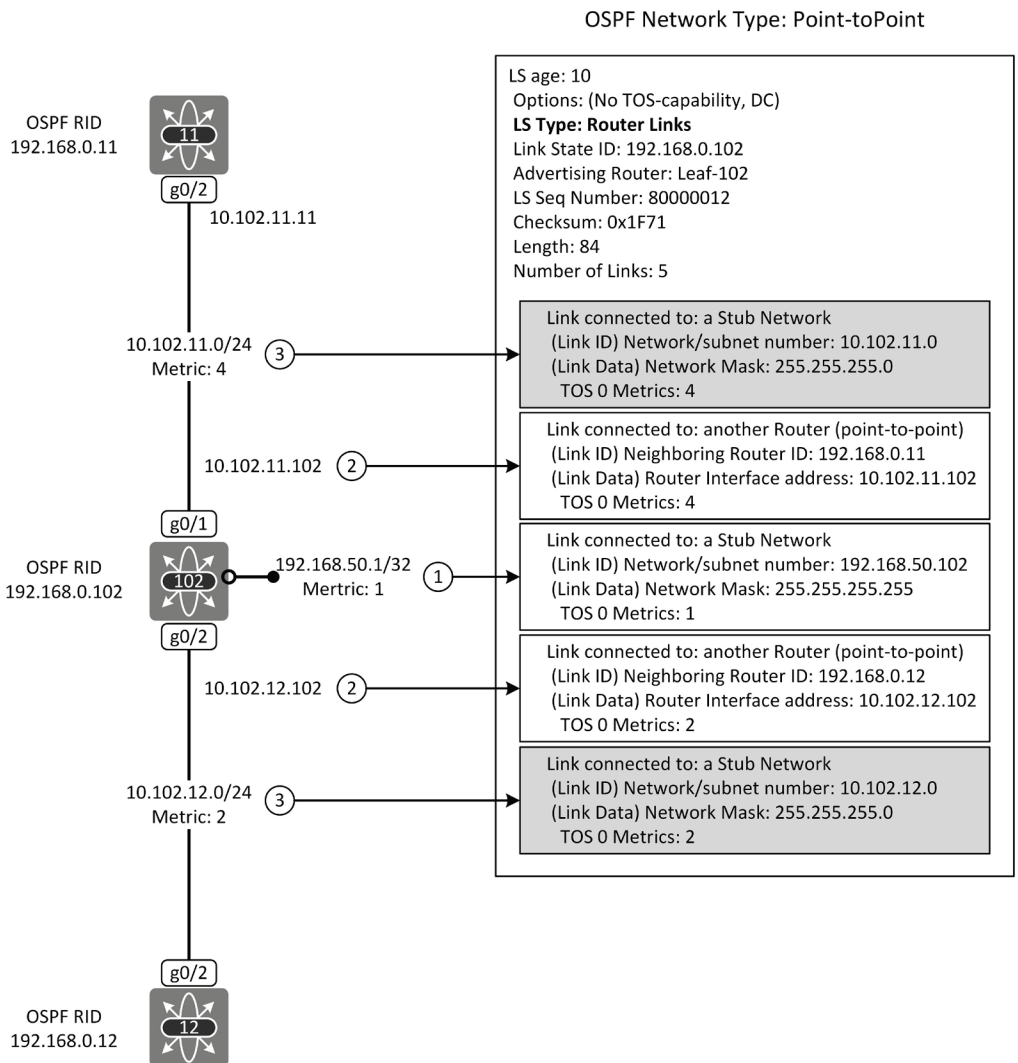


Figure 11-3: Router LSA originated by Leaf-102.

Example 11-1 illustrates the OSPF LSDB taken from 102 before prefix-suppression. The link count in Area 0 is 45.

```
Leaf-102#sh ip ospf database | b Link
```

Link ID count	ADV Router	Age	Seq#	Checksum	Link
192.168.0.11	Spine-11	31	0x80000003	0x00701F	8
192.168.0.12	Spine-12	18	0x80000006	0x007C05	8
192.168.0.101	Leaf-101	24	0x80000006	0x003B53	9
192.168.0.102	Leaf-102	62	0x80000004	0x003B63	5
192.168.0.103	Leaf-103	58	0x80000004	0x009EE6	5
192.168.0.104	Leaf-104	52	0x80000003	0x00DB8A	5
192.168.0.105	Leaf-105	45	0x80000003	0x00FD74	5

Example 11-1: OSPF LSDB from Leaf-102.

Example 11-2 shows the routing table of switch 102 before prefix-suppression. Each Transit network is learned via OSPF.

```
Leaf-102#sh ip route ospf
<snipped>
```

```

    10.0.0.0/8 is variably subnetted, 12 subnets, 2 masks
O      10.12.11.0/24 [110/4] via 10.102.12.12, 00:32:53, GigabitEthernet0/2
O      10.101.11.0/24 [110/6] via 10.102.12.12, 00:33:07, GigabitEthernet0/2
        [110/6] via 10.102.11.11, 00:33:07, GigabitEthernet0/1
O      10.101.12.0/24 [110/6] via 10.102.12.12, 00:32:53, GigabitEthernet0/2
O      10.103.11.0/24 [110/8] via 10.102.12.12, 00:33:07, GigabitEthernet0/2
        [110/8] via 10.102.11.11, 00:33:07, GigabitEthernet0/1
O      10.103.101.0/24
        [110/10] via 10.102.12.12, 00:32:43, GigabitEthernet0/2
        [110/10] via 10.102.11.11, 00:32:43, GigabitEthernet0/1
O      10.104.101.0/24
        [110/9] via 10.102.12.12, 00:33:31, GigabitEthernet0/2
O      10.105.12.0/24 [110/3] via 10.102.12.12, 00:32:53, GigabitEthernet0/2
O      10.105.104.0/24
        [110/5] via 10.102.12.12, 00:33:31, GigabitEthernet0/2
    192.168.50.0/32 is subnetted, 5 subnets
O      192.168.50.101 [110/7] via 10.102.12.12, 01:30:05, GigabitEthernet0/2
        [110/7] via 10.102.11.11, 01:29:25, GigabitEthernet0/1
O      192.168.50.103 [110/9] via 10.102.11.11, 01:29:25, GigabitEthernet0/1
O      192.168.50.104 [110/6] via 10.102.12.12, 01:29:35, GigabitEthernet0/2
O      192.168.50.105 [110/4] via 10.102.12.12, 01:29:45, GigabitEthernet0/2
```

Example 11-2: OSPF routes in RIB of Leaf-102.

Figure 11-4 shows the LSU packet and LSAs originated by switch 102 when the OSPF network type is changed to Point-to-Point in all Inter-Switch Links and the Transit networks are excluded from the LSAs by using prefix-suppression. When these changes are done in all OSPF switches, the LSDB optimization is ready. Example 11-3 shows the OSPF configuration related to LSAs optimization.


```

Leaf-102(config)# interface g0/1
Leaf-102(config-if)# ip ospf network point-to-point
Leaf-102(config-if)# ip ospf prefix-suppression

```

Example 11-3: Leaf-102 Interface g0/1 configuration.

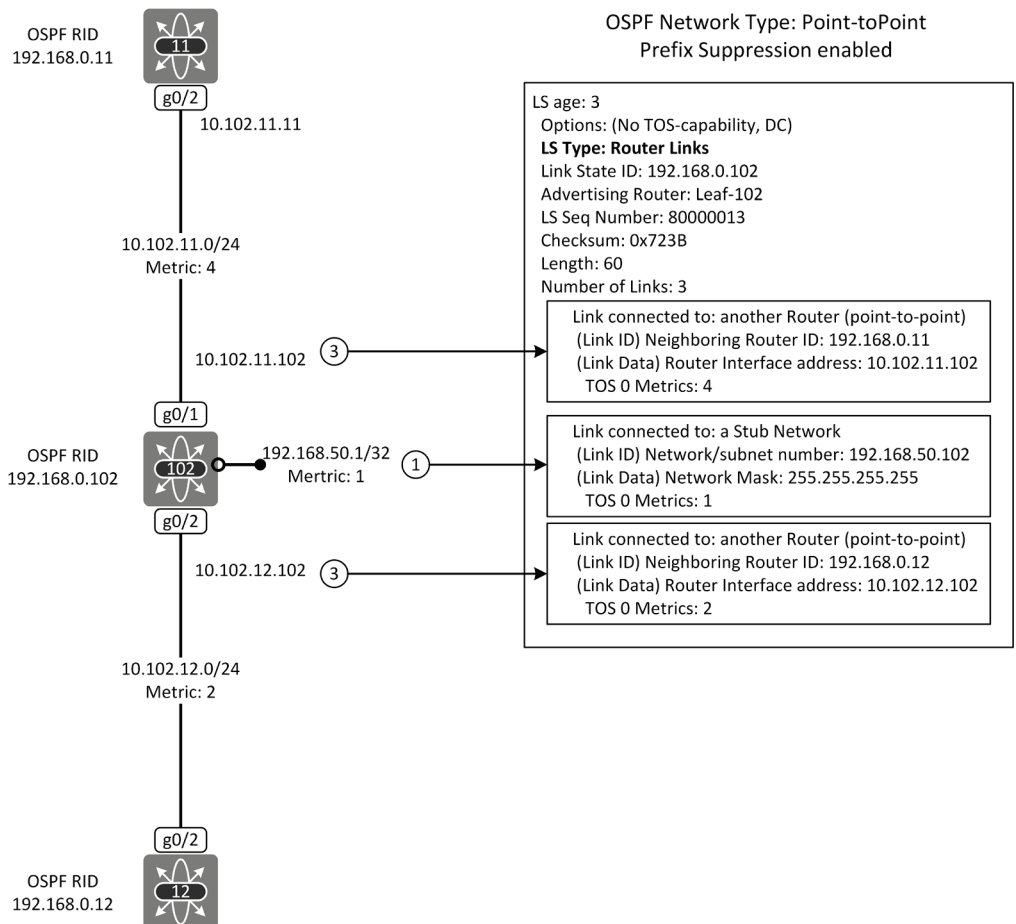


Figure 11-4: Router LSA originated by Switch 102.

Example 11-4 shows the OSPF LSDB taken from Leaf-102 after prefix-suppression. The Link count is now reduced from 45 to 25.

OSPF Router with ID (192.168.0.102) (Process ID 1)				
Router Link States (Area 0)				
Link ID count	ADV Router	Age	Seq#	Checksum Link
192.168.0.11	Spine-11	29	0x80000004	0x00432B 4
192.168.0.12	Spine-12	9	0x80000007	0x000959 4
192.168.0.101	Leaf-101	21	0x80000007	0x007B05 5
192.168.0.102	Leaf-102	69	0x80000005	0x008E2D 3
192.168.0.103	Leaf-103	68	0x80000005	0x00A857 3
192.168.0.104	Leaf-104	65	0x80000004	0x00AF8F 3
192.168.0.105	Leaf-105	44	0x80000004	0x00B13E 3

Example 11-4: *OSPF LSDB from switch 102 after prefix suppression.*

Example 11-5 shows that there are no Transit networks after the prefix-suppression in 102 Routing Information Base (RIB). The highlighted portion of example 11-5 shows that even though the local Transit networks and related IP addresses are hidden from adjacent switches, they still exist in the local RIB.

Leaf-102#sh ip route ospf	
<snipped>	
Gateway of last resort is not set	
10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks	
C	10.102.11.0/24 is directly connected, GigabitEthernet0/1
L	10.102.11.102/32 is directly connected, GigabitEthernet0/1
C	10.102.12.0/24 is directly connected, GigabitEthernet0/2
L	10.102.12.102/32 is directly connected, GigabitEthernet0/2
192.168.50.0/32 is subnetted, 5 subnets	
O	192.168.50.101 [110/7] via 10.102.12.12, 00:28:15, GigabitEthernet0/2
	[110/7] via 10.102.11.11, 00:27:55, GigabitEthernet0/1
C	192.168.50.102 is directly connected, Loopback50
O	192.168.50.103 [110/9] via 10.102.11.11, 00:27:55, GigabitEthernet0/1
O	192.168.50.104 [110/6] via 10.102.12.12, 00:28:05, GigabitEthernet0/2
O	192.168.50.105 [110/4] via 10.102.12.12, 00:28:05, GigabitEthernet0/2

Example 11-5: *RIB of Switch 102 After the Prefix-Suppression*

Shortest-Path First (SPF)/Dijkstra Algorithm

Dijkstra/SPF algorithm is used for calculating a Shortest-Path Tree (SPT) topology in OSPF Area. A router starts the process by setting itself as a root of the tree. At the first stage, the router builds a Shortest-Path Tree between routers by using the Type-1 Link Description (point-to-point) which describes links to neighbor routers in Router LSA. When the Shortest-Path Tree is formed, the router calculates the distance to subnets connected to each router by using the Link Type-3 (Stub) Link Description in Router LSA.

Routers have two lists related to SPT calculation. The *Candidate List* (also known as a Tentative List) is the list that includes all routers that are currently examined by the router. The Tree List (also called Path or Known List) is the list, which includes all the routers participating in a final Shortest-Path Tree. In addition, a Link State Database (LSDB) is a source from where the information is pulled to calculation and actually it is sometimes called the *Unknown List*.

The next section describes the SPT calculation process from switch 102 perspectives. Note that the switches are named as either Spine-xx or Leaf-xx.

SPF Run – Phase I: Building a Shortest-Path Tree

Figure 11-5 shows the initial situation where Leaf-102 starts the Shortest-Path Tree calculation. The Leaf-102 inserts itself into the Candidate-list with cost 0 and with next-hop pointing to itself. All other switches are in Unknown-list at this phase. The Path-list is empty at the initial situation. Table 11-1 illustrates the Unknown/Candidate/Path list progress in this stage.

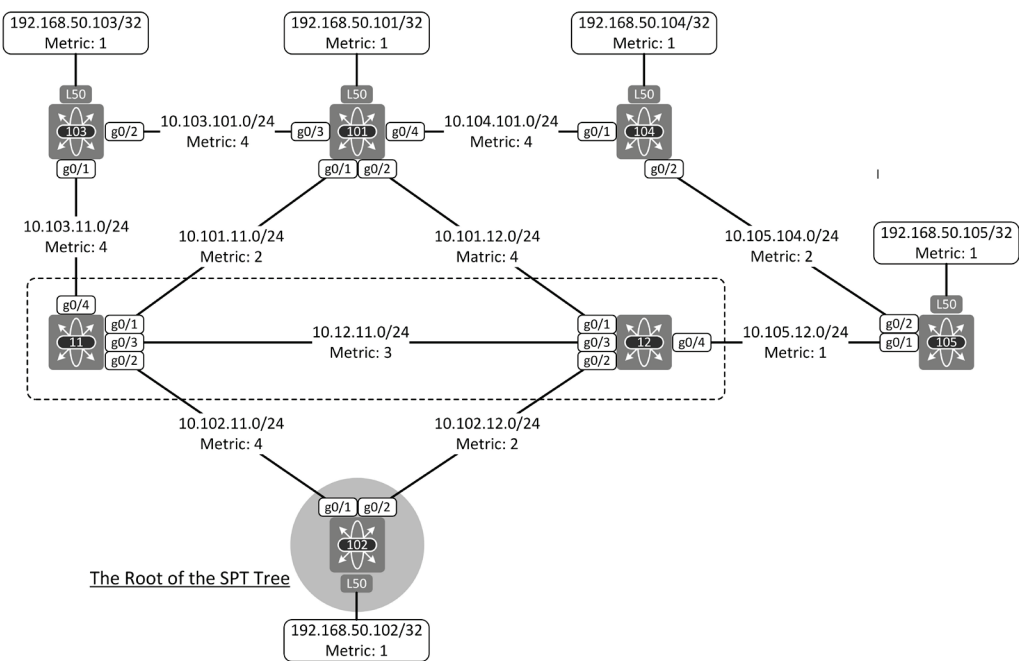


Figure 11-5: Shortest-Path Tree calculation: Starting Point

Unknown List (Routers in LSDB)	Candidate/Tentative List (Dst, Cost, Next-Hop)	Path/Known/Tree List (Dst, Cost, Next-Hop)
Spine-11 (192.168.0.11)	Leaf-102, 0, Leaf-102	
Spine-12 (192.168.0.12)		
Leaf-101 (192.168.0.101)		
Leaf-103 (192.168.0.103)		
Leaf-104 (192.168.0.104)		
Leaf-105 (192.168.0.105)		

Table 11-1: Shortest Path Tree: Starting point on Switch 102.

First iteration round

Figure 11-6 shows the first SPF iteration round. Leaf-102 inserts itself to the Path List (table 1-2 step-A). Leaf-102 examines its self-originated Router LSA. It starts from the first Link Description (LD) found from the LSA. First LD is Link Type-3 (Stub) so Leaf-102 ignores it (1). The next entry describes the link to Spine-11 (Link Type-1), which is adjacent via local interface g0/1. Leaf-102 moves Spine-11 into the Candidate-list with cost 4 via interface g0/1 (2). The last LD describes the link to Spine-12 that is reachable via interface g0/2. Leaf-102 move also Spine-12 to the Candidate List (3). Leaf switches 101, 103-105 are still in the Unknown-list.

