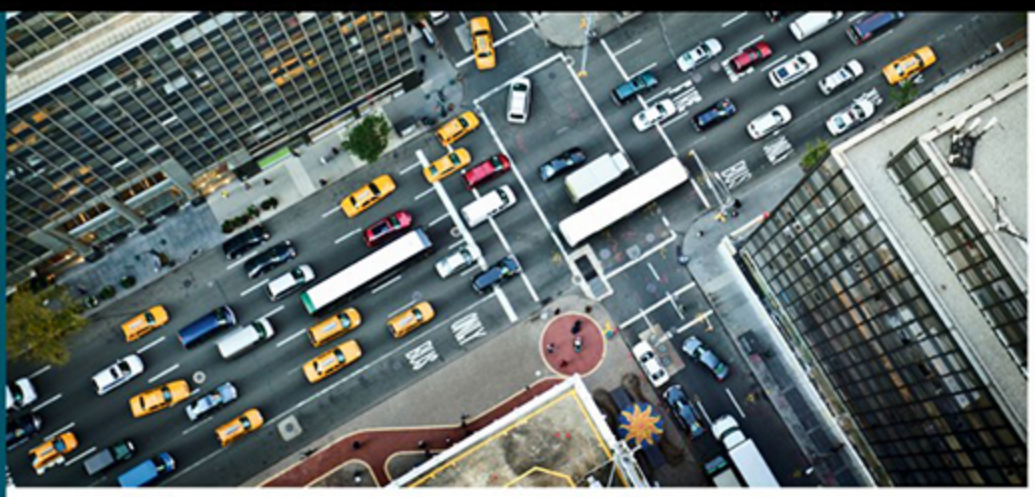


The LISP Network

Evolution to the Next-Generation of Data Networks

Rough Cuts

https://t.me/learningnets

Dino Farinacci
Victor Moreno, CCIE No. 6908

ciscopress.com

The LISP Network

Evolution to the Next-Generation of Data Networks

Dino Farinacci
Victor Moreno



https://t.me/learningnets

Victor Moreno
Dino Farinacci

Copyright © 2019 Cisco Systems, Inc,

Published by:
Cisco Press

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without written permission from the publisher, except for the inclusion of brief quotations in a review.

01 19

Library of Congress Control Number:

ISBN-13: 978-1-58714-471-4

ISBN-10: 1-58714-471-9

Warning and Disclaimer

This book is designed to provide information about the Locator/ID Separation Protocol (LISP). Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied.

The information is provided on an “as is” basis. The authors, Cisco Press, and Cisco Systems, Inc. shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book or from the use of the discs or programs that may accompany it.

The opinions expressed in this book belong to the author and are not necessarily those of Cisco Systems, Inc.

Trademark Acknowledgments

https://t.me/learningnets

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Cisco Press or Cisco Systems, Inc., cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

Special Sales

For information about buying this title in bulk quantities, or for special sales opportunities (which may include electronic versions; custom cover designs; and content particular to your business, training goals, marketing focus, or branding interests), please contact our corporate sales department at corpsales@pearsoned.com or (800) 382-3419.

For government sales inquiries, please contact governmentsales@pearsoned.com.

For questions about sales outside the U.S., please contact intlcs@pearson.com.

Feedback Information

At Cisco Press, our goal is to create in-depth technical books of the highest quality and value. Each book is crafted with care and precision, undergoing rigorous development that involves the unique expertise of members from the professional technical community.

Readers' feedback is a natural continuation of this process. If you have any comments regarding how we could improve the quality of this book, or otherwise alter it to better suit your needs, you can contact us through email at feedback@ciscopress.com. Please make sure to include the book title and ISBN in your message.

We greatly appreciate your assistance.

Editor-in-Chief

Mark Taub

Alliances Manager, Cisco Press

Arezou Gol

Product Line Manager

Brett Bartow

https://t.me/learningnets

Managing Editor

Sandra Schroeder

Development Editor

Marianne Bartow

Senior Project Editor

Tonya Simpson

Copy Editor

Chuck Hutchinson

Technical Editor(s)

Ramiro Garza Rios, Matt Esau

Editorial Assistant

Cindy J. Teeters

Cover Designer

Chuti Prasertsith

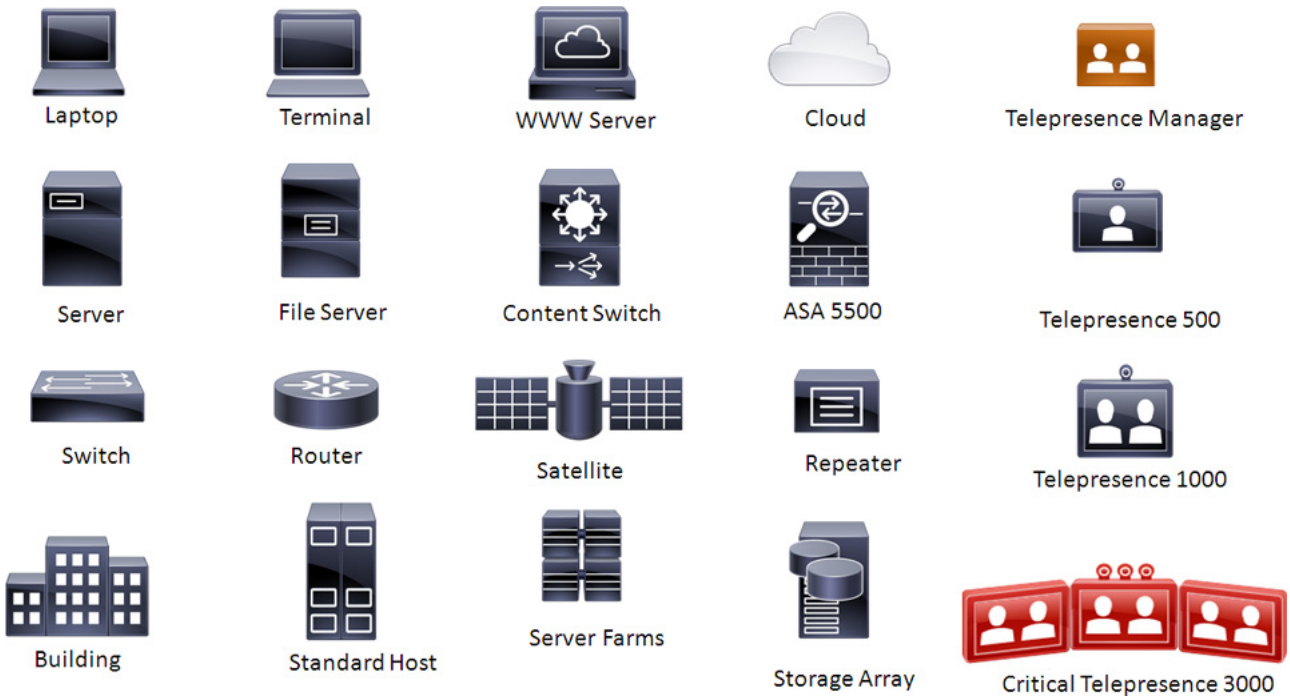
Composition

Indexer

Proofreader

https://t.me/learningnets

Icons Used in This Book



https://t.me/learningnets

Introduction

The LISP Network provides in-depth understanding on the most common applications of the Locator/ID Separation Protocol (LISP) and new applications of LISP that are helping address new trends and challenges in the networking industry. These trends are found in the data center cloud, the campus or branch access network, the WAN edge, the core of a service provider network, and a multitude of purpose-built networks that have emerged to support specific applications. LISP applications include data center workload mobility across private and public cloud locations, enablement of container networking, high rate mobility in cellular and fixed infrastructure, next-generation WAN models for scale and automation, massive scale Internet of Things connectivity, data confidentiality, IPv6 transition, multicast, and traffic engineering. This book provides a fundamental understanding of the underlying architecture and how it pertains to each application of LISP. The book is aimed at giving you the vision of how LISP can dramatically change the way networking is done in response to modern-day challenges and requirements.