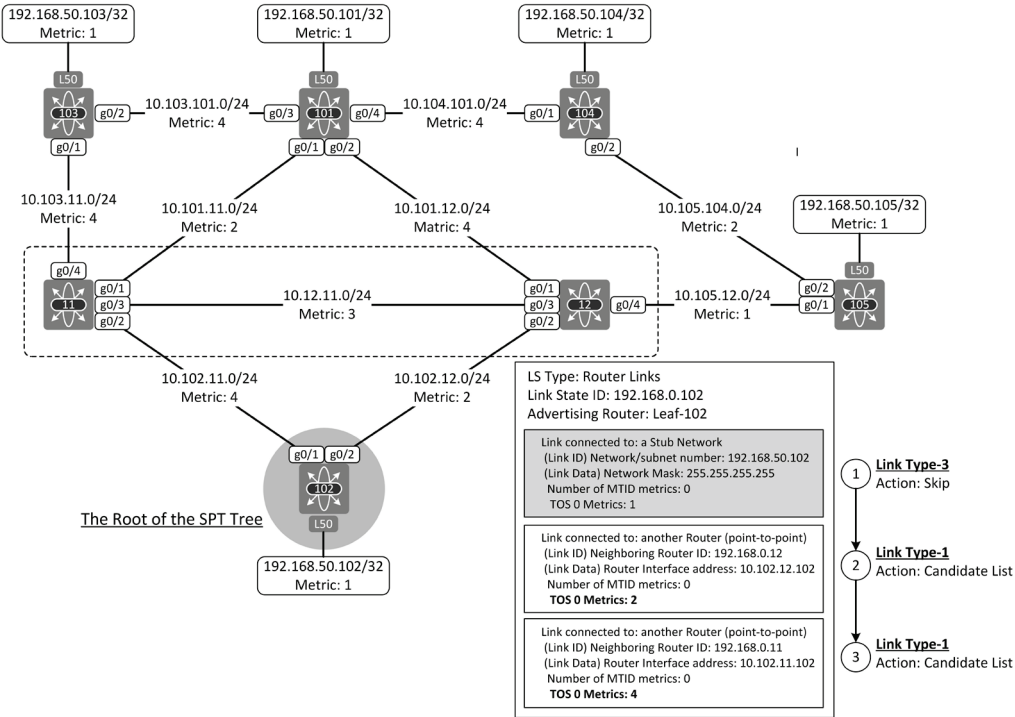


Figure 11-6: Shortest-Path Tree calculation: First iteration round on Leaf-102.

Unknown List (Routers in LSDB)	Candidate/Tentative List (Dst, Cost, Next-Hop)	Path/Known/Tree List (Dst, Cost, Next-Hop)
		(A) Leaf-102, 0, Leaf-102
	(2) Spine-11, 4, Gi0/1	
Leaf-101 (192.168.0.101)	(3) Spine-12, 2, Gi0/2	
Leaf-103 (192.168.0.103)		
Leaf-104 (192.168.0.104)		
Leaf-105 (192.168.0.105)		

Table 11-2: Shortest Path Tree: 1st. Iteration on Leaf-102.

Second iteration round

Leaf-102 moves Spine-12 to the Path-list (B) because it has the lowest cost among all the devices listed on the Candidate-list (B). Leaf-102 starts checking the LSA advertised by Spine-12. First LD describes the link to Leaf-105 with cost 1. Leaf-102 inserts Leaf-105 into Candidate-list (1). Next LD describes the link to Spine-11 with cost 3 (total cost 5). There is already an entry with better cost in the Candidate-list (b), so Leaf-102 skip this LD (2). Next one describes to link back to Leaf-102, so Leaf-102 skip it also (3). Last LD points to Leaf-101 with cost 4 (total cost 8). This information is added to the Candidate-list by Leaf-102 (4). Leaf switches 103-104 are still in the Unknown-list.

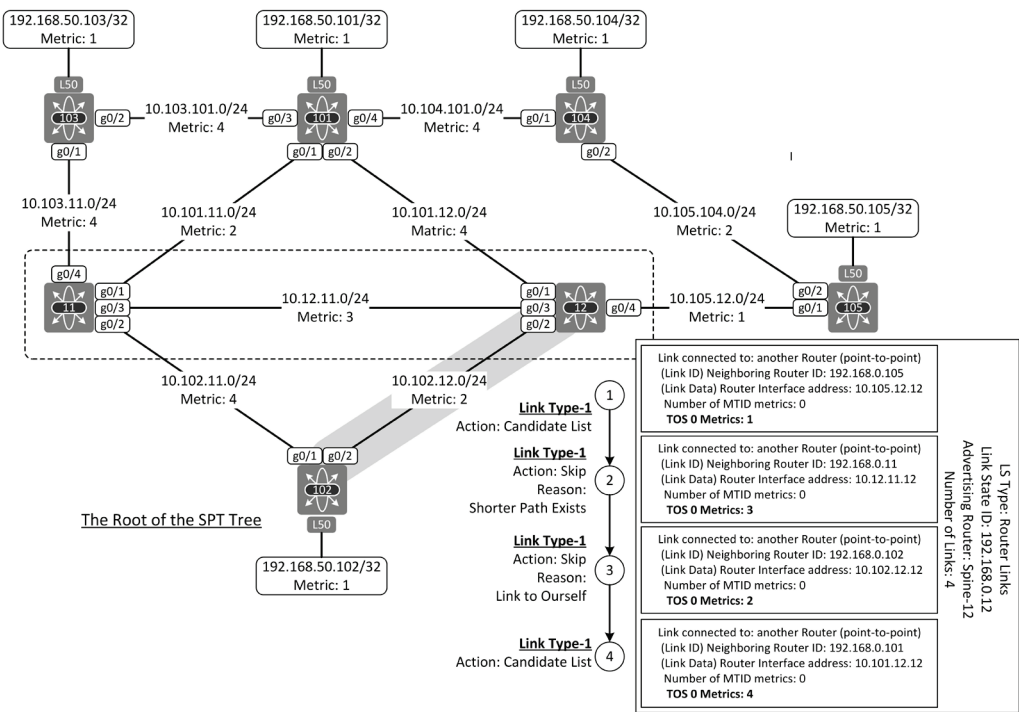


Figure 11-7: Shortest-Path Tree calculation: Second iteration round on Leaf-102.

Unknown List (Routers in LSDB)	Candidate/Tentative List (Dst, Cost, Next-Hop)	Path/Known/Tree List (Dst, Cost, Next-Hop)
		Leaf-102, 0, Leaf-102
	(b) Spine-11, 4, Gi0/1	(B) Spine-12, 2, Gi0/2
Leaf-101 (192.168.0.101)	(1) Leaf-105, 3, Spine-12	
Leaf-103 (192.168.0.103)	(4) Leaf-101, 6, Spine-12	
Leaf-104 (192.168.0.104)		

Table 11-3: Shortest-Path Tree calculation: Second iteration round on Leaf-102

Third iteration round

Leaf-102 moves Leaf-105 to the Path-list (C) because it has the lowest cost among all devices listed on the Candidate-list. Leaf-102 starts checking the LSA advertised by Leaf-105. First LD describes the Stub Network (Link Type-3) and it is skipped (1). Next LD describes the link to Leaf-104. Leaf-102 add Leaf-104 into Candidate-list with total cost 5 (2+1+2) with Spine-12 as a next-hop. The last LD describes the link to Spine-12, which already is installed into Path-list with better cost (3). Only Leaf-103 is now in the Unknown-list.

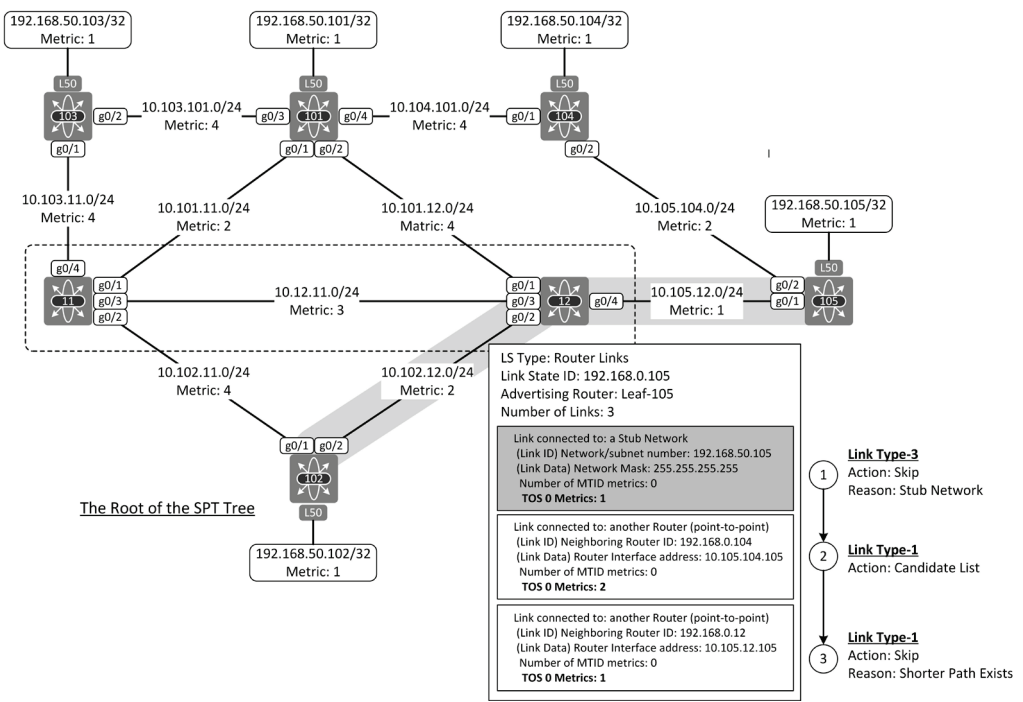


Figure 11-8: Shortest-Path Tree calculation: Third iteration round on Leaf-102.

Unknown List (Routers in LSDB)	Candidate/Tentative List (Dst, Cost, Next-Hop)	Path/Known/Tree List (Dst, Cost, Next-Hop)
		Leaf-102, 0, Leaf-102
	Spine-11, 4, Gi0/1	Spine-12, 2, Gi0/2
Leaf-101 (192.168.0.101)	Leaf-101, 6, Spine-12	(C) Leaf-105, 3, Spine-12
Leaf-103 (192.168.0.103)	(2) Leaf-104, 5, Spine-12	

Table 11-4: Shortest-Path Tree calculation: Third iteration round on Leaf-102.

Fourth iteration round

Leaf-102 moves Spine-11 to the Path-list (D) because it has the lowest cost among all the devices listed on the Candidate-list. Leaf-102 starts checking the LSA originated by Spine-11. First LD describes the link to Leaf-103 (1). Leaf-102 add Leaf-103 to Candidate-list with total cost 8 (4+4) and Spine-11 as a next-hop. Next LD is about Spine-12 that already exists on the Path-list with a better cost. Leaf-102 skips this one. Next LD describes the link back to Leaf-102 so it is also skipped. The fourth LD describes the link to Leaf-101 with the same total cost 6 than what Spine-12 has described. Leaf-102 updates Candidate-list concerning Leaf-101, which now has two equal-cost paths via Spine-11 and Spine-12.

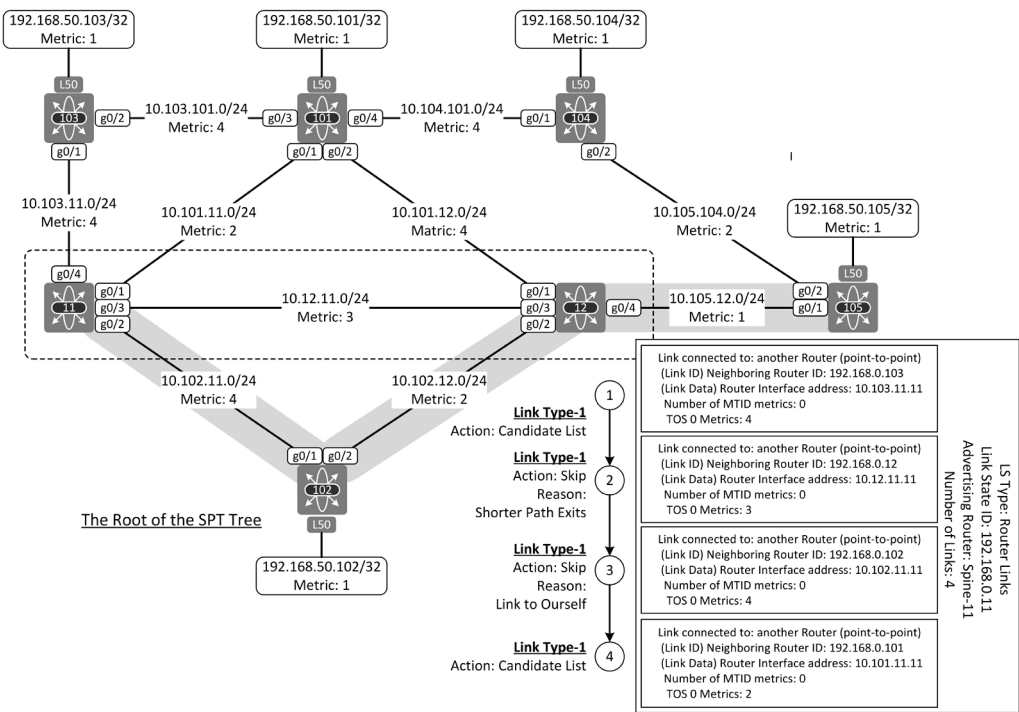


Figure 11-9: Shortest-Path Tree calculation: Fourth iteration round on Leaf-102.

Unknown List (Routers in LSDB)	Candidate/Tentative List (Dst, Cost, Next-Hop)	Path/Known/Tree List (Dst, Cost, Next-Hop)
	Leaf-101, 6, Spine-11	Leaf-102, 0, Leaf-102
	Leaf-101, 6, Spine-12	Spine-12, 2, Gi0/2
	Leaf-104, 5, Spine-12	Leaf-105, 3, Spine-12
	(1) Leaf-103, 8, Spine-11	(D) Spine-11, 4, Gi0/1

Table 11-5: Shortest-Path Tree calculation: Fourth iteration round on Leaf-102.

Fifth iteration round

Leaf-102 moves Leaf-104 from the Candidate-list to Path (total cost 5, next-hop Spine-12) because it has the lowest cost among all the devices listed on the Candidate List (E). The first LD of the LSA generated by Leaf-104 is a Stub Network (Link Type-3) so it is ignored by Leaf-102. Next LD describes the link to Leaf-105 (total cost 5, next-hop Spine-12), which already is added to Path-list with a better cost. Leaf-102 ignores this one too. The last LD describes the link to Leaf-101 (total cost 9, next-hop Spine-12), which already is in Candidate-list with better total costs via Spine-11 and Spine-12, so also the last LD is ignored. None of the LD described in LSA originated by Leaf-104 does not end up to the Path-list.

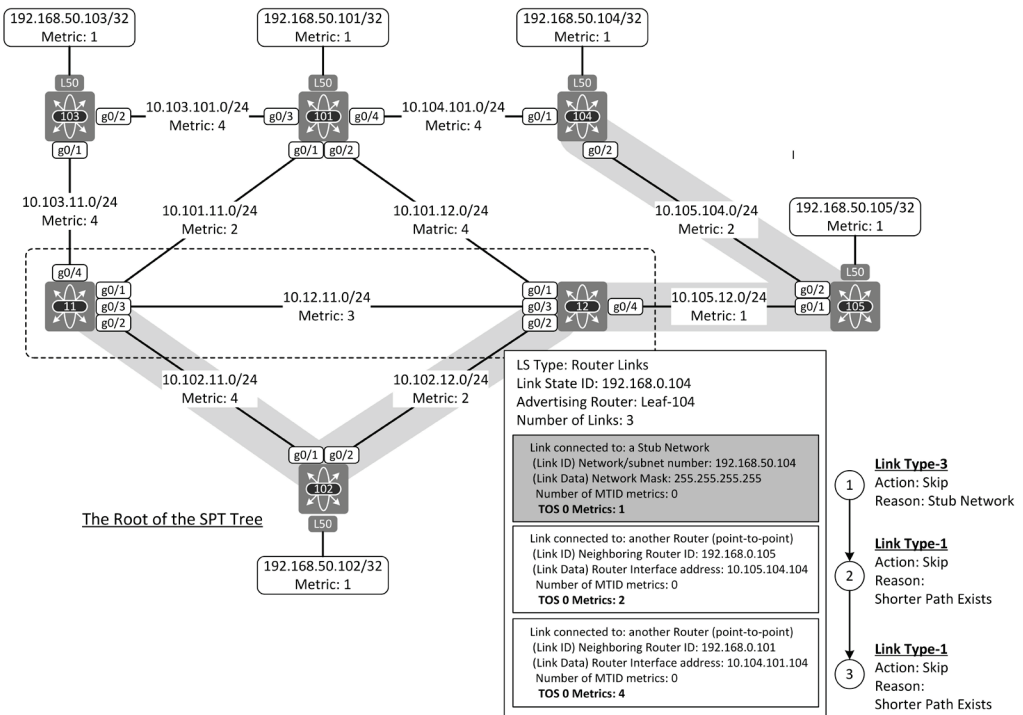


Figure 11-10: Shortest-Path Tree calculation: Fifth iteration round on Leaf-102.

Unknown List (Routers in LSDB)	Candidate/Tentative List (Dst, Cost, Next-Hop)	Path/Known/Tree List (Dst, Cost, Next-Hop)
	Leaf-101, 6, Spine-11	Leaf-102, 0, Leaf-102
	Leaf-101, 6, Spine-12	Spine-12, 2, Gi0/2
	Leaf-103, 8, Spine-11	Leaf-105, 3, Spine-12
		Spine-11, 4, Gi0/1
		(E) Leaf-104, 5, Spine-12

Table 11-6: Shortest-Path Tree calculation: Fifth iteration round on Leaf-102.

Sixth iteration round

Leaf-102 moves Leaf-101 from the Candidate-list to Path-list with total cost 6 via Spine-11 and Spine-12 (F). First LD originated by Leaf-101 is Stub Network and it is skipped from Shortest-Path Tree calculation. The next four LDs describes the links to switches Leaf-103, Leaf-104, Spine-11, and Spine-12. Leaf-102 has a better path with lower costs related to these switches and this is why all four Link Descriptions are ignored links. Just like in the case of Leaf-104, none of the LD described in LSA originated by Leaf-101 does not end up to the Path-list.

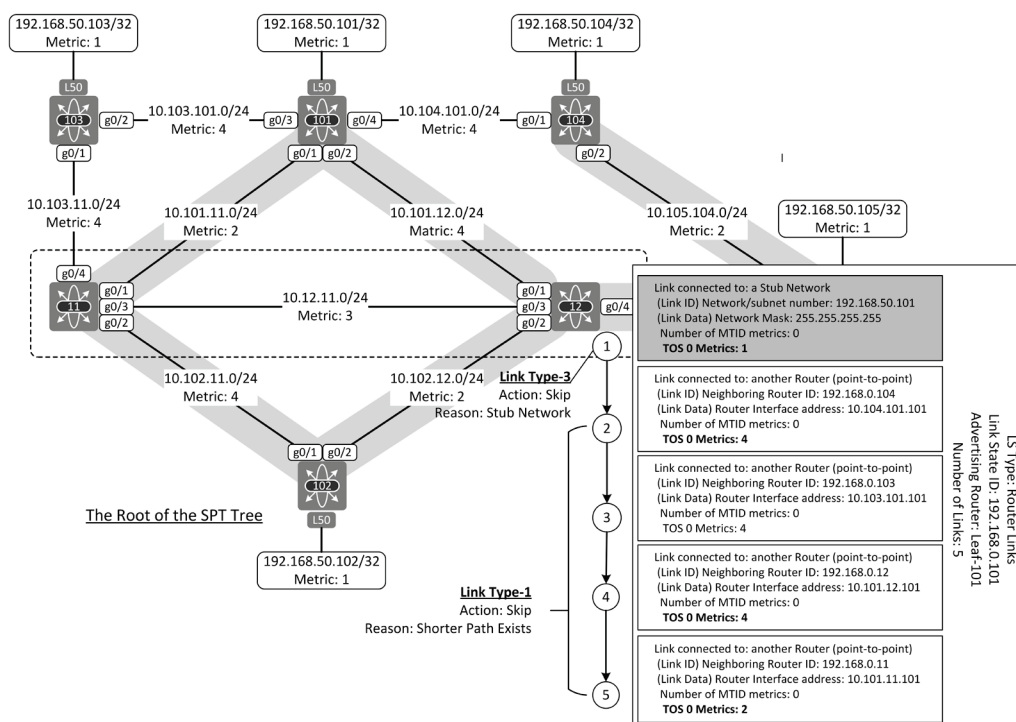


Figure 11-11: Shortest-Path Tree calculation: Sixth iteration round on Leaf-102.

Unknown List (Routers in LSDB)	Candidate/Tentative List (Dst, Cost, Next-Hop)	Path/Known/Tree List (Dst, Cost, Next-Hop)
		Leaf-102, 0, Leaf-102
		Spine-12, 2, Gi0/2
	Leaf-103, 8, Spine-11	Leaf-105, 3, Spine-12
		Spine-11, 4, Gi0/1
		Leaf-104, 5, Spine-12
		(F) Leaf-101, 6, Spine-11 and Spine-12 (ECMP)

Table 11-7: Shortest-Path Tree calculation: Sixth iteration round on Leaf-102.

Seventh iteration round

Leaf-102 moves the last switch Leaf-103 Candidate-list (G) to the Path-list with a total cost of 8 and Spine-11 as a next-hop. The first LD describes the Stub Network, so it is excluded from Shortest-Path Tree calculation. The second LD describes the link to Leaf-101, which already is in Path-list with a better metric, so it is also left out from the SPT calculation. The last LDs describe the link to Spine-11, which also is in Path List with a better overall metric. At this stage, the Shortest-Path Tree is ready and Leaf-102 starts phase two where it calculates the Shortest-Paths to Stub Networks.

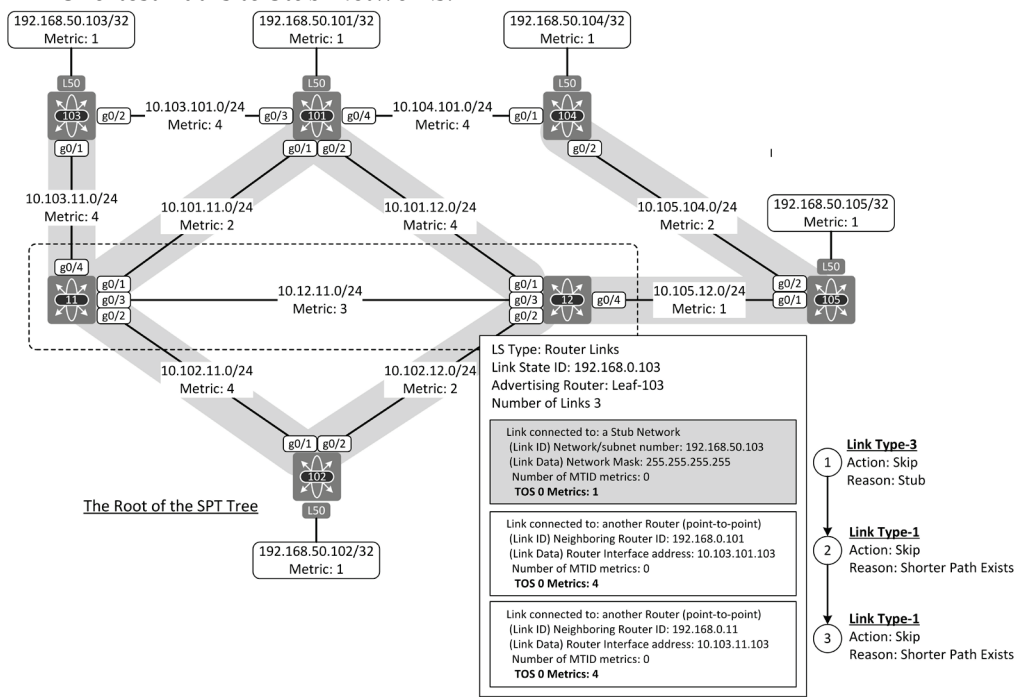


Figure 11-12: Shortest-Path Tree calculation: Seventh iteration round on Leaf-102.

Unknown List (Routers in LSDB)	Candidate/Tentative List (Dst, Cost, Next-Hop)	Path/Known/Tree List (Dst, Cost, Next-Hop)
		Leaf-102, 0, Leaf-102
		Spine-12, 2, Gi0/2
		Leaf-105, 3, Spine-12
		Spine-11, 4, Gi0/1
		Leaf-104, 5, Spine-12
		Leaf-101, 6, Spine-11 and Spine-12 (ECMP)
		(G) Leaf-103, 8, Spine-11

Table 11-8: Shortest-Path Tree calculation: Seventh iteration round on Leaf-102.

SPF Run – Phase II: Adding Leafs to Shortest-Path Tree

At first phase, Leaf-102 forms a Shortest-Path Tree (SPT) by using the Dijkstra/SPF algorithm. In the second Phase, Leaf-102 adds Stub Networks (leafs) to SPT. Leaf-102 starts by examining its self-originated Router LSA where there is a Link Description about Stub Network 192.168.50.102/32. This Stub Network is moved to Path-list (Path-list updates are not included in table 1-9 for simplicity). This information is also installed into Routing Information Base (RIB) as a connected network. Next, Leaf-102 examines the Router LSA originated by the Spine-12, which is the closest OSPF speaker found from the Path-list. There is no Link Description about Stub Networks, so neither Path List nor RIB is updated. Leaf-102 moves on to the next closest OSPF speaker found from the Path List and checks the Router LSA originated by Leaf-105. The Router LSA originated by Leaf-105 includes the Stub Network 192.168.50.105/32. Leaf-102 adds it to Path-list and updates the RIB. Next, OSPF speaker on the Path-list is Spine-11, which Router LSA does not describe any Stub Network, so Leaf-102 moves on to check the Router LSA originated by Leaf-104. The Router LSA of Leaf-104 includes the Stub Network 192.168.50.104/32 which is inserted into the Path-list and into the RIB of Leaf-102. After this, Leaf-102 examines the Router LSA originated by Leaf-101. There is a Link Description about Stub Network 192.168.50.101/32. Leaf-101 is reachable via two equal-cost paths via Spine-11 and Spine-12 so the Stub Network is added into the Path-list and into the RIB with two next-hops. The last OSPF speaker listed in the Path List is Leaf-103, which has Stub Network 192.168.50.103/32 described in its Router LSA. Leaf-102 adds the Stub Network into its Path-list and into the RIB.

Now the Shortest-Path Tree with its Leafs is ready from the Leaf-102 perspective.

	Path/Known/Tree List (Dst, Cost, Next-Hop)	Stub Networks (Router LSA: LT- 3)	Routing Information Base (Network [AD/C] NH Int,)
1	Leaf-102, 0, Leaf-102	192.168.50.102/32	192.168.50.102/32 directly connected
2	Spine-12, 2, Gi0/2		
3	Leaf-105, 3, Spine-12	192.168.50.105/32	192.168.50.105/32 [110/4] 10.102.12.12 via g0/2
4	Spine-11, 4, Gi0/1		
5	Leaf-104, 5, Spine-12	192.168.50.104/32	192.168.50.104/32 [110/6] 10.102.12.12 via g0/2
6	Leaf-101, 6, Spine-11 Spine-12	192.168.50.101/32	192.168.50.101/32 [110/5] 10.102.12.12 via g0/1 [110/5] 10.102.12.12 via g0/2
7	Leaf-103, 8, Spine-11	192.168.50.103/32	192.168.50.103/32 [110/9] 10.102.12.11 via g0/1

Table 11-9: Adding leaves to Shortest-Path Tree calculation on Leaf-102.

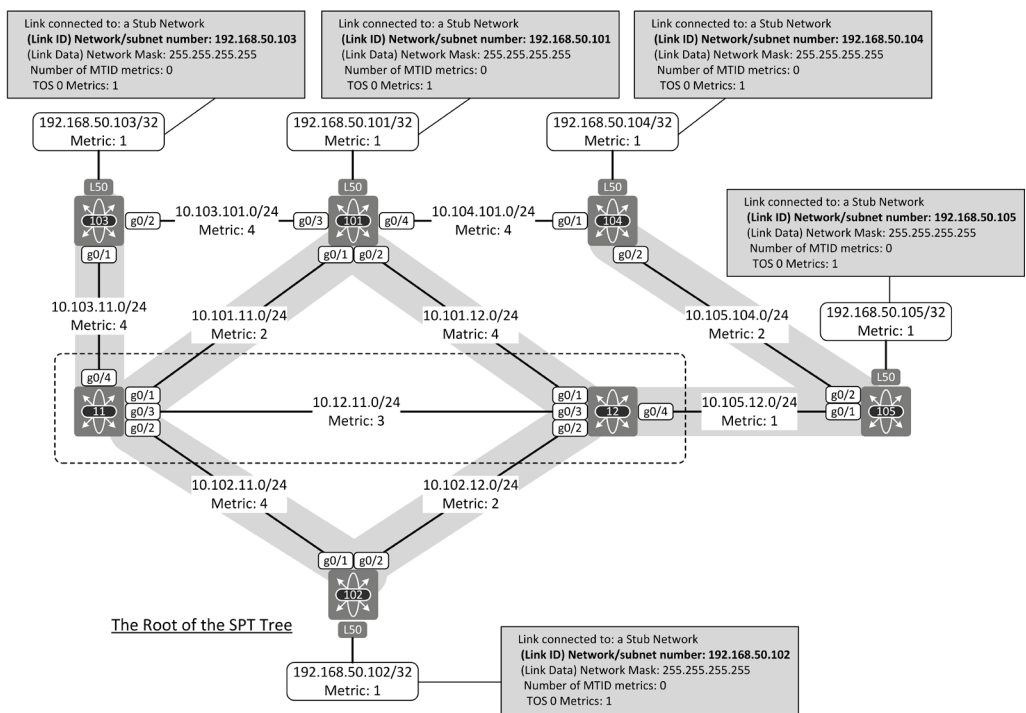


Figure 11-13: Adding leaves to Shortest-Path Tree calculation on Leaf-102.

So far, only the Shortest-Path Tree calculation process is introduced from the Leaf-102 perspective. All other switches in example network naturally execute the same process based on the information found from the Link State Database (LSDB). Even though the LSDBs in each OSPF speaker inside an OSPF area identical (and they have to be) it does not mean that each OSPF speaker has an identical Shortest-Path Tree and RIB. The last table of this chapter illustrates the cost calculated by each Leaf switches. Spine switches are excluded since they do not have any Stub Networks.

To\From	Leaf-101	Leaf-102	Leaf-103	Leaf-104	Leaf-105
Leaf-101	0	6	4	4	5
Leaf-102	6	0	8	6	3
Leaf-103	4	8	0	4	9
Leaf-104	4	5	8	0	2
Leaf-105	5	3	9	2	0
192.168.50.101/32	0	7	5	5	6
192.168.50.102/32	7	0	9	7	4
192.168.50.103/32	5	9	0	5	10
192.168.50.104/32	5	6	9	0	3
192.168.50.105/32	6	4	10	3	0

Table 11-10: Cost table between Leaf-to-Leaf and Leaf-to-Stub Network.

References:

- [RFC 1195] R. Callon, "Use of OSI IS-IS for Routing in TCP/IP and Dual Environments", RFC 1195, December 1990.
- [RFC 2328] J. Moy, "OSPF Version 2", RFC 2328, April 1998.
- [RFC 6860] Y. Yang et al., "Hiding Transit-Only Networks in OSPF", RFC 6860, January 2013.
- [RFC 8365] A. Sajassi et al. "A Network Virtualization Overlay Solution Using Ethernet VPN (EVPN)", RFC 8363, March 2018.
- [ISIS-OSPF] M. Bhatia et al., "IS-IS and OSPF Difference Discussion", Draft-bhatia-manral-isis-ospf-01" January 2006.

Chapter 12: Underlay Network – Comparison of OSPF and IS-IS

This chapter discusses the differences between the OSPF and the IS-IS from the Network Virtualization Overlay (NVO) solution, especially from the VXLAN network perspective. First, this chapter shortly introduces some of the differences between these two protocols (terminology, timers, and LSAs). Next, this chapter explains the default behavior of the Shortest Path First (SPF) by explaining first the IS-IS reaction when Stub Network goes down. Then the same event is explained from the OSPF perspective. This chapter also introduces the OSPF reaction when an Incremental SPF (iSPF) is enabled, and the interface on a link that is not belonging to the Shortest-Path Tree (SPT) goes down. The same event is also discussed with and without iSPF concerning IS-IS.

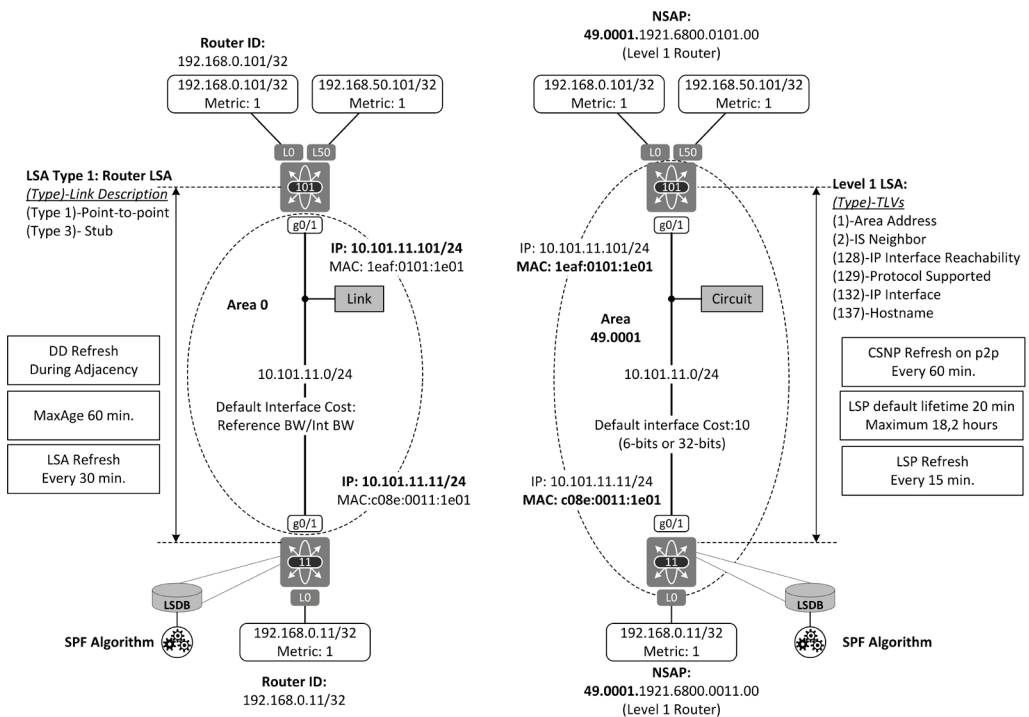


Figure 12-1: Comparison of OSPF and IS-IS.

Figure 12-1 illustrates some of the differences between the OSPF and the IS-IS Link-State Protocols. OSPF routers are identified by Router-Id (Usually a Loopback 0) while IS-IS routers are identified by NSAP address (Network Service Access Point). There is no relationship in the OSPF RID and OSPF Area numbering, while the first 13 bits in an NSAP address defines an IS-IS area and the next 48 bits identify a router (usually a Loopback 0). OSPF uses Link State Updates (LSU) packets, which carry LSAs (there are several different types of LSAs), which in turn describes links connected to the router. IS-IS uses a single Link State PDU (LSP), which carries different TLVs (Type, Length, Value) fields where router describes things like its hostname (RFC5301), the IS-IS area, IP addresses of links, and IPv4 prefixes that are directly connected to the router. From the router processing perspective fixed fields in OSPF LSAs do not require as much processing effort than variable length fields in IS-IS TLVs. OSPF uses Database Description (DD) to give a shortcut of its Links State Database (LSDB) to its' peer during adjacency negotiation (Exchange state). IS-IS uses Complete Sequence Number Packet (CSNP) for the same purpose but it is refreshed every 60 minutes by default on P2P links. OSPF LSA has a fixed 60 minutes lifetime that cannot be changed and each LSA is refreshed every 30 minutes by default. IS-IS LSP lifetime is 20 minutes by default and it can change all the way up to 18.2 hours. IS-IS LSPs are refreshed every 15 minutes by default and it launches the SPF algorithm (range 1-65535 seconds). To form an OSPF adjacency (and maintain it) between OSPF speaker the following attributes in Hello-packets must match; area-id, timers (hello and dead interval), authentication, network mask, stub-flags, options, Interface MTU and OSPF network type. In addition, the interface IP address and OSPF router-id has to be unique in the adjacent router. While OSPF has nine attributes that have to match, IS-IS only requires four matching attributes: Interface MTU, Levels, The Area-Id in Level 1 router and authentication. Also, System-Id has to be unique. Among these things, there are differences in area structure. OSPF area boundary falls on the router while in IS-IS the link is the border. The default metric in OSPF is calculated based on defined reference bandwidth/Interface bandwidth while in IS-IS the default interface metric is 10. In addition, IS-IS can signal memory overload by using "overload-bit", OSPF does not have this option. There are many other differences between the OSPF and the IS-IS but there are also similarities. All router inside an area, whether we are speaking about OSPF or IS-IS, has to have a common LSDB. Both protocols use the SPF algorithm to form a loop-free Shortest-Path Tree (SPT) between each router inside an area as well as adding leafs (networks) to SPT. Based on the calculation result, routers also updates their Routing Information Bases (RIB) as well as Forwarding Information Bases (FIB) if there are no better route sources like static routes available.