GOALS AND METHODS

This book answers important questions for any fast-growing technology in the market, such as

• What problems does the technology address?

• How does the technology address the problems?

• How does the technology work?

• What are its applications?

• What is the future of this technology?

WHO SHOULD READ THIS BOOK?

This book addresses the preceding questions and provides insight into the latest

https://t.me/learningnets

applications gaining traction in the industry. The emphasis of the book is on the architecture of LISP and its applicability to modern-day IT requirements and trends. The book is an indispensable guide for any reader who wants to understand the future of the Internet and how LISP can be the solution for many of the challenges that enterprises face today to evolve the network into the next generation of Internet and support trends such as Agile Network programmability, IoT, security, and IPv6 with LISP.

We have structured the content so that the book is implementation agnostic and focused on the essence of the technology and its applicability. The intent was to separate timeless topics on technology architecture and applicability from implementation-dependent information.

HOW THIS BOOK IS ORGANIZED

Chapter 1: LISP and the Future of Networking

This chapter introduces the motivation, base principles, and history behind LISP. You read about how the base principles upon which LISP is built relate to the challenges and evolution of the Internet. The chapter also introduces some of the revolutionary applications that LISP enables. These applications are discussed in more detail in later chapters.

Chapter 2: LISP Architecture

The objective of this chapter is to provide a comprehensive overview of the technical architecture of LISP and how it works. You learn about the different architectural components of LISP and the key mechanisms and workflows that the protocol uses to deliver different network services.

Chapter 3: Data Center Trends

This chapter discusses the predominant trends of the data center and the role of LISP in enabling these trends. It examines how LISP and the revolutionary concepts introduced throughout its development have played a pivotal role in the evolution of the connectivity required in data centers to date. The chapter discusses mobility, network segmentation, and policy along with the potential role of LISP in the data center network moving forward.

Chapter 4: The Wide-Area Network: Bringing Traffic from Access to the Data Center

https://t.me/learningnets

This chapter discusses the challenges encountered in the wide-area network (WAN) and how networking technology evolved to meet these challenges and enable alternative approaches to WAN, such as the software-defined WAN (SD-WAN). LISP plays a pivotal role in the technological evolution of the WAN. How LISP addresses the different aspects of the modern SD-WAN is the focus of this chapter.

Chapter 5: Mega-Scale Access Networks: LISP, User Access, and the Internet of Things

The number of connected devices has grown dramatically in recent years. This trend continues and accelerates as the Internet of Things (IoT) becomes a reality. In this chapter, you learn about the considerations pertinent to connecting an unprecedented number of devices that are mobile and require data confidentiality. You also learn how LISP enables the evolution of the access network to address the stringent requirements of pervasive and high-density modern connectivity.

Chapter 6: Security

Traditionally, security was added onto the network as a separate stack of solutions and functionality. LISP is able to offer a comprehensive set of security functions that are integrated into the networking control and data plane to deliver segmentation, access control policy enforcement, connection integrity, confidentiality for data in-flight, and end-point anonymity. This chapter discusses how LISP provides integrated security services to improve the scale, flexibility, and manageability of the necessary security functions that the network must deliver. It also discusses how the LISP infrastructure itself is secured and protected from attacks that may attempt to compromise the network or use the network as a platform to launch an attack.

Chapter 7: LISP and the Next-Generation Mobile Network

The endpoint densities, rates of mobility, network redundancy, and path requirements being driven by next-generation applications pose demanding requirements on the network. These requirements go beyond what can be addressed by simply optimizing the existing incumbent networking models. A shift toward overlays and overlay-optimized demand control planes is necessary to satisfy this next wave of requirements. This chapter discusses how LISP supports mobility and how these mechanisms adapt to different use cases that should illustrate the high bar that was set for the network to surmount.

https://t.me/learningnets

Chapter 1. LISP and the Future of Networking

When turning on your faucet, do you ever think that water will not pour out? What about plugging your electrical cord into a power outlet and nothing happening? Even worse, what would you think if the faucet starting pouring out orange juice? Would that surprise you? Would you be disappointed? Perhaps if it was a certain fine grape juice from the Bordeaux region of France, that might be a pleasant surprise.

The whole point of this line of questioning is that you have come to expect certain benefits from certain services. When they are not available, you have come to believe your life is impacted in such a way that you are hindered in some capacity. The question then becomes whether network connectivity, and particularly the Internet, at this point in your life is like a utility. Can you live without it? How do you react when you need information and cannot get it? Do you feel lost when you're on an airplane or when that inevitable power outage occurs? Is the Internet more important than air or water? Can it be self-sufficient? Right now if there is no power, there is no Internet. You depend on the Internet almost as much as air and water.

In the 1980s, access to the Internet was a privilege. University researchers used it, and you may have used it at work. You could not send email to just anyone anywhere on earth, just to your coworkers. At that time, the user interface using the Internet was geeky. You had to memorize commands. You had to use stationary computers. No equipment was lightweight, quiet, or easy to use. As you can imagine, processing power and memory capacity were challenged and limited. But whoever thought you would need more? Whoever thought you would watch motion video images on a computer screen? Who knew you would do these things without a tape or disk attached to a computer? Who would know where and how far the video was coming from? This evolution is quite amazing.

Then came the 1990s, when the Internet really started to grow up. Many computer operating systems provided a communication language called Transmission Control Protocol/Internet Protocol (TCP/IP). This protocol was critical because it came at a time when beautiful displays and creative user interfaces were developed at the same time, referring, of course, to Windows, Linux, MacOS, and the game-changing World

https://t.me/learningnets

Wide Web. You could read magazines in full color from your computer. These capabilities came about at the turn of the millennium, when researchers, companies, doctors, magazine and newspaper publishers, and product vendors had their own web pages. You could shop on the Internet. You might see URLs for underwear ads on the sides of buses. Everything had a URL, and search engines took you where you wanted to go to get information. You could find anything imaginable online. Nothing had to be on paper anymore. You had instantaneous access to anything you wanted.