Figure 12-2 shows the example topology where both OSPF and IS-IS are running on each router and both protocols calculate the Shortest-Path (Links participating SPT are highlighted with grey color). The topology is based on the Spine-Leaf model with the difference that there is a link between the two Spine switches. The purpose of the Spine-to-Spine link is to demonstrate the SPF process when the link that is not participating in Shortest-Path Tree goes down. This kind of cabling structure might be relevant in some enterprise networks. There are six scenarios in this chapter. First, this chapter explains the operation of the SPF algorithm from the IS-IS perspective when the loopback 50 interface on Leaf-101 goes down (Partial SPF). Then the same event is monitored from the OSPF perspective (Full SPF). Next, this section shows how the OSPF SPF algorithm can be optimized (Incremental SPF). Hence, this section explains how OSPF reacts when the link between Spine-11 and Spine-12 goes down when iSPF is enabled. The last section explains the same operation from the IS-IS perspective and also introduces the IS-IS SPF optimization using iSPF.

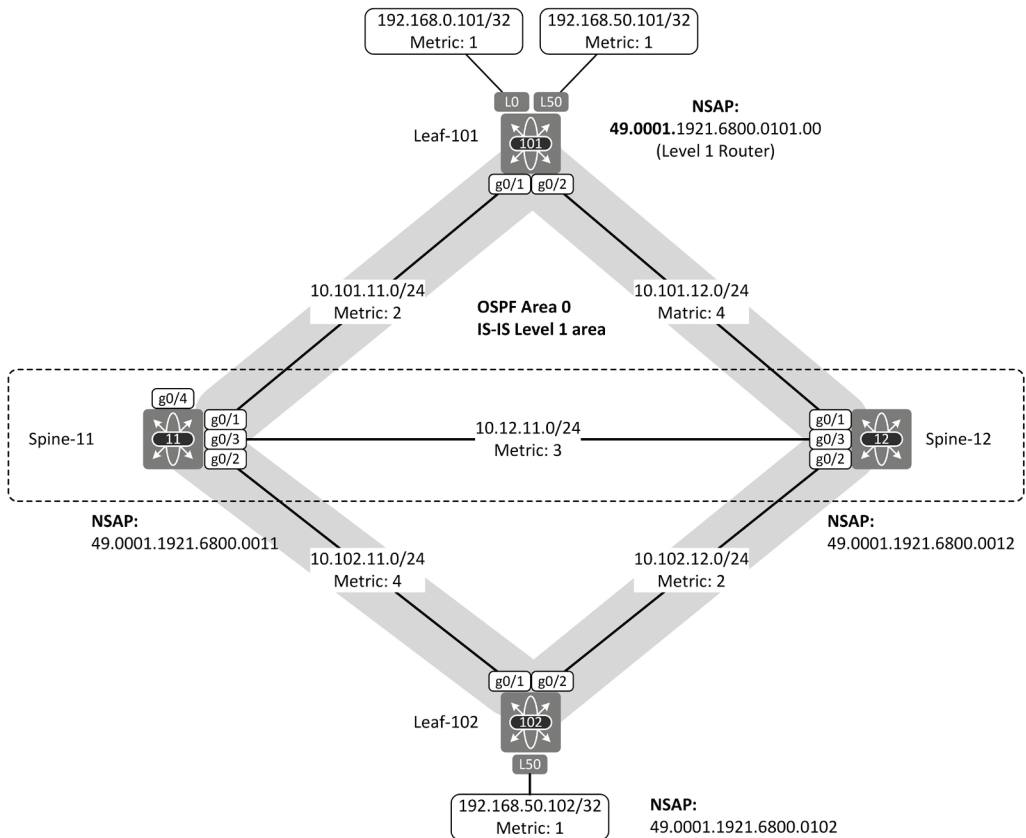


Figure 12-2: Example topology.

Scenario-1: Interface loopback 50 down on Leaf-101 (IS-IS)

Figure 12-3 illustrates the situation where the Loopback 50 interface (192.168.50.101/32) state is moved from Up to Down on Leaf-101. As a reaction to this event, Leaf-101 generates a new LSP and sends it to IS-IS neighbors. The “IP Internal Reachability Information” (Type-128) describes the network that is directly connected to the IS-IS router. The IP network 192.168.50.101 is excluded from this LSP. The packet capture is taken from the interface g0/1 on Leaf-101. LSP is flooded to Leaf-102 by both Spine switches. When Leaf-102 receives the LSP, instead of executing full SPF run by processing each LSA from its LSDB it only processes the new LSP. Leaf-102 ages out the old LSP from the LSDB and installs the information received on the newest LSP into LSDB. Also, it removes routes describing the reachability of net 192.168.50.101 from its RIB and FIB. IS-IS does not run the full SPF where the whole SPT is recalculated when the Leaf route is changed. This is why the term Partial SPF is used when speaking about the SPF run concerning IS-IS.

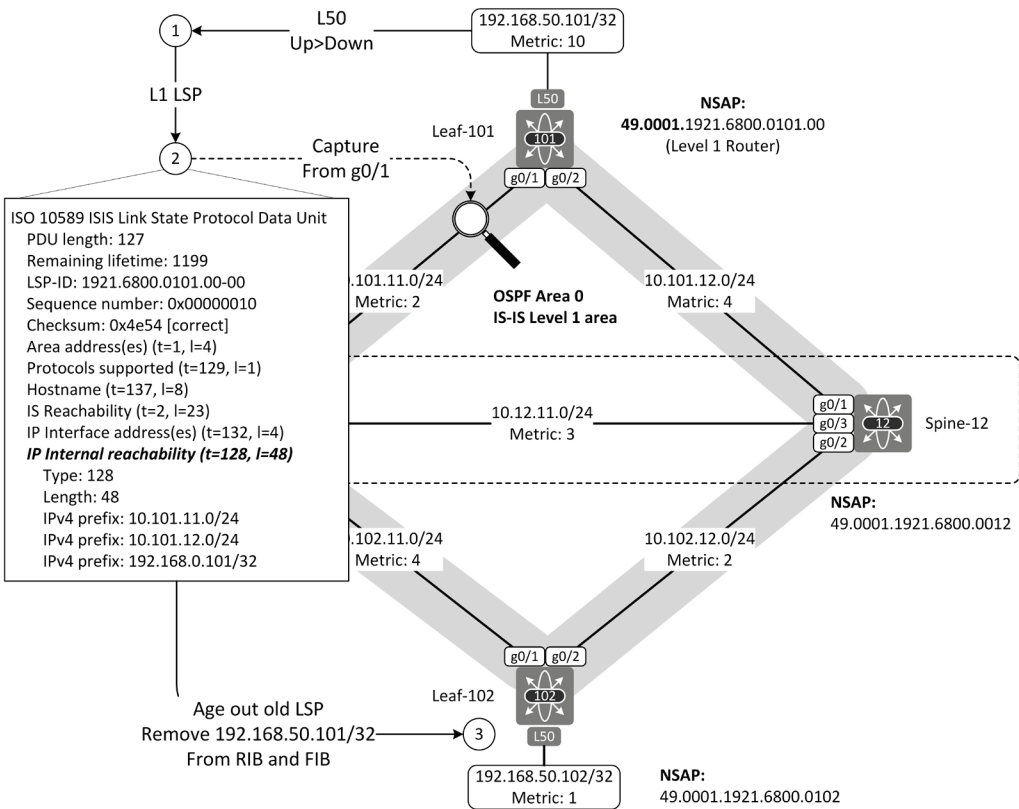


Figure 12-3: Interface Loopback 50 DOWN on Leaf-101 (IS-IS).

Example 12-1 shows the debug information in Leaf-102 when it receives the new LSP originated by Leaf-101.

```
Leaf-102#debug isis spf-events
IS-IS SPF events debugging is on for IPv4 unicast topology base
ISIS-SPF: L1 LSP 3 (1921.6800.0101.00-00) flagged for recalculation
from 252321F
ISIS-SPF: LSP 3 (1921.6800.0101.00-00) Type STD
ISIS-SPF: spf_result: next_hop_parents:0x1050133C root_distance:6,
parent_count:2, parent_index:4 db_on_paths:1
ISIS-SPF: Calculating routes for L1 LSP 3 (1921.6800.0101.00-00)
ISIS-SPF: lsptype:0, current_lsp(1921.6800.0101.00-00) (3)
current_lsp:0xDABACA8, lsp_fragment:0xDABACA8 calling isis_walk_lsp
ISIS-SPF: Aging L1 LSP 3 (1921.6800.0101.00-00), version 15
```

Example 12-1: “Debug isis spf-events” on Leaf-102

Example 12-2 shows the LSDB concerning information received from Leaf-101. NLPID (Network Layer Protocol Identifier) describes the supported protocols (0xCC = IPv4). NLPID is shown in packet capture as “Supported Protocols” in figure 2-3. Also, the hostname of the originating router is carried in TLVs. To be able to see hostname in OSPF LSDB, hostname-to-IP address mapping information has to be configured to each router.

```
Leaf-102#show isis database level-1 Leaf-101.00-00 detail

Tag UnderlayNet:

IS-IS Level-1 LSP Leaf-101.00-00
LSPID                               LSP Seq Num    LSP Checksum   LSP Hold-time/Rcvd
ATT/P/OL
Leaf-101.00-00                      0x00000010    0x4E54         1178/1198      0/0/0
Area Address: 49.0001
NLPID: 0xCC
Hostname: Leaf-101
Metric: 4 IS Spine-12.00
Metric: 2 IS Spine-11.00
IP Address: 192.168.0.101
Metric: 2 IP 10.101.11.0 255.255.255.0
Metric: 4 IP 10.101.12.0 255.255.255.0
Metric: 4 IP 10.103.101.0 255.255.255.0
Metric: 0 IP 192.168.0.101 255.255.255.255
```

Example 12-2: “show isis database level-1 Leaf-101.00-00 detail” on Leaf-102.

Scenario-2: Interface loopback 50 down on Leaf-101 (OSPF)

Figure 12-4 describes the same scenario from the OSPF perspective. As soon as the interface Loopback 50 state is changed from Up to Down, Leaf-101 generates a Link State Update (LSU) packet that carries a Router-LSA including information about current usable links. Since interface Loopback 50 is down, there are only two P2P (type-1) link descriptions describing the connection to Spine switches. When Leaf-102 receives the LSU it notifies that this is the newest Router-LSA. Leaf-102 then executes the full Shortest-Path Tree calculation even though the core of the tree has not changed.

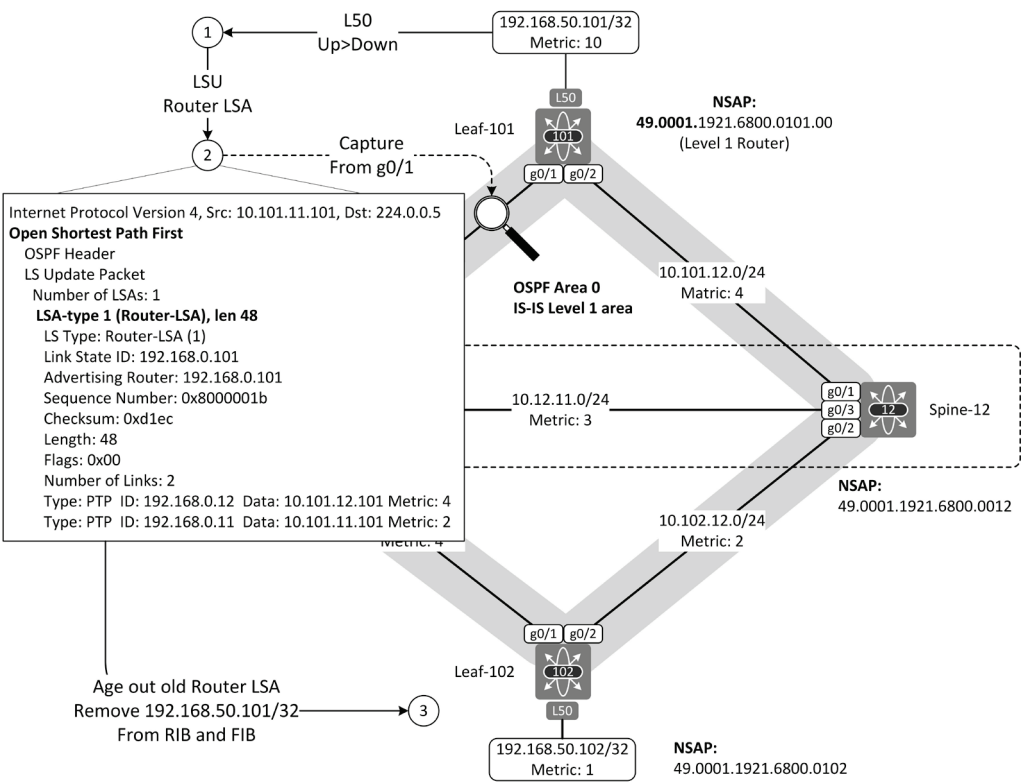


Figure 12-4: Interface Loopback 50 DOWN on Leaf-101 (OSPF).

The first highlighted (black and white) shows the start of the SPF algorithm. Leaf-102 re-builds the whole SPT by going through each router-LSA found on its LSDB. It moves itself as a root for the SPT. Then it finds out the closest OSPF peer which is Spine-12 and processes LSA originated by the Spine-12.

```

Leaf-102#debug ip ospf spf intra
OSPF SPF intra debugging is on
04:50:38: OSPF-1 SPF : Detect change in LSA type 1, LSID 192.168.0.101 from 192.168.0.101
area 0
04:50:43: OSPF-1 MON : Begin SPF at 17443.145ms, process time 19052ms
04:50:43: OSPF-1 INTRA: Running SPF for area 0, SPF-type Full
04:50:43: OSPF-1 INTRA: Initializing to run spf
04:50:43: OSPF-1 INTRA: spf intra() - rebuilding the tree
04:50:43: OSPF-1 INTRA: It is a router LSA 192.168.0.102. Link Count 2
04:50:43: OSPF-1 INTRA: Processing link 0, id 192.168.0.12, link data 10.102.12.102, type 1
04:50:43: OSPF-1 SPF : Add better path to LSA ID 192.168.0.12, gateway 10.102.12.12,
dist 2
04:50:43: OSPF-1 INTRA: Putting LSA on the clist LSID 192.168.0.12, Type 1, Adv Rtr.
192.168.0.12
04:50:43: OSPF-1 SPF : Add path: next-hop 10.102.12.12, interface GigabitEthernet0/2
04:50:43: OSPF-1 INTRA: Processing link 1, id 192.168.0.11, link data 10.102.11.102, type 1
04:50:43: OSPF-1 SPF : Add better path to LSA ID 192.168.0.11, gateway 10.102.11.11, dist
4
04:50:43: OSPF-1 INTRA: Putting LSA on the clist LSID 192.168.0.11, Type 1, Adv Rtr.
192.168.0.11
04:50:43: OSPF-1 INTRA: Upheap LSA ID 192.168.0.11, Type 1, Adv 192.168.0.11 on clist
from index 2 to 2
04:50:43: OSPF-1 SPF : Add path: next-hop 10.102.11.11, interface GigabitEthernet0/1
04:50:43: OSPF-1 INTRA: Downheap LSA ID 192.168.0.11, Type 1, Adv 192.168.0.11 on clist
from index 1 to 1
04:50:43: OSPF-1 INTRA: It is a router LSA 192.168.0.12. Link Count 3
04:50:43: OSPF-1 INTRA: Processing link 0, id 192.168.0.11, link data 10.12.11.12, type 1
04:50:43: OSPF-1 INTRA: Ignore newdist 5 olddist 4
04:50:43: OSPF-1 INTRA: Processing link 1, id 192.168.0.102, link data 10.102.12.12, type 1
04:50:43: OSPF-1 INTRA: Ignore newdist 4 olddist 0
04:50:43: OSPF-1 INTRA: Processing link 2, id 192.168.0.101, link data 10.101.12.12, type 1
04:50:43: OSPF-1 SPF : Add better path to LSA ID 192.168.0.101, gateway 10.101.12.101,
dist 6
04:50:43: OSPF-1 INTRA: Putting LSA on the clist LSID 192.168.0.101, Type 1, Adv Rtr.
192.168.0.101
04:50:43: OSPF-1 INTRA: Upheap LSA ID 192.168.0.101, Type 1, Adv 192.168.0.101 on clist
from index 2 to 2
04:50:43: OSPF-1 SPF : Add path: next-hop 10.102.12.12, interface GigabitEthernet0/2
04:50:43: OSPF-1 INTRA: Downheap LSA ID 192.168.0.101, Type 1, Adv 192.168.0.101 on clist
from index 1 to 1
04:50:43: OSPF-1 INTRA: It is a router LSA 192.168.0.11. Link Count 3
04:50:43: OSPF-1 INTRA: Processing link 0, id 192.168.0.12, link data 10.12.11.11, type 1
04:50:43: OSPF-1 INTRA: Ignore newdist 7 olddist 2
04:50:43: OSPF-1 INTRA: Processing link 1, id 192.168.0.102, link data 10.102.11.11, type 1
04:50:43: OSPF-1 INTRA: Ignore newdist 8 olddist 0
04:50:43: OSPF-1 INTRA: Processing link 2, id 192.168.0.101, link data 10.101.11.11, type 1
04:50:43: OSPF-1 INTRA: Add equal-length path to 192.168.0.101, dist 6
04:50:43: OSPF-1 INTRA: LSA already on the clist LSID 192.168.0.101, Type 1, Adv Rtr.
192.168.0.101
04:50:43: OSPF-1 SPF : Add path: next-hop 10.102.11.11, interface GigabitEthernet0/1
04:50:43: OSPF-1 INTRA: Downheap LSA ID 192.168.0.101, Type 1, Adv 192.168.0.101 on
clist from index 1 to 1
04:50:43: OSPF-1 INTRA: It is a router LSA 192.168.0.101. Link Count 2
04:50:43: OSPF-1 INTRA: Processing link 0, id 192.168.0.12, link data 10.101.12.101, type
1
04:50:43: OSPF-1 INTRA: Ignore newdist 10 olddist 2
04:50:43: OSPF-1 INTRA: Processing link 1, id 192.168.0.11, link data 10.101.11.101, type
1
04:50:43: OSPF-1 INTRA: Ignore newdist 8 olddist 4
04:50:43: OSPF-1 INTRA: Adding Stub nets
04:50:43: OSPF-1 INTRA: Entered intra-area route sync for area 0
04:50:43: OSPF-1 INTRA: Entered intra-area route sync for area 0
04:50:43: OSPF-1 MON : End SPF at 17443.159ms, Total elapsed time 14ms

```

Example 12-3: “debug ip ospf spf intra” on Leaf-102.

Scenario-3: OSPF Incremental SPF – L50 Down on Leaf-101 (Stub)

OSPF SPT calculation can be optimized by using incremental SPF. The configuration is simple; the command **ispf** is added under the OSPF process. Incremental SPF is activated and SPF is run right after the command.

Example 12-4 illustrates the reaction of Leaf-102 when the interface Loopback 50 goes down on Leaf-101 and the iSPF is enabled on Leaf-102. Instead of running a full SPF, Leaf-102 runs Incremental SPF because the only change compared to the old and new LSA originated by Leaf-101 concerns Stub Network.

```
Leaf-102#debug ip ospf spf intra
OSPF SPF intra debugging is on
06:13:33: OSPF-1 SPF : Detect change in LSA type 1, LSID 192.168.0.101 from
192.168.0.101 area 0
06:13:33: OSPF-1 INTRA: Insert LSA to New_LSA list type 1, LSID 192.168.0.101,
from 192.168.0.101 area 0
06:13:38: OSPF-1 MON : Begin SPF at 22418.763ms, process time 23574ms
06:13:38: OSPF-1 INTRA: Running SPF for area 0, SPF-type Incremental
06:13:38: OSPF-1 INTRA: Initializing to run spf
06:13:38: OSPF-1 INTRA: Running incremental SPF for area 0
06:13:38: OSPF-1 INTRA: iSPF: Processing node 1/192.168.0.101/192.168.0.101
from the New List
06:13:38: OSPF-1 INTRA: iSPF: Checking parents
06:13:38: OSPF-1 INTRA: iSPF: trying to find a link to parent
1/192.168.0.11/192.168.0.11 in the new LSA
06:13:38: OSPF-1 INTRA: iSPF: ...found
06:13:38: OSPF-1 INTRA: iSPF: trying to find a link to parent
1/192.168.0.12/192.168.0.12 in the new LSA
06:13:38: OSPF-1 INTRA: iSPF: ...found
06:13:38: OSPF-1 INTRA: iSPF: No change in parents
06:13:38: OSPF-1 INTRA: iSPF: Scanning new LSA of node
1/192.168.0.101/192.168.0.101
06:13:38: OSPF-1 INTRA: iSPF: node 1/192.168.0.101/192.168.0.101 is NOT a
parent of node 1/192.168.0.12/192.168.0.12
06:13:38: OSPF-1 INTRA: iSPF: node 1/192.168.0.12/192.168.0.12 is parent of
node 1/192.168.0.101/192.168.0.101
06:13:38: OSPF-1 INTRA: iSPF: node 1/192.168.0.101/192.168.0.101 is NOT a
parent of node 1/192.168.0.11/192.168.0.11
06:13:38: OSPF-1 INTRA: iSPF: node 1/192.168.0.11/192.168.0.11 is parent of
node 1/192.168.0.101/192.168.0.101
06:13:38: OSPF-1 INTRA: iSPF: init all stub routes on delete list of node
1/192.168.0.101/192.168.0.101
06:13:38: OSPF-1 INTRA: iSPF: initializing node 0/192.168.50.101/192.168.0.101
06:13:38: OSPF-1 INTRA: iSPF: process all stub-routes of node
1/192.168.0.101/192.168.0.101
06:13:38: OSPF-1 INTRA: iSPF: Checking lost links of node
1/192.168.0.101/192.168.0.101
06:13:38: OSPF-1 INTRA: iSPF: Re-attaching nodes on orphans
06:13:38: OSPF-1 INTRA: spf_intra() - rebuilding the tree
06:13:38: OSPF-1 INTRA: Adding Stub nets
06:13:38: OSPF-1 INTRA: Entered intra-area route sync for area 0
06:13:38: OSPF-1 INTRA: Entered intra-area route sync for area 0
06:13:38: OSPF-1 MON : End SPF at 22418.771ms, Total elapsed time 8ms
```

Example 12-4: “debug ip ospf spf intra” on Leaf-102.

Scenario-4: OSPF Incremental SPF – Interface g0/3 Down on Spine-12 (transit link does not participate in SPT)

Incremental SPF (iSPF) also works with the transit links that are not part of the core of the SPT. When the interface g0/3 in Spine-12 goes down, Spine-12 will originate a new LSU packet where it sends fresh link descriptions. Since the link does not belong to SPT from the Leaf-102 perspective, Leaf-102 does not calculate the whole tree, it just processes the router LSA received from Spine-11 and adjust the LSDB based on it.

```
Leaf-102#debug ip ospf spf intra
OSPF SPF intra debugging is on
06:41:37: OSPF-1 SPF : Detect change in LSA type 1, LSID 192.168.0.12 from
192.168.0.12 area 0
06:41:37: OSPF-1 INTRA: Insert LSA to New_LSA list type 1, LSID
192.168.0.12, from 192.168.0.12 area 0
06:41:42: OSPF-1 MON : Begin SPF at 24102.468ms, process time 25012ms
06:41:42: OSPF-1 INTRA: Running SPF for area 0, SPF-type Incremental
06:41:42: OSPF-1 INTRA: Initializing to run spf
06:41:42: OSPF-1 INTRA: Running incremental SPF for area 0
06:41:42: OSPF-1 INTRA: iSPF: Processing node 1/192.168.0.12/192.168.0.12
from the New List
06:41:42: OSPF-1 INTRA: iSPF: Checking parents
06:41:42: OSPF-1 INTRA: iSPF: trying to find a link to parent
1/192.168.0.102/192.168.0.102 in the new LSA
06:41:42: OSPF-1 INTRA: iSPF: ...found
06:41:42: OSPF-1 INTRA: iSPF: No change in parents
06:41:42: OSPF-1 INTRA: iSPF: Scanning new LSA of node
1/192.168.0.12/192.168.0.12
06:41:42: OSPF-1 INTRA: iSPF: node 1/192.168.0.12/192.168.0.12 is NOT a
parent of node 1/192.168.0.102/192.168.0.102
06:41:42: OSPF-1 INTRA: iSPF: node 1/192.168.0.102/192.168.0.102 is
parent of node 1/192.168.0.12/192.168.0.12
06:41:42: OSPF-1 INTRA: iSPF: node 1/192.168.0.102/192.168.0.102 is
parent of node 1/192.168.0.12/192.168.0.12
06:41:42: OSPF-1 INTRA: iSPF: node 1/192.168.0.12/192.168.0.12 is parent
of node 1/192.168.0.101/192.168.0.101
06:41:42: OSPF-1 INTRA: iSPF: init all stub routes on delete list of node
1/192.168.0.12/192.168.0.12
06:41:42: OSPF-1 INTRA: iSPF: process all stub-routes of node
1/192.168.0.12/192.168.0.12
06:41:42: OSPF-1 INTRA: iSPF: Checking lost links of node
1/192.168.0.12/192.168.0.12
06:41:42: OSPF-1 INTRA: iSPF: node 1/192.168.0.12/192.168.0.12 is NOT a
parent of node 1/192.168.0.11/192.168.0.11
06:41:42: OSPF-1 INTRA: iSPF: Re-attaching nodes on orphans
06:41:42: OSPF-1 INTRA: spf_intra() - rebuilding the tree
06:41:42: OSPF-1 INTRA: Adding Stub nets
06:41:42: OSPF-1 INTRA: Entered intra-area route sync for area 0
06:41:42: OSPF-1 INTRA: Entered intra-area route sync for area 0
06:41:42: OSPF-1 MON : End SPF at 24102.474ms, Total elapsed time 6ms
```

Example 12-5: “debug ip ospf spf intra” on Leaf-102.

Scenario-5: IS-IS SPF – Interface g0/3 Down on Spine-12 (Full SPF computation)

Example 12-6 explains how ISIS reacts when interface g0/3 on Spine-12 goes down. Leaf-101 runs a full SPF and re-builds an SPT. When comparing debug output received from IS-IS to OSPF, the IS-IS is a bit more descriptive. The Output clearly shows how the Leaf-102 installs itself from the TENT list into the PATHS list. Then it moves its own IS-IS peers to the TENT list. Then Leaf-102 removes the peer with the lowest metric from the TENT list and adds it to the PATHS list. As a next step, Spine-12 peer Leaf-101 is moved into the TENT list (which already includes Spine-11). Then, the metric of these two is compared and the Spine-11 is moved into the PATHS list based on the lowest metric value. The process moves on and it is stopped when the TENT list is empty.

```
08:45:33: ISIS-SPF: Compute L1 SPT
08:45:33: ISIS-Stats: Compute L1 SPT
08:45:33: ISIS-SPF: Starting level-1 SPF with 1 nodes into TENT
08:45:33: ISIS-SPF: Move 1921.6800.0102.00-00 to PATHS, metric 0
08:45:33: ISIS-SPF: Remove the node from the TENT list lsp(1921.6800.0102.00-00)(1):0xF584E10
08:45:33: ISIS-SPF: Attempt to add each adj of the node to tent via add lsp routes,
lsp(1921.6800.0102.00-00)(1), lsptype:0
08:45:33: ISIS-SPF: lsptype:0, current_lsp(1921.6800.0102.00-00)(1) current_lsp:0xF584E10,
lsp_fragment:0xF584E10 calling isis_walk_lsp
08:45:33: ISIS-SPF: Added neighbor lsp to the tent list, link_is_p2p:1
08:45:33: ISIS-SPF: considering adj to 1921.6800.0012 (GigabitEthernet0/2) metric 2, level
1, circuit 3, adj 1
08:45:33: ISIS-SPF: (accepted)
08:45:33: ISIS-Spf: Add 1921.6800.0012.00-00 to TENT, metric 2
08:45:33: ISIS-Spf: Next hop 1921.6800.0012 (GigabitEthernet0/2)
08:45:33: ISIS-SPF: Added neighbor lsp to the tent list, link_is_p2p:1
08:45:33: ISIS-SPF: considering adj to 1921.6800.0011 (GigabitEthernet0/1) metric 4, level
1, circuit 3, adj 1
08:45:33: ISIS-SPF: (accepted)
08:45:33: ISIS-Spf: Add 1921.6800.0011.00-00 to TENT, metric 4
08:45:33: ISIS-Spf: Next hop 1921.6800.0011 (GigabitEthernet0/1)
08:45:33: ISIS-SPF: Move 1921.6800.0012.00-00 to PATHS, metric 2
08:45:33: ISIS-SPF: Remove the node from the TENT list lsp(1921.6800.0012.00-00)(2):0xFE85C8
08:45:33: ISIS-SPF: Attempt to add each adj of the node to tent via add lsp routes,
lsp(1921.6800.0012.00-00)(2), lsptype:0
08:45:33: ISIS-SPF: lsptype:0, current_lsp(1921.6800.0012.00-00)(2) current_lsp:0xFE85C8,
lsp_fragment:0xFE85C8 calling isis_walk_lsp
08:45:33: ISIS-SPF: Failed to add neighbor lsp to the tent list, link_is_p2p:1
08:45:33: ISIS-SPF: Added neighbor lsp to the tent list, link_is_p2p:1
08:45:33: ISIS-Spf: Add 1921.6800.0101.00-00 to TENT, metric 6
08:45:33: ISIS-Spf: Next hop 1921.6800.0012 (GigabitEthernet0/2)
08:45:33: ISIS-SPF: Move 1921.6800.0011.00-00 to PATHS, metric 4
08:45:33: ISIS-SPF: Remove the node from the TENT list lsp(1921.6800.0011.00-00)(4):0xDABACA8
08:45:33: ISIS-SPF: Attempt to add each adj of the node to tent via add lsp routes,
lsp(1921.6800.0011.00-00)(4), lsptype:0
08:45:33: ISIS-SPF: lsptype:0, current_lsp(1921.6800.0011.00-00)(4) current_lsp:0xDABACA8,
lsp_fragment:0xDABACA8 calling isis_walk_lsp
08:45:33: ISIS-SPF: Added neighbor lsp to the tent list, link_is_p2p:1
08:45:33: ISIS-Spf: Add 1921.6800.0101.00-00 to TENT, metric 6
08:45:33: ISIS-Spf: Next hop 1921.6800.0011 (GigabitEthernet0/1)
08:45:33: ISIS-Spf: Next hop 1921.6800.0012 (GigabitEthernet0/2)
08:45:33: ISIS-SPF: Failed to add neighbor lsp to the tent list, link_is_p2p:1
08:45:33: ISIS-SPF: Failed to add neighbor lsp to the tent list, link_is_p2p:1
08:45:33: ISIS-SPF: Move 1921.6800.0101.00-00 to PATHS, metric 6
```



```

08:45:33: ISIS-SPF: Remove the node from the TENT list lsp(1921.6800.0101.00-
00)(3):0xDABB208
08:45:33: ISIS-SPF: Attempt to add each adj of the node to tent via add lsp routes,
lsp(1921.6800.0101.00-00)(3), lsptype:0
08:45:33: ISIS-SPF: lsptype:0, current_lsp(1921.6800.0101.00-00)(3) current_lsp:0xDABB208,
lsp_fragment:0xDABB208 calling isis_walk_lsp
08:45:33: ISIS-SPF: Failed to add neighbor lsp to the tent list, link_is_p2p:1
08:45:33: ISIS-SPF: Failed to add neighbor lsp to the tent list, link_is_p2p:1
08:45:33: ISIS-SPF: Aging L1 LSP 1 (1921.6800.0102.00-00), version 41
08:45:33: ISIS-SPF: Aging L1 LSP 2 (1921.6800.0012.00-00), version 43
08:45:33: ISIS-SPF: Aging L1 LSP 3 (1921.6800.0101.00-00), version 64
08:45:33: ISIS-SPF: Aging L1 LSP 4 (1921.6800.0011.00-00), version 44
08:45:33: ISIS-Stats: SPF only compute time 0.014
08:45:33: ISIS-Stats: IPv4 RIB only compute time 0.000
08:45:33: ISIS-Stats: Complete L1 SPT,
08:45:33: ISIS-Stats: Compute time 0.014/0.014, 4/0 nodes, 4/0 links, 0 suspends

```

Example 12-6: *debug isis sfp-events on Leaf-102.*

Scenario-6: IS-IS Incremental SPF – Interface g0/3 Down on Spine-12 (transit link does not participate in SPT)

It is also possible to optimize the SPF algorithm in IS-IS by using Incremental SPF. The configuration is simple; add the command `ispf level-1 20` under the IS-IS routing process (default value when beginning iSPF is 120 seconds). Unlike in OSPF, IS-IS configuration allows the administrator to define when the Incremental SPF is activated (when activated, SPF is done immediately).

Example 12-7 describes the SPF process in Leaf-102 when the incremental SPF is enabled and the interface g0/3 on Spine-12 state changes from up to down. Note that there is a new LSP packet received from both Spine-11 (Adding LSP: 4) and Spine-12 (Adding LSP: 2). For simplicity processing of LSA received from Spine-12 is highlighted with grey color.

```

08:51:41: ISIS-SPF: Compute L1 SPT
08:51:41: ISIS-Stats: Compute L1 SPT
08:51:41: ISIS-Stats:
08:51:41: ISIS-Stats: Starting incremental SPF for level-1
08:51:41: ISIS-SPF: I-SPF: Adding LSP: (2) to NewLSP, metric: 2
08:51:41: ISIS-SPF: I-SPF: Adding LSP: (2) to 1035B410, metric: 2. From 24FD63D
08:51:41: ISIS-SPF: I-SPF: Adding LSP: (4) to NewLSP, metric: 4
08:51:41: ISIS-SPF: I-SPF: Adding LSP: (4) to 1035B410, metric: 4. From 24FD63D
08:51:41: ISIS-SPF: I-SPF: Entering in read_lsp with node: (2)
08:51:41: ISIS-SPF: I-SPF: Delta distance to (1) is 4
08:51:41: ISIS-SPF: I-SPF: Delta distance to (3) is 0
08:51:41: ISIS-SPF: I-SPF: Searching (2) orphans
08:51:41: ISIS-SPF: L1 LSP 2 (1921.6800.0012.00-00) flagged for recalculation from 2500635
08:51:41: ISIS-SPF: I-SPF: Entering in read_lsp with node: (4)
08:51:41: ISIS-SPF: I-SPF: Delta distance to (3) is 0
08:51:41: ISIS-SPF: I-SPF: Delta distance to (1) is 8
08:51:41: ISIS-SPF: I-SPF: Delta distance to (2) is 5
08:51:41: ISIS-SPF: I-SPF: Searching (4) orphans
08:51:41: ISIS-SPF: L1 LSP 4 (1921.6800.0011.00-00) flagged for recalculation from 2500635
08:51:41: ISIS-SPF: I-SPF: Entering isis_ispf_reattach_node
08:51:41: ISIS-SPF: Aging L1 LSP 1 (1921.6800.0102.00-00), version 44
08:51:41: ISIS-SPF: Aging L1 LSP 2 (1921.6800.0012.00-00), version 46
08:51:41: ISIS-SPF: Aging L1 LSP 3 (1921.6800.0101.00-00), version 67
08:51:41: ISIS-SPF: Aging L1 LSP 4 (1921.6800.0011.00-00), version 47
08:51:41: ISIS-Stats: SPF only compute time 0.007
08:51:41: ISIS-Stats: IPv4 RIB only compute time 0.000
08:51:41: ISIS-Stats: Complete L1 SPT,
08:51:41: ISIS-Stats: Compute time 0.007/0.007, 0/0 nodes, 0/0 links, 0 suspends

```

Example 12-7: *“debug isis sfp-events” on Leaf-102.*

Conclusion

SPF algorithm on both OSPF and IS-IS is possible to optimize by using Incremental SPF, which means that when either Stub Network or non-SPT Transit Network goes down, routers do not have to run full SPF. Though when any link/network comes up, the full SPF is calculated by both protocols.