The applications were amazing. The way people were connected and how they efficiently communicated with each other were revolutionary. And the reason was the base layer, the TCP/IP protocol infrastructure, the piece of this puzzle called the Internet network. Wherever you traveled in the world, you could still reach the same people and resources that you did at home. Everyone and everything were addressable. The IP address was fundamental to making the Internet work. Each entity that communicated was assigned an IP address. That is how you identified the person or thing you wanted to talk to. IP addresses were the equivalent to telephone numbers of the circuit-based telephone network.

Telephone numbers not only identified that you wanted "to talk to Vint Cerf" (Father of the Internet) but also indicated that the number was "Vint Cerf at the Internet Society (ISOC)." The telephone device was on a desktop at Apple headquarters with a physical copper cable connection that the telephone company installed for Vint Cerf. Part of the telephone number had a type of area code that described what region of the United States the telephone resided. The telephones were geographically addressed because they were physically fixed.

Then in the later part of the first decade of the new millennium, cellular phones started becoming ubiquitous. These cell phones were wireless; they moved around more than they sat on a desk. When you used those telephone numbers, the geographical addressing was based on where you bought the cell phone and not necessarily where it was going to be used. The cell phone that moved around was now a roaming device. This capability is referred to in networking as *device mobility*.

The telephone number provided identification value because you, as a caller, wanted to talk to Vint Cerf. You really didn't care if he was in Cupertino, New York, or outside the country. A telephone number has no network connection relationship but instead identifies the person you can reach.

On the Internet, the IP address is just like a telephone number, but rather than making a call to a person with a telephone number, you send data packets to the resource that is

https://t.me/learningnets

assigned an IP address. IP addresses are not human friendly, they are difficult to remember, and their assignment to a particular computer can change often. Thus, the Internet took addressing one step further so that people didn't have to remember telephone numbers like they did for phones. Internet users used names, and those names were mapped into IP addresses. Some refer to these names as *Internet addresses*, whereas others like to refer to these names as *human addresses*. So, a name like www.ccn.com is considered an Internet address; similarly, a name like vint@isoc.org is referred to as an email address. The Internet address makes using the Internet easier for humans. The underlying machinery maps those Internet addresses to IP addresses so that the data can be forwarded by the IP routers that make up the Internet. In a way, the Internet has an automated phone book that maps Internet addresses to IP addresses so you can operate directly on Internet addresses and not worry about the IP addresses that the IP routers in the network need to forward the traffic.

Despite the introduction of Internet addresses, however, the network is built using IP addresses, which is what the TCP/IP stack on computer endpoints and routers understands and requires. The problem with IP addresses is that they are topologically dependent, even more so than telephone numbers are to the telephone network. The area code part of a telephone number depends on geographical location, but the rest of the telephone number identifies a person or party you can call. In contrast, all of the IP address assigned to a host represents where a host is attached to the Internet. The Internet Assigned Numbers Authority (IANA) assigns IP addresses, and they are usually assigned to large service providers or large corporations, but rarely to individuals. For the Internet to scale, the service providers and corporations that are assigned the IP addresses deploy the addresses in a structured way that allows the aggregation of many granular IP prefixes into coarser prefixes. The aggregation of prefixes is done at topological boundaries and makes IP addressing inherently align with the network topology. The topological distribution of IP addresses makes the use of IP addresses rather rigid. For instance, if AT&T provides your home connection to the Internet, the IP address you are assigned is part of the block of addresses assigned to AT&T and describes "a connection to Vint Cerf's house in Palo Alto connected to AT&T," which means you are getting an address within the AT&T block that is assigned to the San Francisco Bay Area. But, what if you wanted to send packets to Vint Cerf and he is not always in Palo Alto? Vint may be constantly moving around. Smartphones send and receive data over the Internet. They are assigned IP addresses. Is Vint Cerf's smartphone always connected to AT&T? What if he roams to where AT&T does not provide service? You want to talk to Vint Cerf and don't care how he is connected to the Internet. In this case, you need the IP address to identify Vint Cerf and don't care how

https://t.me/learningnets

packets get to him or where he is connected to the Internet. As Vint moves from one portion of the network to another, the IP address assigned to his device changes depending on where in the network topology he connects to the network.

IP addresses therefore are representative of location and tend to align with the network topology in an effort to enable summarization similar to what was provided with area codes in the telephone numbering system. The upper bits of the IP address represent the coarser prefix that more specific addresses belong to. Thus, an IP router in the Internet core could route traffic based on coarse prefixes and have routers toward the periphery of the network route based on more granular address information as the packets get closer to their destination. This principle of aggregability allows routers to hold a moderate amount of routing information and still provide reachability to all destinations. However, the ability to summarize is highly dependent on how well the IP prefixes align with the topology of the network. As IPv4 prefixes become more scarce and as IPv6 usage proliferates, the entropy of the prefixes assigned to different locations in the network increases. This translates to an increase in the number of unique prefixes that must be handled in the network core and throughout. To extract more prefixes out of the IPv4 space, smaller prefixes are used and injected from different locations in the network. This is done to utilize any idle addresses that are available in the current allocation of the IPv4 address space. The challenge is that the granular prefixes aren't always used in locations that are easily aggregated at topology convergence points. Thus, the granular prefixes often must be carried through the network core. The granular prefixes cause a significant increase in the number of prefixes that must be handled by the network core.

The genesis of this problem is the fact that the IP addresses are representative of both location and identity. The location must be topologically aligned; meanwhile, the identity should ideally not be constrained to the topology. The Internet architecture has an overloaded semantic with the IP address. IP addresses, in the traditional Internet, represent both the identity of the endpoint as well as the location at which the endpoint can be found. To ease this problem, the identity and location would ideally be decoupled from each other. The IP address for an endpoint needs to solely have identification semantics and become independent of the topology and location. You need an *endpoint ID*. You need this ID to not change when it is assigned to a person, smartphone, server computer, or anything else that is IP addressable. Let's call this endpoint ID an *EID* for short.

Now, when this topologically independent EID is available, you can solely identify whom you want to send data packets to. The Internet's responsibility is to deliver packets to wherever the EID is attached to the network. Maybe the EID is attached to

https://t.me/learningnets

two or more different points on the Internet. Smartphones have multiple wireless radios that can connect to a cell phone carrier network at the same time it connects to a Wi-Fi network. Maybe the EID can be reached via both paths. Maybe you can get more bandwidth to the EID because of the multiple connections it has. Maybe you can get more reliability for the EID. Consequently, reaching the EID is a different problem than that of identifying the EID.

In the Locator/ID Separation Protocol (LISP) architecture, these connection points for an EID are called *routing locators*, or *RLOCs*. They say where the EID is located on the Internet. The EID is the person you want to talk to, and the RLOCs are where the EID is located. Think of it like this: Jane Doe lives in a corner house, with two streets adjacent to her house. One street is called Main Street, and the other is called Wall Street. In this example, the EID is Jane Doe, and the RLOCs are Main and Wall. That is, you can reach Jane from Main Street or Wall Street. Using either street allows you to reach her house. Maybe Wall Street is congested with cars but Main Street is clear; in that case, you can get to Jane's house faster going on Main Street. The concept is the same with the Internet. Say you are in Starbucks, and you are connected to the cell network and the Starbucks Wi-Fi network. Perhaps the Wi-Fi network is congested because dozens of people are watching their favorite videos. You want good connectivity, so you can tell the Internet sites you are surfing that you want them to send their packets to you on the cell network.