When discussing which Link-State Protocol is better for the Underlay Network from the VXLAN perspective, there is no clear answer. Some IS-IS properties like adjustable LSP lifetime, the new property requires only a new TLV (not totally new LSA type) and fewer attributes that have to match to form and keep an IS-IS adjacency up might make IS-IS a bit better than OSPF. The question is; are these properties relevant from the VXLAN fabric Underlay Network perspective, at the end of the day there are only a couple of loopback addresses (if unnumbered uplinks are used) per router that needs to be reachable throughout the infrastructure. Though it is possible that there are hundreds or even thousands of switches in VXLAN fabric. Both OSPF and IS-IS are suitable for Underlay Network (Authors opinion). Use the one that you know better.

References:

- [RFC 1195] R. Callon, "Use of OSI IS-IS for Routing in TCP/IP and Dual Environments", RFC 1195, December 1990.
- [RFC 2328] J. Moy, "OSPF Version 2", RFC 2328, April 1998.
- [RFC 4970] A. Lindem et al., "Extensions to OSPF for Advertising Optional Router Capabilities", RFC 4970, July 2007.
- [RFC 5301] D. McPherson et al., "Dynamic Hostname Exchange Mechanism for IS-IS", RFC 5301, October 2008.
- [RFC 5340] R. Coltun et al., "OSPF for IPv6", RFC 5340, July 2008.
- [RFC 6860] Y. Yang et al., "Hiding Transit-Only Networks in OSPF", RFC 6860, January 2013.
- [RFC 8365] A. Sajassi et al. "A Network Virtualization Overlay Solution Using Ethernet VPN (EVPN)", RFC 8363, March 2018.
- [ISIS-OSPF] M. Bhatia et al., "IS-IS and OSPF Difference Discussion", Draft-bhatia-manral-isis-ospf-01" January 2006.

Appendix A. Chapter 4 Device Configurations

A.1 Edge-1 Configuration

```
Edge-1#sh run
Building configuration...

Current configuration : 6327 bytes
!
! Last configuration change at 16:41:50 UTC Fri Jan 3 2020
!
version 16.6
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
no platform punt-keepalive disable-kernel-core
platform console serial
!
hostname Edge-1
!
boot-start-marker
boot-end-marker
!
!
vrf definition 100_White
 rd 1:100
 !
  address-family ipv4
   route-target export 1:100
   route-target import 1:100
  exit-address-family
!
vrf definition 200_Grey
 rd 1:200
 !
  address-family ipv4
   route-target export 1:200
   route-target import 1:200
  exit-address-family
!
vrf definition DEFAULT_VN
 rd 1:4097
```

```
!  
address-family ipv4  
  route-target export 1:4097  
  route-target import 1:4097  
exit-address-family  
!  
!  
redundancy  
bridge-domain 30  
bridge-domain 40  
bridge-domain 100  
!  
interface Loopback0  
  ip address 192.168.1.1 255.255.255.255  
  ip ospf network point-to-point  
  ip ospf 1 area 0  
!  
interface LISP0  
!  
interface LISP0.100  
!  
interface LISP0.200  
!  
interface GigabitEthernet1  
  ip address 10.1.11.1 255.255.255.0  
  ip lisp source-locator Loopback0  
  ip ospf network point-to-point  
  ip ospf 1 area 0  
  negotiation auto  
  no mop enabled  
  no mop sysid  
!  
interface GigabitEthernet2  
  no ip address  
  ip mtu 1500  
  negotiation auto  
  no mop enabled  
  no mop sysid  
  service instance 100 ethernet  
    encapsulation untagged  
    bridge-domain 100  
!  
interface GigabitEthernet3  
  no ip address  
  negotiation auto  
  no mop enabled  
  no mop sysid  
  service instance 200 ethernet  
    encapsulation untagged
```

```
    bridge-domain 200
!
interface BDI40
  mac-address 0001.0001.0001
  no ip address
  no ip redirects
  ip local-proxy-arp
  no mop enabled
  no mop sysid
!
interface BDI100
  mac-address 0001.0001.0001
  vrf forwarding 100_White
  ip address 172.16.100.1 255.255.255.0
  no ip redirects
  ip local-proxy-arp
  ip route-cache same-interface
  lisp mobility 172_16_100_0-100_White
  no mop enabled
  no mop sysid
!
interface BDI200
  mac-address 0001.0001.0001
  vrf forwarding 200_Grey
  ip address 172.16.200.1 255.255.255.0
  no ip redirects
  ip local-proxy-arp
  ip route-cache same-interface
  lisp mobility 172_16_200_0-200_Grey
  no mop enabled
  no mop sysid
!
router lisp
  locator-table default
  locator-set RLOC-SET1
    IPv4-interface Loopback0 priority 1 weight 1
  exit-locator-set
!
  locator default-set RLOC-SET1
service ipv4
  encapsulation vxlan
  itr map-resolver 192.168.111.111
  itr
  etr map-server 192.168.111.111 key nwktimes
  etr map-server 192.168.111.111 proxy-reply
  etr
  sgt
  exit-service-ipv4
!
```

```
service ethernet
  itr map-resolver 192.168.111.111
  itr
  etr map-server 192.168.111.111 key nwktimes
  etr map-server 192.168.111.111 proxy-reply
  etr
  exit-service-ethernet
!
instance-id 100
  remote-rloc-probe on-route-change
  dynamic-eid 172.16.100.0-100_White
  database-mapping 172.16.100.0/24 locator-set RLOC-SET1
  exit-dynamic-eid
!
service ipv4
  eid-table vrf 100_White
  map-cache 0.0.0.0/0 map-request
  exit-service-ipv4
!
exit-instance-id
!
instance-id 200
  remote-rloc-probe on-route-change
  dynamic-eid 172.16.200.0-200_Grey
  database-mapping 172.16.200.0/24 locator-set RLOC-SET1
  exit-dynamic-eid
!
service ipv4
  eid-table vrf 200_Grey
  map-cache 0.0.0.0/0 map-request
  exit-service-ipv4
!
exit-instance-id
!
exit-router-lisp
!
router ospf 1
  router-id 192.168.1.1
```

A.2 Edge-2 Configuration

```
Edge-2#sh run
!
hostname Edge-2
!
vrf definition 100_White
  rd 1:100
  !
  address-family ipv4
    route-target export 1:100
    route-target import 1:100
  exit-address-family
!
vrf definition 200_Grey
  rd 1:200
  !
  address-family ipv4
    route-target export 1:200
    route-target import 1:200
  exit-address-family
!
no ip domain lookup
!
!
interface Loopback0
  ip address 192.168.2.2 255.255.255.255
  ip ospf network point-to-point
  ip ospf 1 area 0
!
interface LISP0
!
interface LISP0.100
!
interface LISP0.200
!
interface GigabitEthernet1
  ip address 10.2.11.2 255.255.255.0
  ip lisp source-locator Loopback0
  ip ospf network point-to-point
  ip ospf 1 area 0
  negotiation auto
!
interface GigabitEthernet2
  no ip address
  negotiation auto
service instance 100 ethernet
  encapsulation untagged
  bridge-domain 100
!
```



```
interface GigabitEthernet3
  no ip address
  negotiation auto
  no mop enabled
  no mop sysid
  service instance 200 ethernet
    encapsulation untagged
    bridge-domain 200
!
interface BDI100
  mac-address 0001.0001.0001
  vrf forwarding 100_White
  ip address 172.16.100.1 255.255.255.0
  no ip redirects
  ip local-proxy-arp
  ip route-cache same-interface
  lisp mobility 172_16_100_0-100_White
!
interface BDI200
  mac-address 0001.0001.0001
  vrf forwarding 200_Grey
  ip address 172.16.200.1 255.255.255.0
  no ip redirects
  ip local-proxy-arp
  ip route-cache same-interface
  lisp mobility 172_16_200_0-200_Grey
!
router lisp
  locator-table default
  locator-set RLOC-SET1
    IPv4-interface Loopback0 priority 1 weight 1
  exit-locator-set
!
  locator default-set RLOC-SET1
  service ipv4
    encapsulation vxlan
    itr map-resolver 192.168.111.111
    itr
    etr map-server 192.168.111.111 key nwktimes
    etr map-server 192.168.111.111 proxy-reply
    etr
    exit-service-ipv4
!
  service ethernet
    itr map-resolver 192.168.111.111
    itr
    etr map-server 192.168.111.111 key nwktimes
    etr map-server 192.168.111.111 proxy-reply
    etr
```

```
    exit-service-ethernet
    !
instance-id 100
    remote-rloc-probe on-route-change
    dynamic-eid 172_16_100_0-100_White
        database-mapping 172.16.100.0/24 locator-set RLOC-SET1
    exit-dynamic-eid
    !
service ipv4
    eid-table vrf 100_White
    map-cache 0.0.0.0/0 map-request
    exit-service-ipv4
    !
exit-instance-id
!
instance-id 200
    remote-rloc-probe on-route-change
    dynamic-eid 172_16_200_0-200_Grey
        database-mapping 172.16.200.0/24 locator-set RLOC-SET1
    exit-dynamic-eid
    !
service ipv4
    eid-table vrf 200_Grey
    map-cache 0.0.0.0/0 map-request
    exit-service-ipv4
    !
exit-instance-id
!
exit-router-lisp
!
router ospf 1
    router-id 192.168.2.2
```

A.3 ControlPlane Configuration

```
CtrlPlane#sh run
!
hostname CtrlPlane
!
interface Loopback0
 ip address 192.168.111.111 255.255.255.255
 ip ospf network point-to-point
 ip ospf 1 area 0
!
interface GigabitEthernet1
 ip address 10.11.111.111 255.255.255.0
 ip lisp source-locator Loopback0
 ip ospf network point-to-point
 ip ospf 1 area 0
!
router lisp
 locator-table default
 service ipv4
  encapsulation vxlan
  map-server
  map-resolver
  exit-service-ipv4
!
 service ethernet
  map-server
  map-resolver
  exit-service-ethernet
!
 site Network-Times
  authentication-key nwktimes
  eid-record instance-id 100 172.16.100.0/24 accept-more-specifics
  eid-record instance-id 200 172.16.200.0/24 accept-more-specifics
  exit-site
!
 exit-router-lisp
!
router ospf 1
 router-id 192.168.111.111
```

A.4 ISP-Router Configuration

```
ISP-NORTH-R11#sh run
!
hostname ISP-NORTH-R11
!
interface Loopback0
 ip address 192.168.11.11 255.255.255.255
 ip ospf network point-to-point
 ip ospf 1 area 0
!
interface GigabitEthernet1
 ip address 10.1.11.11 255.255.255.0
 ip ospf network point-to-point
 ip ospf 1 area 0
 negotiation auto
 no mop enabled
 no mop sysid
!
interface GigabitEthernet2
 ip address 10.2.11.11 255.255.255.0
 ip ospf network point-to-point
 ip ospf 1 area 0
 negotiation auto
 no mop enabled
 no mop sysid
!
interface GigabitEthernet3
 ip address 10.11.111.11 255.255.255.0
 ip ospf network point-to-point
 ip ospf 1 area 0
 negotiation auto
router ospf 1
 router-id 192.168.11.11
```

Appendix B. Chapter 7-8 Device Configurations

B.1 Border-PxTR Configurations

```
Border-Shared#sh run
Building configuration...

Current configuration : 4276 bytes
!
! Last configuration change at 14:35:02 UTC Sat Jan 18 2020
!
version 16.6
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
no platform punt-keepalive disable-kernel-core
platform console serial
!
hostname Border-Shared
!
boot-start-marker
boot-end-marker
!
!
vrf definition 100_White
 rd 1:100
 !
  address-family ipv4
   route-target export 1:100
   route-target import 1:100
  exit-address-family
!
vrf definition 200_Grey
 rd 1:200
 !
  address-family ipv4
   route-target export 1:200
   route-target import 1:200
  exit-address-family
!
!
interface Loopback0
 ip address 192.168.222.222 255.255.255.255
 ip ospf network point-to-point
 ip ospf 1 area 0
!
```

```
interface LISP0
!
interface LISP0.100
!
interface LISP0.200
!
interface GigabitEthernet1
 ip address 10.222.11.222 255.255.255.0
 ip lisp source-locator Loopback0
 ip ospf network point-to-point
 ip ospf 1 area 0
 negotiation auto
 no mop enabled
 no mop sysid
!
interface GigabitEthernet2
 vrf forwarding 100 White
 ip address 10.222.22.222 255.255.255.0
 negotiation auto
 no mop enabled
 no mop sysid
!
interface GigabitEthernet3
 vrf forwarding 200 Grey
 ip address 10.222.33.222 255.255.255.0
 negotiation auto
 no mop enabled
 no mop sysid
!
interface GigabitEthernet4
 no ip address
 shutdown
 negotiation auto
 no mop enabled
 no mop sysid
!
router lisp
 security
 locator-table default
 locator-set RLOC-SET1
  IPv4-interface Loopback0 priority 1 weight 1
 exit-locator-set
!
 locator default-set RLOC-SET1
 service ipv4
  encapsulation vxlan
  itr map-resolver 192.168.111.111
  itr
  etr map-server 192.168.111.111 key nwktimes
  etr map-server 192.168.111.111 proxy-reply
  etr
  sgt
  distance site-registrations 250
 exit-service-ipv4
!
 service ethernet
```

```
itr map-resolver 192.168.111.111
itr
etr map-server 192.168.111.111 key nwktimes
etr map-server 192.168.111.111 proxy-reply
etr
exit-service-ethernet
!
instance-id 100
service ipv4
  eid-table vrf 100 White
  map-cache 0.0.0.0/0 map-request
  route-import database bgp 65000 locator-set RLOC-SET1
  distance site-registrations 250
  map-cache site-registration
  proxy-etr
  proxy-itr 192.168.222.222
  exit-service-ipv4
!
exit-instance-id
!
instance-id 200
service ipv4
  eid-table vrf 200 Grey
  map-cache 0.0.0.0/0 map-request
  route-import database bgp 65000 locator-set RLOC-SET1
  distance site-registrations 250
  map-cache site-registration
  proxy-etr
  proxy-itr 192.168.222.222
  exit-service-ipv4
!
exit-instance-id
!
exit-router-lisp
!
router ospf 1
  router-id 192.168.22.22
!
router bgp 65000
  no bgp log-neighbor-changes
  neighbor 10.222.22.22 remote-as 65022
  neighbor 10.222.33.33 remote-as 65022
  neighbor 192.168.111.111 remote-as 65000
  neighbor 192.168.111.111 update-source Loopback0
!
address-family ipv4
  no neighbor 10.222.22.22 activate
  neighbor 10.222.33.33 activate
  no neighbor 192.168.111.111 activate
exit-address-family
!
address-family vpnv4
  neighbor 192.168.111.111 activate
  neighbor 192.168.111.111 send-community extended
exit-address-family
!
```

```
address-family ipv4 vrf 100_White
  neighbor 10.222.22.22 remote-as 65022
  neighbor 10.222.22.22 activate
  neighbor 10.222.22.22 allowas-in
exit-address-family
!
address-family ipv4 vrf 200_Grey
  neighbor 10.222.33.33 remote-as 65022
  neighbor 10.222.33.33 activate
  neighbor 10.222.33.33 allowas-in
exit-address-family
!
```

B.2 Control Plane Configurations

```
CtrlPlane#sh run
Building configuration...

Current configuration : 6251 bytes
!
! Last configuration change at 08:03:25 UTC Sun Jan 19 2020
!
version 16.6
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
no platform punt-keepalive disable-kernel-core
platform console serial
!
hostname CtrlPlane
!
boot-start-marker
boot-end-marker
!
!
vrf definition 100_White
  rd 1:100
  !
  address-family ipv4
    route-target export 1:100
    route-target import 1:100
  exit-address-family
!
vrf definition 200_Grey
  rd 1:200
  !
  address-family ipv4
    route-target export 1:200
    route-target import 1:200
  exit-address-family
!
!
!
```



```
interface Loopback0
 ip address 192.168.111.111 255.255.255.255
 ip ospf network point-to-point
 ip ospf 1 area 0
!
interface GigabitEthernet1
 ip address 10.11.111.111 255.255.255.0
 ip lisp source-locator Loopback0
 ip ospf network point-to-point
 ip ospf 1 area 0
 negotiation auto
 no mop enabled
 no mop sysid
!
interface GigabitEthernet2
 no ip address
 shutdown
 negotiation auto
 no mop enabled
 no mop sysid
!
interface GigabitEthernet3
 no ip address
 shutdown
 negotiation auto
 no mop enabled
 no mop sysid
!
interface GigabitEthernet4
 no ip address
 shutdown
 negotiation auto
 no mop enabled
 no mop sysid
!
router lisp
 security
 locator-table default
 service ipv4
 encapsulation vxlan
 sgt
 map-server
 map-resolver
 exit-service-ipv4
!
 service ethernet
 map-server
 map-resolver
 exit-service-ethernet
!
instance-id 100
 service ipv4
 eid-table vrf 100_White
 route-export site-registrations
 distance site-registrations 250
 exit-service-ipv4
```

```
!
exit-instance-id
!
instance-id 200
service ipv4
  eid-table vrf 200_Grey
  route-export site-registrations
  distance site-registrations 250
  exit-service-ipv4
!
exit-instance-id
!
site Network-Times
authentication-key nwktimes
eid-record instance-id 100 172.16.22.0/24 accept-more-specifics
eid-record instance-id 100 172.16.100.0/24 accept-more-specifics
eid-record instance-id 100 172.16.200.0/24 accept-more-specifics
eid-record instance-id 200 172.16.22.0/24 accept-more-specifics
eid-record instance-id 200 172.16.100.0/24 accept-more-specifics
eid-record instance-id 200 172.16.200.0/24 accept-more-specifics
exit-site
!
exit-router-lisp
!
router lisp 100
  exit-router-lisp
!
router ospf 1
  router-id 192.168.111.111
!
router bgp 65000
  bgp log-neighbor-changes
  neighbor 192.168.222.222 remote-as 65000
  neighbor 192.168.222.222 update-source Loopback0
!
address-family ipv4
  no neighbor 192.168.222.222 activate
exit-address-family
!
address-family vpnv4
  neighbor 192.168.222.222 activate
  neighbor 192.168.222.222 send-community extended
exit-address-family
!
address-family ipv4 vrf 100_White
  aggregate-address 172.16.100.0 255.255.255.0 summary-only
  redistribute lisp metric 10 route-map LISP100-TO-100_White-RIB
exit-address-family
!
address-family ipv4 vrf 200_Grey
  aggregate-address 172.16.200.0 255.255.255.0 summary-only
  redistribute lisp metric 10 route-map LISP200-TO-200_Grey-RIB
exit-address-family
!
!
virtual-service csr_mgmt
```

```

!
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
!
!
!
ip prefix-list LISP100-TO-100_White-RIB seq 10 permit 172.16.100.0/24
ge 32
!
ip prefix-list LISP200-TO-200_Grey-RIB seq 10 permit 172.16.200.0/24
ge 32
!
!
route-map LISP200-TO-200_Grey-RIB permit 10
  match ip address prefix-list LISP200-TO-200_Grey-RIB
!
route-map LISP100-TO-100_White-RIB permit 10
  match ip address prefix-list LISP100-TO-100_White-RIB
!