What you achieve with this idea is removing both identification and location semantics from the IP address. You separate location from identification. You can use multiple IP addresses to solve this problem. These IP addresses are used as EIDs. Other IP addresses are used as RLOCs. A device could be assigned multiple EIDs as well as multiple RLOCs. Or a device can be assigned just EIDs, and a network infrastructure device can be the RLOCs for those EIDs.

Separating the semantics of identity and location is the fundamental concept for the focus of this book and the architecture of LISP. This level of indirection changes how you use the Internet. LISP allows you to roam, use multiple connections at one time, and scale the core of the Internet. Scaling the core of the Internet is crucial so that it can grow and support the newer applications that are coming with the proliferation of cloud-based services.

A BRIEF HISTORY OF LISP: MOTIVATION, BASE PREMISES, EVOLUTION

By the mid 2000s, the core of the Internet held 500,000 routes. A *route* is a piece of

https://t.me/learningnets

state that allows data packets to flow from a data source to a data destination. This doesn't mean the core of the Internet supported only 500,000 connected devices. A route describes a collection of devices, sometimes thousands of devices. So, a route provides reachability and direction to a network collection of devices. A collection of devices is identified by an IP prefix, so for each IP prefix, there must be a route. At different stages of the topology, routes for more or less granular prefixes are required. If the IP address space is well organized, prefixes are aggregated into a smaller number of broad-covering prefixes, which in theory should allow the Internet routing tables to scale.

By the mid 2000s, IANA had assigned IP prefixes to large organizations, and these organizations managed them the best way they could. Organizations tried their best to deploy their assigned prefixes in ways that allowed the prefixes to be aggregated so that they could be summarized in the network core. However, the IPv4 address space had mostly been assigned and was a scarce commodity. Any unused portion of the space had to be disaggregated into smaller prefixes that could be redeployed regardless of whether they could be aggregated and summarized from their new location. The topological location where these smaller prefixes were redeployed was usually not a location from which the prefix could be aggregated into the larger prefix it originated from. This lack of aggregability resulted in a large amount of entropy in the distribution of the IPv4 address space. In addition, the random distribution of the IPv4 prefixes meant they could not be summarized and that many of these smaller prefixes must be carried in the core of the network. As a result of the scarcity of IPv4 addresses and the increasing use of IPv6 addresses, the core Internet routing table was growing at a faster pace than the routers could support and faster than core network providers could upgrade routers. Adding more hardware with more memory temporarily solved the problem, but it was a cat-and-mouse race. You had to be smarter about dealing with this rate of growth and needed to do more than brute-force hardware upgrades.

In the fall of 2006, the Internet Architecture Board (IAB) assembled a set of experts for a routing workshop in Amsterdam. The goal was to discuss the routing table growth problem and to start research groups in the Internet Research Task Force (IRTF) to explore solutions as quickly as possible. The research groups decided that building a level of indirection in the routing system was a fundamental concept that had been discussed decades ago but was too hard to add to the Internet architecture. However, if you could insert the level of indirection by doing Locator/ID address separation incrementally, you could reduce the load on the core routing system sooner rather than changing the Internet architecture in a more disruptive way. If you could do it while supporting both IPv4 and IPv6, even better. Actually, it was absolutely necessary to support IPv6 because the Internet was running out of IPv4 addresses.

https://t.me/learningnets

Then in 2007, the first LISP Internet Draft was written. A few engineers at Cisco Systems developed the architecture and protocols to start a prototype early in the year. By the spring of 2007, pilot LISP networks were deployed for experimentation. The LISP effort started with hundreds of contributions by individuals who had a passion to save and scale the Internet. Soon this pilot network grew to hundreds of sites in 30 countries. This became known as the LISP Beta Network.

LISP continued to be developed in the IRTF research groups though 2009, when it then became an Internet Engineering Task Force (IETF) working group. As of this writing, the LISP working group is still active, and many use cases are being developed.

Because LISP provides a level of indirection for routing and addressing, it creates an overlay network where the core routers forward packets to RLOCs and the overlay forwards packets to EIDs. Because the EID assigned to an endpoint can be constant and the RLOCs can change, a natural mobility feature is the result. One of the many use cases for using LISP is the basic support for EIDs moving around. Whether smartphones, virtual machines, provider-to-provider roaming (either physical or in the cloud), or IoT devices, all are assigned EIDs with changing RLOCs.

LISP IN THE STANDARDS AND OPEN COMMUNITY

LISP is an open protocol via open standards from the IETF. All work performed in the past ten years has been with published specifications and open-source implementations. LISP is free from IPR because the original designers believed that was the best way to bring Internet goodwill for such a sharable critical resource as the Internet.

You can find the LISP RFC specifications and work-in-progress Internet Drafts at http://datatracker.ietf.org/wg/lisp. Public domain open-source implementations are located at http://www.github.com. Many vendors have LISP product offerings with a wealth of information, slideware, white papers, and marketing materials.

Implementations of LISP have been put in routers, switches, smartphones, virtual machines running virtual switches, and smaller devices like Raspberry Pis. LISP runs in many cloud providers as well as in container technology on a variety of operating systems.

USE CASES FOR LISP: SUPPORTING FUTURE TRENDS

After the routing workshop in Amsterdam, it became evident that several fundamental

https://t.me/learningnets

applications were available for LISP beyond the applications that fueled the original motivation for the LISP work. The original use cases developed from the routing workshop in Amsterdam were

• Reducing the core router's routing table size

• Preventing multiply connected sites (multihoming) from adding more routes in the core routing system but making multihoming easier to manage

• Allowing sites to keep their addresses and easily move their site connections from one service provider to another and encouraging the use of provider independent addresses

Because a level of indirection in routing and addressing is introduced with LISP, EID addresses are not known in the core. Most of the entropy exists in the EID addresses, so if they can be removed from the core routers, the remaining address space can be summarized well and the size of the routing tables will be reduced significantly.

Similarly, one or more RLOCs may be able to reach the EIDs, and this association doesn't require that any additional RLOC state be held in the core. So multihoming is achieved without incurring any scale cost when locator and ID are separated and the EIDs are removed from the underlying core network.

Due to the indirection used between identity and location, many different types of addressing can be used on the edges of the network, in the LISP sites, different from what is used in the core network. Therefore, LISP sites could use IPv4 private addresses or IPv6 site-local addresses, and they can be kept separate from the core in their own virtual private networks (VPNs). These private addresses can connect sites together without the core knowing. This means that the overlay network supports VPNs with any address structure while the core network just routes packets to the RLOCs of the sites.

Even different address families can be used on the overlay than what is used in the core network, also known as the underlay. So you create a blend from the use cases above to make new use cases. For instance, LISP sites are assigned IPv6 EID addresses while the underlay delivers packets to LISP sites using IPv4 RLOC addresses.