```

B.3 Edge-1 Configurations

```

Edge-1#sh run
Building configuration...

Current configuration : 6372 bytes
!
! Last configuration change at 12:31:48 UTC Sat Jan 18 2020
!
version 16.6
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
no platform punt-keepalive disable-kernel-core
platform console serial
!
hostname Edge-1
!
boot-start-marker
boot-end-marker
!
!
vrf definition 100_White
  rd 1:100
  !
  address-family ipv4
    route-target export 1:100
    route-target import 1:100
  exit-address-family
!
vrf definition 200_Grey
  rd 1:200
  !
  address-family ipv4

```

```
    route-target export 1:200
    route-target import 1:200
exit-address-family
!
vrf definition DEFAULT_VN
  rd 1:4097
  !
  address-family ipv4
    route-target export 1:4097
    route-target import 1:4097
  exit-address-family
!
!
!
interface Loopback0
  ip address 192.168.1.1 255.255.255.255
  ip ospf network point-to-point
  ip ospf 1 area 0
!
interface LISP0
!
interface LISP0.100
!
interface LISP0.200
!
interface GigabitEthernet1
  ip address 10.1.11.1 255.255.255.0
  ip lisp source-locator Loopback0
  ip ospf network point-to-point
  ip ospf 1 area 0
  negotiation auto
  no mop enabled
  no mop sysid
!
interface GigabitEthernet2
  no ip address
  ip mtu 1500
  negotiation auto
  cts manual
  policy static sgt 100 trusted
  cts role-based sgt-map sgt 100
  no mop enabled
  no mop sysid
  service instance 100 ethernet
    encapsulation untagged
    bridge-domain 100
  !
!
interface GigabitEthernet3
  no ip address
  negotiation auto
  no mop enabled
  no mop sysid
  service instance 200 ethernet
    encapsulation untagged
    bridge-domain 200
```

```

!
!
interface GigabitEthernet4
 no ip address
 negotiation auto
 no mop enabled
 no mop sysid
!
interface BDI40
 mac-address 0001.0001.0001
 no ip address
 no ip redirects
 ip local-proxy-arp
 no mop enabled
 no mop sysid
!
interface BDI100
 mac-address 0001.0001.0001
 vrf forwarding 100_White
 ip address 172.16.100.1 255.255.255.0
 no ip redirects
 ip local-proxy-arp
 ip route-cache same-interface
 lisp mobility 172_16_100_0-100_White
 no mop enabled
 no mop sysid
!
interface BDI200
 mac-address 0001.0001.0001
 vrf forwarding 200_Grey
 ip address 172.16.200.1 255.255.255.0
 no ip redirects
 ip local-proxy-arp
 ip route-cache same-interface
 lisp mobility 172_16_200_0-200_Grey
 no mop enabled
 no mop sysid
!
router lisp
 locator-table default
 locator-set RLOC-SET1
  IPv4-interface Loopback0 priority 1 weight 1
  exit-locator-set
!
 locator default-set RLOC-SET1
 service ipv4
  encapsulation vxlan
  itr map-resolver 192.168.111.111
  itr
  etr map-server 192.168.111.111 key nwktimes
  etr map-server 192.168.111.111 proxy-reply
  etr
  sgt
  exit-service-ipv4
!
 service ethernet

```

```
itr map-resolver 192.168.111.111
itr
etr map-server 192.168.111.111 key nwktimes
etr map-server 192.168.111.111 proxy-reply
etr
exit-service-ethernet
!
instance-id 100
remote-rloc-probe on-route-change
dynamic-eid 172_16_100_0-100_White
database-mapping 172.16.100.0/24 locator-set RLOC-SET1
exit-dynamic-eid
!
service ipv4
eid-table vrf 100_White
map-cache 0.0.0.0/0 map-request
use-petr 192.168.222.222
no proxy-itr
exit-service-ipv4
!
exit-instance-id
!
instance-id 200
remote-rloc-probe on-route-change
dynamic-eid 172_16_200_0-200_Grey
database-mapping 172.16.200.0/24 locator-set RLOC-SET1
exit-dynamic-eid
!
service ipv4
eid-table vrf 200_Grey
map-cache 0.0.0.0/0 map-request
exit-service-ipv4
!
exit-instance-id
!
exit-router-lisp
!
router ospf 1
router-id 192.168.1.1
!
```

B.4 Edge-2 Configurations

```
Edge-2#sh run
Building configuration...

Current configuration : 6250 bytes
!
! Last configuration change at 14:14:10 UTC Sat Jan 18 2020
!
version 16.6
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
no platform punt-keepalive disable-kernel-core
platform console serial
```

```
!  
hostname Edge-2  
!  
boot-start-marker  
boot-end-marker  
!  
!  
vrf definition 100_White  
  rd 1:100  
  !  
  address-family ipv4  
    route-target export 1:100  
    route-target import 1:100  
  exit-address-family  
!  
vrf definition 200_Grey  
  rd 1:200  
  !  
  address-family ipv4  
    route-target export 1:200  
    route-target import 1:200  
  exit-address-family  
!  
vrf definition DEFAULT_VN  
  rd 1:4097  
  !  
  address-family ipv4  
    route-target export 1:4097  
    route-target import 1:4097  
  exit-address-family  
!  
!  
!  
!  
interface Loopback0  
  ip address 192.168.2.2 255.255.255.255  
  ip ospf network point-to-point  
  ip ospf 1 area 0  
!  
interface LISP0  
!  
interface LISP0.100  
!  
interface LISP0.200  
!  
interface GigabitEthernet1  
  ip address 10.2.11.2 255.255.255.0  
  ip lisp source-locator Loopback0  
  ip ospf network point-to-point  
  ip ospf prefix-suppression  
  ip ospf 1 area 0  
  negotiation auto  
  no mop enabled  
  no mop sysid  
!  
interface GigabitEthernet2
```

```
no ip address
negotiation auto
no mop enabled
no mop sysid
service instance 100 ethernet
  encapsulation untagged
  bridge-domain 100
!
!
interface GigabitEthernet3
no ip address
negotiation auto
no mop enabled
no mop sysid
service instance 200 ethernet
  encapsulation untagged
  bridge-domain 200
!
!
interface GigabitEthernet4
no ip address
negotiation auto
no mop enabled
no mop sysid
service instance 30 ethernet
  encapsulation untagged
  bridge-domain 30
!
!
interface BDI100
mac-address 0001.0001.0001
vrf forwarding 100_White
ip address 172.16.100.1 255.255.255.0
no ip redirects
ip local-proxy-arp
ip route-cache same-interface
lisp mobility 172_16_100_0-100_White
no mop enabled
no mop sysid
!
interface BDI200
mac-address 0001.0001.0001
vrf forwarding 200_Grey
ip address 172.16.200.1 255.255.255.0
no ip redirects
ip local-proxy-arp
ip route-cache same-interface
lisp mobility 172_16_200_0-200_Grey
no mop enabled
no mop sysid
!
router lisp
locator-table default
locator-set RLOC-SET1
  IPv4-interface Loopback0 priority 1 weight 1
exit-locator-set
```



```

!
locator default-set RLOC-SET1
service ipv4
  encapsulation vxlan
  itr map-resolver 192.168.111.111
  itr
  etr map-server 192.168.111.111 key nwktimes
  etr map-server 192.168.111.111 proxy-reply
  etr
  sgt
  exit-service-ipv4
!
service ethernet
  itr map-resolver 192.168.111.111
  itr
  etr map-server 192.168.111.111 key nwktimes
  etr map-server 192.168.111.111 proxy-reply
  etr
  exit-service-ethernet
!
instance-id 100
  remote-rloc-probe on-route-change
  dynamic-eid 172_16_100_0-100_White
  database-mapping 172.16.100.0/24 locator-set RLOC-SET1
  exit-dynamic-eid
!
service ipv4
  eid-table vrf 100_White
  map-cache 0.0.0.0/0 map-request
  exit-service-ipv4
!
exit-instance-id
!
instance-id 200
  remote-rloc-probe on-route-change
  dynamic-eid 172_16_200_0-200_Grey
  database-mapping 172.16.200.0/24 locator-set RLOC-SET1
  exit-dynamic-eid
!
service ipv4
  eid-table vrf 200_Grey
  map-cache 0.0.0.0/0 map-request
  use-petr 192.168.222.222
  no proxy-itr
  exit-service-ipv4
!
exit-instance-id
!
exit-router-lisp
!
router ospf 1
router-id 192.168.2.2

```

```
Shared-Ro22#sh run
```

```
Building configuration...
```

```
Current configuration : 2021 bytes
```

```
!
```

```
! Last configuration change at 13:07:16 UTC Sat Jan 18 2020
```

```
!
```

```
version 16.6
```

```
service timestamps debug datetime msec
```

```
service timestamps log datetime msec
```

```
platform qfp utilization monitor load 80
```

```
no platform punt-keepalive disable-kernel-core
```

```
platform console serial
```

```
!
```

```
hostname Shared-Ro22
```

```
!
```

```
boot-start-marker
```

```
boot-end-marker
```

```
!
```

```
!
```

```
interface Loopback0
```

```
ip address 192.168.22.22 255.255.255.255
```

```
!
```

```
interface GigabitEthernet1
```

```
ip address 10.222.22.22 255.255.255.0
```

```
negotiation auto
```

```
no mop enabled
```

```
no mop sysid
```

```
!
```

```
interface GigabitEthernet2
```

```
ip address 172.16.22.1 255.255.255.0
```

```
negotiation auto
```

```
no mop enabled
```

```
no mop sysid
```

```
!
```

```
interface GigabitEthernet3
```

```
ip address 10.222.33.33 255.255.255.0
```

```
negotiation auto
```

```
no mop enabled
```

```
no mop sysid
```

```
!
```

```
interface GigabitEthernet4
```

```
no ip address
```

```
shutdown
```

```
negotiation auto
```

```
no mop enabled
```

```
no mop sysid
```

```
!
```

```
router bgp 65022
```

```
bgp log-neighbor-changes
```

```
neighbor 10.222.22.222 remote-as 65000
```

```
neighbor 10.222.33.222 remote-as 65000
```

```
!
```

```
address-family ipv4
```

```
network 172.16.22.0 mask 255.255.255.0
```

```
neighbor 10.222.22.222 activate
```

```
neighbor 10.222.22.222 send-community both
```

```
neighbor 10.222.33.222 activate
neighbor 10.222.33.222 send-community both
exit-address-family
!
Shared-Ro22#
```

Appendix C. Chapter 10 Device Configurations

C.1 xTR14 Configurations

```
hostname xTR14
!
!
vrf definition 100_White
  rd 1:100
  !
  address-family ipv4
    route-target export 1:100
    route-target import 1:100
  exit-address-family
!
bridge-domain 100
!
interface Loopback0
  ip address 192.168.14.14 255.255.255.255
  ip ospf network point-to-point
  ip ospf 1 area 0
!
interface LISP0
!
interface LISP0.100
!
interface GigabitEthernet1
  ip address 10.15.14.14 255.255.255.0
  ip lisp source-locator Loopback0
  ip ospf network point-to-point
  ip ospf 1 area 0
!
interface GigabitEthernet2
  no ip address
  ip mtu 1500
  cts manual
  policy static sgt 4
  cts role-based sgt-map sgt 100
service instance 100 ethernet
  encapsulation untagged
  bridge-domain 100
!
```

```

!
!
interface BDI100
 mac-address 0001.0001.0001
 vrf forwarding 100_White
 ip address 172.16.100.1 255.255.255.0
 no ip redirects
 ip local-proxy-arp
 ip pim sparse-mode
 ip route-cache same-interface
 lisp mobility 172_16_100_0-100_White
!
router lisp
 locator-table default
 locator-set RLOC-SET1
   IPv4-interface Loopback0 priority 1 weight 1
 exit-locator-set
!
locator default-set RLOC-SET1
service ipv4
 encapsulation vxlan
 itr map-resolver 192.168.11.11
 itr
 etr map-server 192.168.11.11 key nwktimes
 etr map-server 192.168.11.11 proxy-reply
 etr
 sgt
 use-petr 192.168.12.12
 exit-service-ipv4
!
service ethernet
 itr map-resolver 192.168.11.11
 itr
 etr map-server 192.168.11.11 key nwktimes
 etr map-server 192.168.11.11 proxy-reply
 etr
 exit-service-ethernet
!
instance-id 100
 remote-rloc-probe on-route-change
 dynamic-eid 172_16_100_0-100_White
   database-mapping 172.16.100.0/24 locator-set RLOC-SET1
 exit-dynamic-eid
!
service ipv4
 eid-table vrf 100_White
 map-cache 0.0.0.0/0 map-request
 use-petr 192.168.12.12
 no proxy-itr

```

```
    exit-service-ipv4
    !
    exit-instance-id
    !
    exit-router-lisp
    !
router ospf 1
    router-id 192.168.14.14
    !
```

C.2 xTR24 Configurations

```
xTR24#sh run
hostname xTR24
!
vrf definition 100_White
    rd 1:100
    !
    address-family ipv4
        route-target export 1:100
        route-target import 1:100
    exit-address-family
    !
interface Loopback0
    ip address 192.168.24.24 255.255.255.255
    ip ospf network point-to-point
    ip ospf 1 area 0
    !
interface LISP0
    !
interface LISP0.100
    !
interface GigabitEthernet1
    ip address 10.25.24.24 255.255.255.0
    ip lisp source-locator Loopback0
    ip ospf network point-to-point
    ip ospf 1 area 0
    !
interface GigabitEthernet2
    no ip address
    ip mtu 1500
    cts manual
    policy static sgt 4
    cts role-based sgt-map sgt 100
    service instance 100 ethernet
        encapsulation untagged
        bridge-domain 100
    !
    !
interface BDI100
```

```
mac-address 0001.0001.0001
vrf forwarding 100_White
ip address 172.16.101.1 255.255.255.0
no ip redirects
ip local-proxy-arp
ip pim sparse-mode
ip route-cache same-interface
lisp mobility 172_16_101_0-101_White
!
router lisp
 locator-table default
 locator-set RLOC-SET1
   IPv4-interface Loopback0 priority 1 weight 1
 exit-locator-set
!
 locator default-set RLOC-SET1
service ipv4
 encapsulation vxlan
 itr map-resolver 192.168.21.21
 itr
 etr map-server 192.168.21.21 key nwktimes
 etr map-server 192.168.21.21 proxy-reply
 etr
 sgt
 use-petr 192.168.22.22
 exit-service-ipv4
!
service ethernet
 itr map-resolver 192.168.21.21
 itr
 etr map-server 192.168.21.21 key nwktimes
 etr map-server 192.168.21.21 proxy-reply
 etr
 exit-service-ethernet
!
instance-id 100
 remote-rloc-probe on-route-change
 dynamic-eid 172_16_101_0-101_White
   database-mapping 172.16.101.0/24 locator-set RLOC-SET1
 exit-dynamic-eid
!
service ipv4
 eid-table vrf 100_White
 map-cache 0.0.0.0/0 map-request
 use-petr 192.168.22.22
 no proxy-itr
 exit-service-ipv4
!
exit-instance-id
!
```

```
exit-router-lisp
!
router ospf 1
  router-id 192.168.24.24
```

C.3 Co15 Configurations

```
Co15#sh run
!
hostname Co15
!
!
interface Loopback0
  ip address 192.168.15.15 255.255.255.255
  ip ospf network point-to-point
  ip ospf 1 area 0
!
interface GigabitEthernet1
  ip address 10.15.11.15 255.255.255.0
  ip ospf network point-to-point
  ip ospf 1 area 0
!
interface GigabitEthernet2
  ip address 10.15.12.15 255.255.255.0
  ip ospf network point-to-point
  ip ospf 1 area 0
!
interface GigabitEthernet3
  ip address 10.15.13.15 255.255.255.0
  ip ospf network point-to-point
  ip ospf 1 area 0
  negotiation auto
  no mop enabled
  no mop sysid
!
interface GigabitEthernet4
  ip address 10.15.14.15 255.255.255.0
  ip ospf network point-to-point
  ip ospf 1 area 0
!
router ospf 1
  router-id 192.16.15.15
!
```

C.4 Co25 Configurations

```
Co25#sh run
!
hostname Co25
!
interface Loopback0
```



```
ip address 192.168.25.25 255.255.255.255
ip ospf network point-to-point
ip ospf 1 area 0
!
interface GigabitEthernet1
ip address 10.25.21.25 255.255.255.0
ip ospf network point-to-point
ip ospf 1 area 0
!
interface GigabitEthernet2
ip address 10.25.22.25 255.255.255.0
ip ospf network point-to-point
ip ospf 1 area 0
!
interface GigabitEthernet3
ip address 10.25.23.25 255.255.255.0
ip ospf network point-to-point
ip ospf 1 area 0
!
interface GigabitEthernet4
ip address 10.25.24.25 255.255.255.0
ip ospf network point-to-point
ip ospf 1 area 0
!
router ospf 1
router-id 192.16.25.25
```

C.5 CP11 Configurations

```
CP11#sh run
!
hostname CP11
!
vrf definition 100_White
rd 1:100
!
address-family ipv4
route-target export 1:100
route-target import 1:100
exit-address-family
!
!
interface Loopback0
ip address 192.168.11.11 255.255.255.255
ip ospf network point-to-point
ip ospf 1 area 0
!
interface GigabitEthernet1
ip address 10.15.11.11 255.255.255.0
ip lisp source-locator Loopback0
ip ospf network point-to-point
ip ospf 1 area 0
!
```

```
router lisp
 security
 locator-table default
 service ipv4
 encapsulation vxlan
 sgt
 map-server
 map-resolver
 exit-service-ipv4
 !
 service ethernet
 map-server
 map-resolver
 exit-service-ethernet
 !
 instance-id 100
 service ipv4
 eid-table vrf 100_White
 route-export site-registrations
 distance site-registrations 250
 exit-service-ipv4
 !
 exit-instance-id
 !
 site Network-Times
 authentication-key nwktimes
 eid-record instance-id 100 172.16.100.0/24 accept-more-specifics
 eid-record instance-id 100 172.16.101.0/24 accept-more-specifics
 exit-site
 !
 exit-router-lisp
 !
router lisp 100
 exit-router-lisp
 !
router ospf 1
 router-id 192.168.11.11
 !
router bgp 65000
 bgp log-neighbor-changes
 neighbor 192.168.12.12 remote-as 65000
 neighbor 192.168.12.12 update-source Loopback0
 neighbor 192.168.13.13 remote-as 65000
 neighbor 192.168.13.13 update-source Loopback0
 !
 address-family ipv4
 neighbor 192.168.12.12 activate
 neighbor 192.168.13.13 activate
 exit-address-family
 !
 address-family vpnv4
 neighbor 192.168.12.12 activate
 neighbor 192.168.12.12 send-community extended
 neighbor 192.168.13.13 activate
 neighbor 192.168.13.13 send-community extended
 exit-address-family
```

```

!
address-family ipv4 vrf 100_White
  aggregate-address 172.16.100.0 255.255.255.0 summary-only
  redistribute lisp metric 10 route-map LISP100-TO-100_White-RIB
exit-address-family
!
ip prefix-list LISP100-TO-100_White-RIB seq 10 permit 172.16.100.0/24
ge 32
!
route-map LISP100-TO-100_White-RIB permit 10
  match ip address prefix-list LISP100-TO-100_White-RIB

```

C.6 CP21 Configurations

```

CP21#sh run
platform console serial
!
hostname CP21
!
!
vrf definition 100_White
  rd 1:100
  !
  address-family ipv4
    route-target export 1:100
    route-target import 1:100
  exit-address-family
!
!
interface Loopback0
  ip address 192.168.21.21 255.255.255.255
  ip ospf network point-to-point
  ip ospf 1 area 0
!
interface GigabitEthernet1
  ip address 10.25.21.21 255.255.255.0
  ip lisp source-locator Loopback0
  ip ospf network point-to-point
  ip ospf 1 area 0
!
!
router lisp
  security
  locator-table default
  service ipv4
    encapsulation vxlan
    sgt
    map-server
    map-resolver
  exit-service-ipv4
!
service ethernet
  map-server
  map-resolver
  exit-service-ethernet

```

```

!
instance-id 100
  service ipv4
    eid-table vrf 100_White
    route-export site-registrations
    distance site-registrations 250
    exit-service-ipv4
  !
  exit-instance-id
!
site Network-Times-2
  authentication-key nwktimes
  eid-record instance-id 100 172.16.100.0/24 accept-more-specifics
  eid-record instance-id 100 172.16.101.0/24 accept-more-specifics
  exit-site
!
exit-router-lisp
!
router ospf 1
  router-id 192.168.21.21
!
router bgp 65001
  bgp log-neighbor-changes
  neighbor 192.168.22.22 remote-as 65001
  neighbor 192.168.22.22 update-source Loopback0
  neighbor 192.168.23.23 remote-as 65001
  neighbor 192.168.23.23 update-source Loopback0
!
address-family ipv4
  neighbor 192.168.22.22 activate
  neighbor 192.168.23.23 activate
exit-address-family
!
address-family vpnv4
  neighbor 192.168.22.22 activate
  neighbor 192.168.22.22 send-community extended
  neighbor 192.168.23.23 activate
  neighbor 192.168.23.23 send-community extended
exit-address-family
!
address-family ipv4 vrf 100_White
  aggregate-address 172.16.101.0 255.255.255.0 summary-only
  redistribute lisp metric 10 route-map LISP100-TO-100_White-RIB
exit-address-family
!
!
ip prefix-list LISP100-TO-100_White-RIB seq 10 permit 172.16.101.0/24
ge 32
!
!
route-map LISP100-TO-100_White-RIB permit 10
  match ip address prefix-list LISP100-TO-100_White-RIB

```

C.7 Transit-PxTR13 Configurations

```
Transit-PxTR13#sh run
```

```
!  
hostname Transit-PxTR13  
!  
vrf definition 100_White  
  rd 1:100  
  !  
  address-family ipv4  
    route-target export 1:100  
    route-target import 1:100  
  exit-address-family  
!  
!  
interface Loopback0  
  ip address 192.168.13.13 255.255.255.255  
  ip ospf network point-to-point  
  ip ospf 1 area 0  
!  
interface LISP0  
  ip pim sparse-mode  
!  
interface LISP0.100  
  ip pim sparse-mode  
!  
interface GigabitEthernet1  
  ip address 10.15.13.13 255.255.255.0  
  ip lisp source-locator Loopback0  
  ip ospf network point-to-point  
  ip ospf 1 area 0  
!  
interface GigabitEthernet2  
  vrf forwarding 100_White  
  ip address 10.13.101.13 255.255.255.0  
!  
router lisp  
  security  
  locator-table default  
  locator-set RLOC-SET1  
    IPv4-interface Loopback0 priority 3 weight 1  
  exit-locator-set  
!  
  locator default-set RLOC-SET1  
service ipv4  
  encapsulation vxlan  
  itr map-resolver 192.168.11.11  
  itr  
  etr map-server 192.168.11.11 key nwktimes  
  etr map-server 192.168.11.11 proxy-reply  
  etr  
  sgt  
  distance site-registrations 250  
  exit-service-ipv4  
!  
service ethernet  
  itr map-resolver 192.168.11.11  
  itr  
  etr map-server 192.168.11.11 key nwktimes
```

```
etr map-server 192.168.11.11 proxy-reply
etr
exit-service-ethernet
!
instance-id 100
service ipv4
  eid-table vrf 100_White
  map-cache 0.0.0.0/0 map-request
  route-import database bgp 65000 locator-set RLOC-SET1
  distance site-registrations 250
  map-cache site-registration
  proxy-etr
  proxy-itr 192.168.13.13
  exit-service-ipv4
!
exit-instance-id
!
exit-router-lisp
!
router ospf 1
  router-id 192.168.13.13
!
router bgp 65000
  no bgp log-neighbor-changes
  neighbor 10.13.101.101 remote-as 65101
  neighbor 192.168.11.11 remote-as 65000
  neighbor 192.168.11.11 update-source Loopback0
!
address-family ipv4
  neighbor 10.13.101.101 activate
  neighbor 192.168.11.11 activate
exit-address-family
!
address-family vpnv4
  neighbor 192.168.11.11 activate
  neighbor 192.168.11.11 send-community extended
exit-address-family
!
address-family ipv4 vrf 100_White
  neighbor 10.13.101.101 remote-as 65101
  neighbor 10.13.101.101 activate
exit-address-family
```

C.8 Transit-PxTR23 Configurations

```
Transit-PxTR23#sh run

hostname Transit-PxTR23
!
!
vrf definition 100_White
  rd 1:100
!
address-family ipv4
  route-target export 1:100
  route-target import 1:100
```

```
exit-address-family
!
!
interface Loopback0
ip address 192.168.23.23 255.255.255.255
ip ospf network point-to-point
ip ospf 1 area 0
!
interface LISP0
!
interface LISP0.100
!
interface GigabitEthernet1
ip address 10.25.23.23 255.255.255.0
ip lisp source-locator Loopback0
ip ospf network point-to-point
ip ospf 1 area 0
!
interface GigabitEthernet2
vrf forwarding 100_White
ip address 10.23.101.23 255.255.255.0
!
router lisp
security
locator-table default
locator-set RLOC-SET1
  IPv4-interface Loopback0 priority 3 weight 1
  exit-locator-set
!
locator default-set RLOC-SET1
service ipv4
  encapsulation vxlan
  itr map-resolver 192.168.21.21
  itr
  etr map-server 192.168.21.21 key nwktimes
  etr map-server 192.168.21.21 proxy-reply
  etr
  sgt
  distance site-registrations 250
  exit-service-ipv4
!
service ethernet
  itr map-resolver 192.168.21.21
  itr
  etr map-server 192.168.21.21 key nwktimes
  etr map-server 192.168.21.21 proxy-reply
  etr
  exit-service-ethernet
!
instance-id 100
service ipv4
  eid-table vrf 100_White
  map-cache 0.0.0.0/0 map-request
  route-import database bgp 65001 locator-set RLOC-SET1
  distance site-registrations 250
  map-cache site-registration
```

```
    proxy-etr
    proxy-itr 192.168.23.23
    exit-service-ipv4
    !
    exit-instance-id
    !
    exit-router-lisp
    !
router ospf 1
    router-id 192.168.23.23
    !
router bgp 65001
    no bgp log-neighbor-changes
    neighbor 10.23.101.101 remote-as 65101
    neighbor 192.168.21.21 remote-as 65001
    neighbor 192.168.21.21 update-source Loopback0
    !
    address-family ipv4
        neighbor 10.23.101.101 activate
        neighbor 192.168.21.21 activate
    exit-address-family
    !
    address-family vpnv4
        neighbor 192.168.21.21 activate
        neighbor 192.168.21.21 send-community extended
    exit-address-family
    !
    address-family ipv4 vrf 100_White
        neighbor 10.23.101.101 remote-as 65101
        neighbor 10.23.101.101 activate
    exit-address-family
```

C.9 Inet-PxTR12 Configurations

```
Inet-PxTR12#sh run
console serial
!
hostname Inet-PxTR12
!
!
vrf definition 100_White
    rd 1:100
    !
    address-family ipv4
        route-target export 1:100
        route-target import 1:100
    exit-address-family
    !
    !
    !
interface Loopback0
    ip address 192.168.12.12 255.255.255.255
    ip ospf network point-to-point
    ip ospf 1 area 0
    !
```



```

interface LISP0
!
interface LISP0.100
!
interface GigabitEthernet1
ip address 10.15.12.12 255.255.255.0
ip ospf network point-to-point
ip ospf 1 area 0
!
interface GigabitEthernet2
vrf forwarding 100_White
ip address 10.12.201.12 255.255.255.0
!
router lisp
security
locator-table default
locator-set RLOC-SET1
IPv4-interface Loopback0 priority 3 weight 1
exit-locator-set
!
locator default-set RLOC-SET1
service ipv4
encapsulation vxlan
itr map-resolver 192.168.11.11
itr
etr map-server 192.168.11.11 key nwktimes
etr map-server 192.168.11.11 proxy-reply
etr
sgt
distance site-registrations 250
exit-service-ipv4
!
service ethernet
itr map-resolver 192.168.11.11
itr
etr map-server 192.168.11.11 key nwktimes
etr map-server 192.168.11.11 proxy-reply
etr
exit-service-ethernet
!
instance-id 100
service ipv4
eid-table vrf 100_White
map-cache 0.0.0.0/0 map-request
route-import database bgp 65000 locator-set RLOC-SET1
distance site-registrations 250
map-cache site-registration
proxy-etr
proxy-itr 192.168.12.12
exit-service-ipv4
!
exit-instance-id
!
exit-router-lisp
!
router ospf 1

```

```
!  
router bgp 65000  
  no bgp log-neighbor-changes  
  neighbor 10.12.201.201 remote-as 65201  
  neighbor 192.168.11.11 remote-as 65000  
  neighbor 192.168.11.11 update-source Loopback0  
  !  
  address-family ipv4  
    neighbor 10.12.201.201 activate  
    neighbor 192.168.11.11 activate  
  exit-address-family  
  !  
  address-family vpnv4  
    neighbor 192.168.11.11 activate  
    neighbor 192.168.11.11 send-community extended  
  exit-address-family  
  !  
  address-family ipv4 vrf 100_White  
    neighbor 10.12.201.201 remote-as 65201  
    neighbor 10.12.201.201 activate  
  exit-address-family
```

C.10 Inet-PxTR22 Configurations

```
Inet-PxTR22#sh run  
!  
hostname Inet-PxTR22  
!  
!  
vrf definition 100_White  
  rd 1:100  
  !  
  address-family ipv4  
    route-target export 1:100  
    route-target import 1:100  
  exit-address-family  
!  
!  
!  
interface Loopback0  
  ip address 192.168.22.22 255.255.255.255  
  ip ospf network point-to-point  
  ip ospf 1 area 0  
  !  
interface LISP0  
  !  
interface LISP0.100  
  !  
interface GigabitEthernet1  
  ip address 10.25.22.22 255.255.255.0  
  ip ospf network point-to-point  
  ip ospf 1 area 0  
  !  
interface GigabitEthernet2  
  vrf forwarding 100_White
```

```
ip address 10.22.201.22 255.255.255.0
!
router lisp
security
locator-table default
locator-set RLOC-SET1
  IPv4-interface Loopback0 priority 3 weight 1
  exit-locator-set
!
locator default-set RLOC-SET1
service ipv4
  encapsulation vxlan
  itr map-resolver 192.168.21.21
  itr
  etr map-server 192.168.21.21 key nwktimes
  etr map-server 192.168.21.21 proxy-reply
  etr
  sgt
  distance site-registrations 250
  exit-service-ipv4
!
service ethernet
  itr map-resolver 192.168.21.21
  itr
  etr map-server 192.168.21.21 key nwktimes
  etr map-server 192.168.21.21 proxy-reply
  etr
  exit-service-ethernet
!
instance-id 100
service ipv4
  eid-table vrf 100_White
  map-cache 0.0.0.0/0 map-request
  route-import database bgp 65001 locator-set RLOC-SET1
  distance site-registrations 250
  map-cache site-registration
  proxy-etr
  proxy-itr 192.168.22.22
  exit-service-ipv4
!
  exit-instance-id
!
exit-router-lisp
!
router ospf 1
!
router bgp 65001
no bgp log-neighbor-changes
neighbor 10.22.201.201 remote-as 65201
neighbor 192.168.21.21 remote-as 65001
neighbor 192.168.21.21 update-source Loopback0
!
address-family ipv4
  neighbor 10.22.201.201 activate
  neighbor 192.168.21.21 activate
exit-address-family
```

```
!  
address-family vpnv4  
  neighbor 192.168.21.21 activate  
  neighbor 192.168.21.21 send-community extended  
exit-address-family  
!  
address-family ipv4 vrf 100_White  
  neighbor 10.22.201.201 remote-as 65201  
  neighbor 10.22.201.201 activate  
exit-address-family
```

C.11 ISP-201 Configurations

```
ISP-201#sh run  
!  
hostname ISP-201  
!  
interface GigabitEthernet1  
  ip address 10.12.201.201 255.255.255.0  
  negotiation auto  
  no mop enabled  
  no mop sysid  
!  
interface GigabitEthernet2  
  ip address 10.22.201.201 255.255.255.0  
  negotiation auto  
  no mop enabled  
  no mop sysid  
!  
interface GigabitEthernet3  
  ip address 172.16.200.1 255.255.255.0  
  negotiation auto  
  no mop enabled  
  no mop sysid  
!  
router bgp 65201  
  bgp log-neighbor-changes  
  network 172.16.200.0 mask 255.255.255.0  
  neighbor 10.12.201.12 remote-as 65000  
  neighbor 10.12.201.12 route-map PERMIT-ONLY-LOCAL out  
  neighbor 10.22.201.22 remote-as 65001  
  neighbor 10.22.201.22 route-map PERMIT-ONLY-LOCAL out  
!  
ip as-path access-list 100 permit ^$  
!  
route-map PERMIT-ONLY-LOCAL permit 10  
  match as-path 100
```

C.12 ISP-201 Configurations

```
Transit-SP101#sh run  
!  
hostname Transit-SP101  
!  
interface GigabitEthernet1
```

```
ip address 10.13.101.101 255.255.255.0
negotiation auto
no mop enabled
no mop sysid
!
interface GigabitEthernet2
ip address 10.23.101.101 255.255.255.0
negotiation auto
no mop enabled
no mop sysid
!
router bgp 65101
  bgp log-neighbor-changes
  neighbor 10.13.101.13 remote-as 65000
  neighbor 10.23.101.23 remote-as 65001
```