Because EIDs are independent from the core, they cannot move only from one service provider to another, but they move anywhere and get assigned new RLOCs dynamically. Therefore, all use cases previously described also provide mobility support at the same time. For example, a vehicle that drives on a highway is assigned a provider-independent EID address from the vehicle identification number (VIN). This address is not routable and known by the core network. When the vehicle drives by a

https://t.me/learningnets

road-side-unit (RSU), it sends data that the RSU forwards upstream to the Internet infrastructure. At the same time, the EID gets a binding with the RSU's RLOC address, so packets are returned from the Internet to the RSU and forwarded to the vehicle.

You will find all sorts of uses for EIDs and RLOCs. They can represent newer types of information. For example, an EID can be a character string identifying a service, person, or city name, for example. You'll find that RLOCs can be geo-coordinates describing a physical location for an EID. RLOCs could also encode a path that a packet may traverse from one EID to another.

Later chapters describe in simple terms how EIDs get bindings with RLOCs and how they are stored in a Mapping Database System. This book explores how the Mapping Database System provides scale for billions of addressable objects on the Internet by using a pull-based approach versus a push-based approach that the Internet core uses today.

The Internet architecture in place today must handle all EIDs and RLOCs in its core and therefore cannot support these many roaming addressable objects at near the scale that the level of indirection of LISP offers. This book describes how access control and policy are supported via access to the Mapping Database System. It becomes evident how the Mapping Database System is centralized for access but deployed in a distributed fashion for scale and security.

The Mapping Database System and how to access it are part of the control plane design for LISP. How EIDs in packets get mapped to RLOCs and then forwarded in the underlay core network is part of the data plane design for LISP. Later chapters cover in detail of how both of these designs work.

https://t.me/learningnets

Chapter 2. LISP Architecture

The Locator/ID Separation Protocol (LISP) enables the fundamental notion of separating location and identity. It does so by providing the necessary control and data plane mechanisms to support a distributed directory of the mappings between identities and locations.

This chapter describes the control and data plane architecture of LISP in the context of its foundational principles and their implications in enabling networking services that augment the functionality delivered by existing networking protocols.

SEMINAL IDEA: LOCATION-IDENTITY SEPARATION

Identity and location in networking are akin to what you would consider these concepts to be in your daily life. In your daily life, your identity is usually represented by your name, and your location is usually represented by a street address. Street addresses may correspond to your home, office, parents' home, and so on. When someone wants to send a gift or letter to you, that person looks up your street address and uses this address to instruct the mail service where to deliver the gift. From that point onward, the mail service routes the packet based solely on location. To obtain your address, the sender usually leverages a directory to locate your address by searching for your name. In the mail system example, the phone book is a likely directory that people use to find addresses for others they need to send packets to.

As discussed in Chapter 1, "LISP and the Future of Networking," addresses of host computers in a data network have traditionally conveyed two sets of information in a single address: the host's identity and its location. As a consequence, your computer's IP address changes when you connect to the network at home, at the coffee shop, or at your office. However, the identity of your computer and its applications don't change during all these location changes. Location and identity are really two loosely coupled yet independent pieces of information, as illustrated in the mail system example. The traditional method of addressing used in IP networks, however, blends location and identity into a single address namespace.

https://t.me/learningnets

LISP proposes the separation of location and identity into two separate namespaces:

• Identity namespace

• Location namespace

Network hosts are referred to as endpoints in LISP and are assigned addresses in the identity namespace. When network addresses play the identity role, in LISP they are called endpoint identifiers (EIDs) and they make up the EID namespace. These addresses are equivalent to the person's name in the mail system example. Just like the person's name, these addresses do not provide enough information to reach the person or endpoint. Therefore, they are not used to route a packet to a destination but are used as a key to find the desired location information in a directory that maps identity to location.

The network devices to which hosts attach are assigned addresses in the location namespace, just like buildings are assigned a street address in the mail system. These addresses represent location; they are equivalent to the street addresses in the mail system example and make up what is known in LISP as the routing locator (RLOC) namespace. Addresses in the RLOC namespace are fully routable, just like the street addresses are fully routable in the mail system. So all network devices participating in the RLOC namespace are able to send packets to each other. The RLOC space with its associated routing protocols and network connectivity is equivalent to the mail system with all of its people, roads, trucks, planes, distribution centers, and post offices designed to transport packets from one location to another, from one street address to another.

Similar to the role the phone book plays in the mail system, LISP maintains a directory of identities and their corresponding locations; basically, LISP maintains a directory mapping the EID space to the RLOC space. LISP as a protocol defines all the necessary signaling to populate this directory, keep it updated, and enable the network elements to consult the directory and resolve the location of EIDs of interest.

LISP is a protocol focused on the specific task of handling the database where identity and location namespaces are mapped to each other; therefore, it isn't a routing protocol as traditionally defined. Routing and forwarding of data packets ultimately continue to be the responsibility of traditional routing protocols in the RLOC namespace. LISP augments these protocols by adding a layer of namespace handling that enables functionality that is otherwise difficult to procure natively in traditional routing protocols. Because of the separation of the namespaces and their loose coupling with

https://t.me/learningnets

basic routing and forwarding, the definition of both EIDs as well as RLOCs is extended beyond simple addressing to include policy semantics and other metadata that enables functionality, such as host mobility, large-scale segmentation, traffic engineering, location-aware policies, location tracking services, and other services in which correlating topological location to identity provides a unique advantage. The implications are far reaching and mostly anchored in the notion of being able to handle information in the context of the network topology.

One important implication of the separation of location and identity is that the routing that handles the RLOC namespace is relieved from handling the entropy introduced by the diverse user networks and devices that connect to the network. Different networks and devices connect in a variety of ways and usually without regard to the impact of their connection to the core network. The state related to the user networks and endpoint devices in the EID namespace can be unstructured and very large. Relieving the core network from the responsibility of handling the EID namespace allows the RLOC space in the core network to be structured in the best possible way while remaining stable and hence reliable.

MAP AND ENCAPSULATE

LISP enables what is broadly referred to in the networking industry as an *overlay*. Figure 2-1 illustrates the main elements of an overlay service. In it, two planes of functionality enable an overlay network:

• Virtual network in the overlay plane

• Transport network in the underlay plane

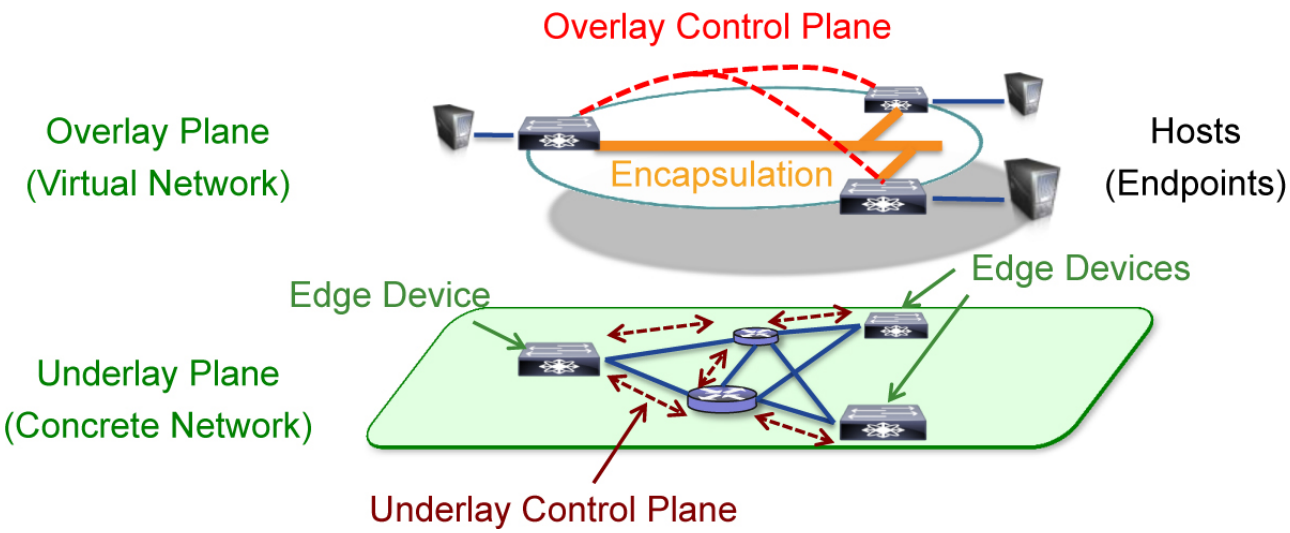


Figure 2-1 *Functional Components of a Network Overlay*

The underlay plane is a traditional network, which provides connectivity between

https://t.me/learningnets

network devices (routers and switches) but isn't aware of the endpoints that attach to the network edges. Multipathing and resiliency are optimized in the underlay network with well-understood traditional routing methods. The underlay handles routing only between RLOC addresses.

The overlay plane is a virtual network service that is delivered over the top of the underlay network. The overlay functionality is enabled at the edges of the network only. Traffic between hosts is tunneled between network edge devices across the underlying core network. To determine where to tunnel the traffic to, the edge devices need to obtain the information regarding which edge device a particular host destination may be connected to. This process of mapping identity to location to encapsulate traffic to the destination's location is often referred to as *map and encapsulate*.

The LISP functionality is enabled mainly at the edges of the network. From the LISP perspective, the edge devices where LISP is enabled are referred to as *tunnel routers*. Because the role of the tunnel router is directional, ingress tunnel routers (ITRs) and egress tunnel routers (ETRs) are used, referring to the ingress to the LISP overlay and egress from the LISP overlay, respectively. It is common to see the general role of an edge device referred to as an xTR when directionality is not relevant. The roles and responsibilities of the different types of xTRs are defined in more detail later in the "LISP Roles" section of this chapter, but it is worth noting at this point that requesting a mapping and encapsulating the traffic are ITR functions.

An ITR uses tunnels to encapsulate EID traffic and transport it over the RLOC underlay. From this perspective, there is an inner header in the EID space and an outer header that uses RLOC addresses. Thus, an ITR can encapsulate traffic for any type of EID address family into tunnels using any type of RLOC address family. For example, ITRs may encapsulate traffic for IPv4 EIDs using an IPv6 outer header, or ITRs may encapsulate traffic for MAC EIDs using an IPv4 outer header. Any combination is possible, in theory, and does not affect the way in which the LISP control plane operates.

DEMAND-BASED ROUTING AND CACHING

As mobility and rich metadata become the norm in the EID namespace, the scale of the EID namespace grows exponentially while the ability to structure the namespace and summarize it around topology boundaries disappears. This basically means that the EID space is, from the perspective of the edge devices, a flat, unstructured, and large namespace. Thus, the edge devices benefit from selectively downloading only the state needed to support the connections they must service. For instance, if an edge device

https://t.me/learningnets

services only connections to EIDs in a particular subnet, it is of little use to the edge device to obtain location mappings for other EID subnets. In fact, the edge device probably does not have enough forwarding memory capacity to hold all that state.

LISP addresses the scale concerns of the growing EID space by using a demand-based model that allows ITRs to download only the information they need rather than the push model used in routing protocols.

The demand model that LISP uses is similar to the Domain Name System (DNS) model illustrated in Figure 2-2. The DNS manages the mapping between a host's human-readable name and its IP address. In DNS, a host queries the DNS only when it needs the IP address (used as the EID in LISP) for a hostname, and it caches the response from the DNS. LISP uses a similar model to resolve host IP addresses (identity or EID) and obtain the address of their connecting router (location or RLOC). In the LISP model (like the DNS model), the Mapping Database System is queried on demand for specific destinations, and only relevant information is cached.

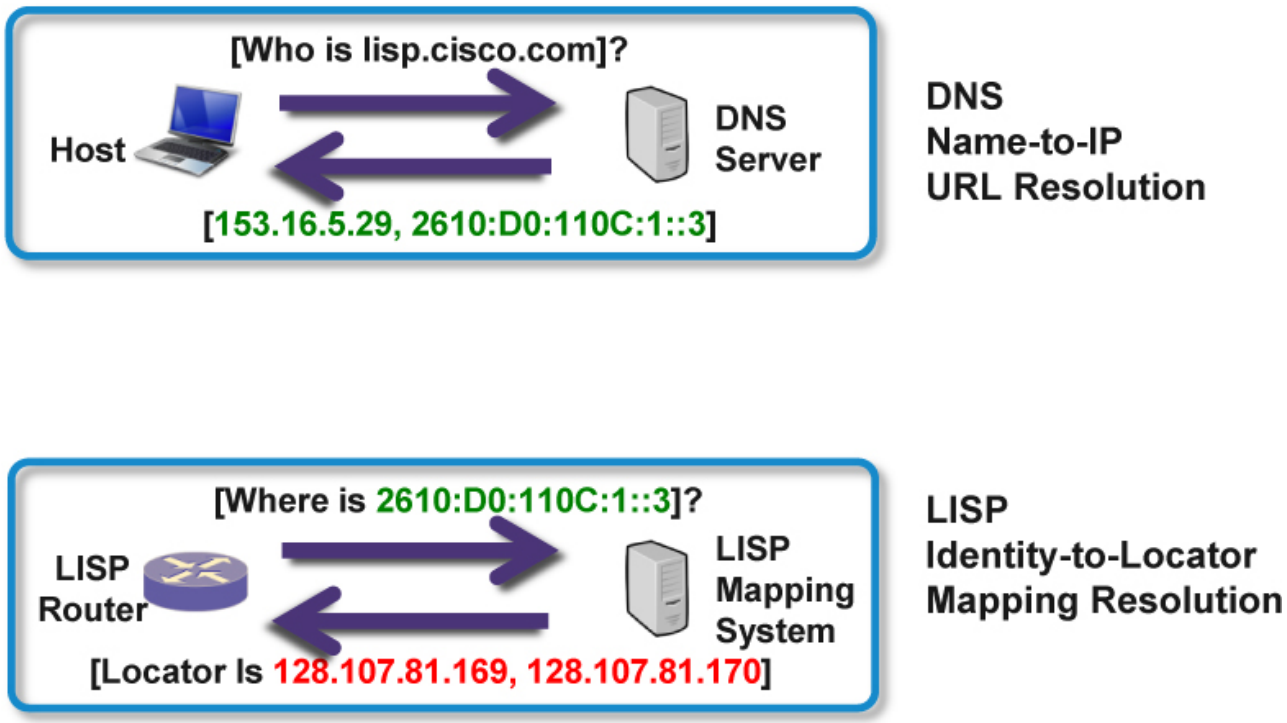


Figure 2-2 *LISP and DNS*

The LISP's demand and cache mechanism is illustrated in more detail in Figure 2-3. Generally, an ITR does not possess a local copy of the location mappings for all EIDs that it may be required to send traffic to. Thus, when an ITR receives traffic for a particular EID destination, it requests the mapping for the destination EID from the LISP Mapping Database System (the LISP database that contains all the mappings). The Mapping Database System responds to this request with the relevant mapping, and the ITR then caches this mapping in its forwarding table. Subsequent packets to the

https://t.me/learningnets

cached destination are encapsulated to the RLOCs specified in the cache without triggering a new query to the Mapping Database System. The cached mapping may refer to an EID for a single host or to an entire EID prefix. When the mapping is for an EID prefix, traffic to any destination covered by the cached EID prefix no longer triggers a query to the Mapping Database System but uses the cached forwarding entry to encapsulate the traffic to the RLOCs specified in the cached mapping.

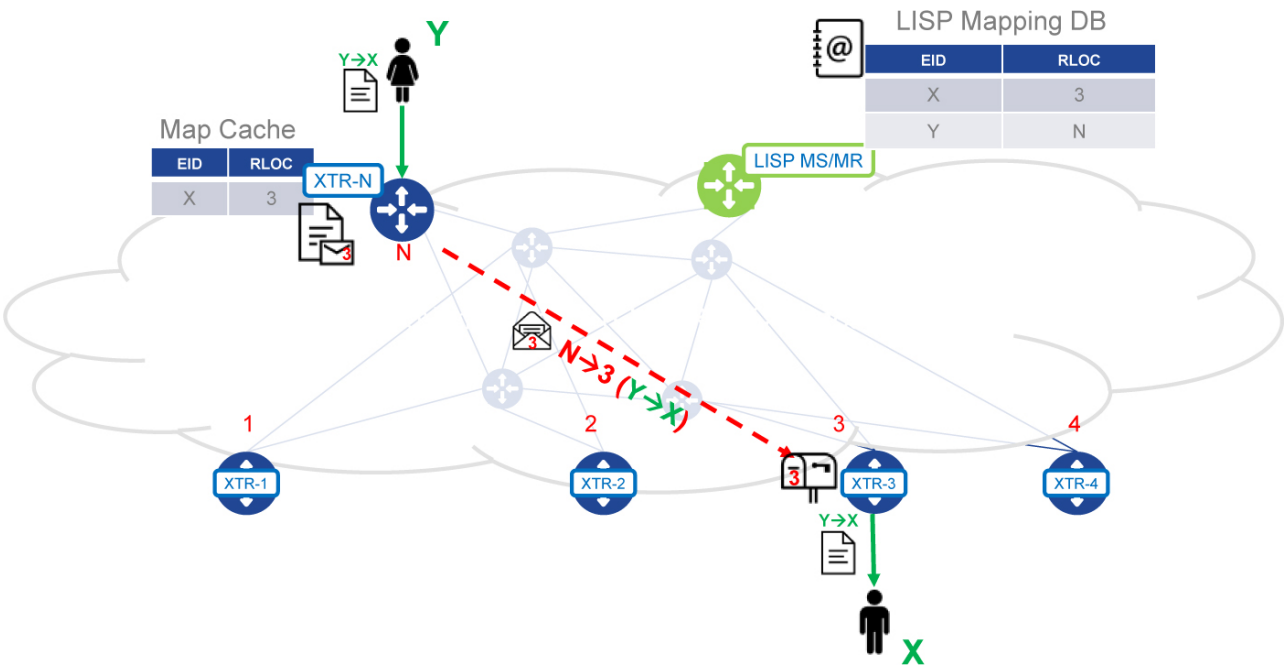


Figure 2-3 *Location Resolution on Demand*

From a name resolution and caching perspective, LISP presents many similarities to the Domain Name System. One of the main differences is that DNS operates solely on fully qualified domain names (FQDNs), whereas the LISP EID space is made of network addresses (IP or MAC) that are augmented with metadata reflecting security or segmentation context. The other salient difference is that LISP includes a series of mechanisms to trigger updates to existing ITR Map-Caches rather than waiting for the caches to expire.

LISP ROLES

For LISP to effectively provide the service of a demand-based overlay that separates location from identity, certain functional roles must exist. A few of these roles were loosely introduced in the description of the foundational principles of LISP. This section formalizes the definition of the different roles.

Tunnel Routers

Tunnel routers are the network devices at the edges of the LISP network. These routers perform the encapsulation and de-encapsulation of EID traffic into RLOC addressed

https://t.me/learningnets

tunnels. These routers also are responsible for populating and querying the Mapping Database System. Based on the direction of traffic, a tunnel router may act as an ingress tunnel router or it may act as an egress tunnel router, where the terms *ingress* and *egress* refer to the LISP overlay.

Most tunnel routers are usually deployed concurrently as both ITRs and ETRs. This is the most common deployment scenario, and a router in such a configuration is often referred to as an xTR. However, in specific cases, the ability to deploy ETR functionality independent from ITR functionality is required. Most LISP documentation, specifically the standard specifications, uses the terms *ITR* and *ETR* distinctly to avoid any sort of confusion. This book, which describes the roles separately, adheres to that practice, but you should keep in mind that they are usually deployed jointly.

Ingress Tunnel Routers

ITRs are the entry point of traffic from the LISP site into the overlay. ITRs are responsible for querying the Mapping Database System to obtain locator mappings for EIDs for which they receive traffic. ITRs use a LISP message known as a Map-Request to issue such queries. In the mail system analogy, the ITR is the sender looking up their receiver in the phone book.

ITRs are also responsible for caching the mappings received from the Mapping Database System in a Map-Cache. Caching is required to minimize the amount of churn on the Mapping Database System and make the system more efficient. In the mail system scenario, the ITR is the sender making a note of the receiver's street address in a personal address book for faster access.

ITRs are also responsible for encapsulating traffic to the destination location. ITRs do this by selecting a viable RLOC record from the mapping for the destination EID and encapsulating the traffic in a tunnel using the selected RLOC as the tunnel destination address. The ITR verifies the viability of an RLOC record in many ways, some of which are described in the "Data Plane" section. In general, the ITR must check that the candidate RLOCs are reachable and available in the underlay, and it must then proceed to calculate a hash on the EID traffic header, including the priority and weight values included for each RLOC record as part of the mapping received from the Mapping Database System. In the mail system analogy, the sender performs a lookup in the directory and finds a few addresses and then does due diligence to make sure all addresses are current. The sender then picks one of the addresses based on what the receiver has stated as a preference. Finally, the sender writes the address on a box, puts the gift inside the box, and hands the packet to the mail system for delivery.

https://t.me/learningnets

Egress Tunnel Routers

ETRs are the exit point of traffic from the LISP overlay network. ETRs are responsible for de-encapsulating the traffic they receive. In the mail system analogy, de-encapsulating traffic is equivalent to the receiver getting the packet in the mail, opening the box, and extracting the gift from the box.

ETRs are authoritative for the set of EIDs locally available at their site. The EID to RLOC mappings, along with their priorities and weights, are defined and kept at the ETRs. The ETRs are responsible for registering these mappings with the Mapping Database System. An ETR uses LISP Map-Register messages to register as authoritative for the mappings of the EIDs local to its site. In the mail system analogy, this is equivalent to the receiving party registering information with the publisher of the phone book (or more likely with the phone company).

ETRs are responsible for replying to queries about the EID mappings they have registered. In this case, the ETR is effectively part of the Mapping Database System and authoritative for replying to map resolution queries. In LISP, map resolution queries are known as *Map-Requests* and the corresponding replies are known as *Map-Replies*. In this mode, rather than just responding, the Mapping Database System routes Map-Requests to the authoritative ETR for the EID being requested and allows the ETR to issue a Map-Reply directly to the ITR. In this role, the ETR is effectively part of the Mapping Database System.

Proxy Tunnel Routers

Tunnel routers, as described up to this point, are assumed to connect to a set of EID networks registered in LISP. However, when a tunnel router is connected to networks that are not registered in LISP, the tunnel router effectively connects non-EID prefixes to prefixes in the EID namespace. It is important to note that non-EID prefixes or prefixes not registered in LISP are part of the RLOC prefix namespace. This interoperability role is key for LISP-enabled networks to communicate with networks that are not LISP-enabled and allow the incremental deployment of LISP. Tunnel routers operating in this mode are known as proxy tunnel routers, and just like regular tunnel routers, they operate differently as they serve ingress or egress traffic. The definition of the PITRs and the protocol mechanisms associated with their deployment to provide interoperability between LISP- and non-LISP-enabled networks is documented in RFC6832.

Proxy Ingress Tunnel Routers

Proxy Ingress Tunnel Routers (PITRs) receive traffic destined to LISP EIDs from non-

https://t.me/learningnets

LISP areas of the network. Upon receipt of the traffic, PITRs behave just like ITRs do: they resolve the mapping for the destination EID and encapsulate the traffic toward the right location. In the mail system analogy, the PITR is equivalent to a sender who wants to send a gift to a receiver but delegates sending this gift to an assistant who acts as a proxy sender by looking up the receiver's address, packaging the gift, and putting it in the mail.

PITRs request mappings and encapsulate traffic toward an EID regardless of whether the source of the traffic is an EID or not; this is the basic difference between configuring the router as an ITR or PITR. When configured as an ITR, the router checks whether the source is registered in LISP as an EID before doing anything else. If the source isn't an EID, the ITR does not handle the traffic as LISP traffic, and forwarding of this traffic depends on the presence of a route to the destination in the underlying routing tables. In other words, if the source is an RLOC (not an EID), an ITR assumes the destination is also an RLOC and allows the router to handle it as such in the underlying routing. A PITR does not check on the source because its role is to actually receive traffic from RLOC sources and forward it to EID destinations. Therefore, the fact that a source is in the RLOC space actually indicates to the PITR that it needs to forward the traffic in LISP.

PITRs must attract traffic to themselves if they are to be able to forward traffic to EID destinations. To this effect, PITRs must be configured to advertise to the non-LISP network any EID prefixes they may be able to service. So, PITRs act as honeypot routers for any traffic destined to the EIDs they can reach. A multitude of PITRs may be deployed to provide a variety of access paths to the LISP network. Some companies leverage their broad presence at a multitude of co-locations across the world to provide PITR services commercially for LISP users to leverage in their network design.

PITRs enable the inbound connection of RLOCs outside the LISP network to EIDs inside the LISP network. Because the return traffic is destined to an RLOC, it is handled by the underlying routing without being encapsulated. This may or may not be viable, depending on traffic symmetry requirements and what kind of Reverse Path Forwarding (RPF) checks are in place in the underlay network. In many cases, traffic between LISP and non-LISP endpoints must be encapsulated in both directions. Thus, an egress equivalent to the PITR is required.

Proxy Egress Tunnel Routers

Proxy Egress Tunnel Routers (PETRs) are the counterpart to the PITRs and allow communication between RLOCs and EIDs to be symmetrically encapsulated as traffic traverses the LISP core.

https://t.me/learningnets

Like a regular ETR, a PETR de-encapsulates traffic tunneled to its RLOCs. However, a PETR is not authoritative for any EIDs because its purpose is to provide connectivity for EID sources to reach destinations in the RLOC space outside the LISP network. Therefore, a PETR does not register any addresses with the Mapping Database System.

In the mail system analogy, a receiver may have a particular address within a separate shipping system—for instance, within the internal mail system in a large corporate campus. However, that address is not exposed in the phone book, so all you can do is send the packet to the shipping and receiving department for the corporation and rely on the staff to deliver the packet to the final receiver within the internal system. From this perspective, the PETR is the shipping and receiving department of the corporation, LISP is the mail system, and the corporate internal mail is, well, the Internet. Funny how things change when you look at them from the LISP perspective!

If the PETR doesn't register any addresses with the Mapping Database System, how does an ITR know that it should send traffic to the PETR? This is a bit like default routing; basically, if the ITR requests a mapping for a particular destination and the destination is not registered in the Mapping Database System, the Mapping Database System sends a Negative Map-Reply message to the ITR indicating that the destination is not registered. The ITR should be configured to send traffic to the PETR for any destinations for which a Negative Map-Reply is received.

When the Mapping Database System receives a Map-Request for a destination that is not registered, it calculates the shortest prefix that covers the requested destination but that does not cover any LISP EIDs. The calculated non-LISP prefix is included in the Negative Map-Reply issued to the ITR so that the ITR includes in its Map-Cache an entry for the non-LISP prefix. The ITR knows from that point onward to send traffic that matches that non-LISP prefix to the PETR.

Mapping Database System

As you have probably inferred by now, the Mapping Database System is the address directory or phone book of LISP. Its function is to maintain the database of mappings and service queries for those mappings. The two distinct roles in the Mapping Database System are as follows:

• Map-Server (MS)

• Map-Resolver (MR)

Often both of these roles are co-located, but they are maintained as separate

https://t.me/learningnets

architectural components because their separation is the basis for providing resiliency, distributing and scaling the Mapping Database System.

The Map-Server (MS) receives all EID registrations that install the registered EID to RLOC mappings in a database. As shown in Figure 2-4, ETRs register EIDs with their corresponding RLOCs. The Map-Register messages are sent from the ETRs to the Map-Server. Thus, the Map-Server provides the main interface between the Mapping Database System and the ETRs. When the Map-Server receives a Map-Register message, it installs the EID to RLOC mappings received in the Mapping Database System.

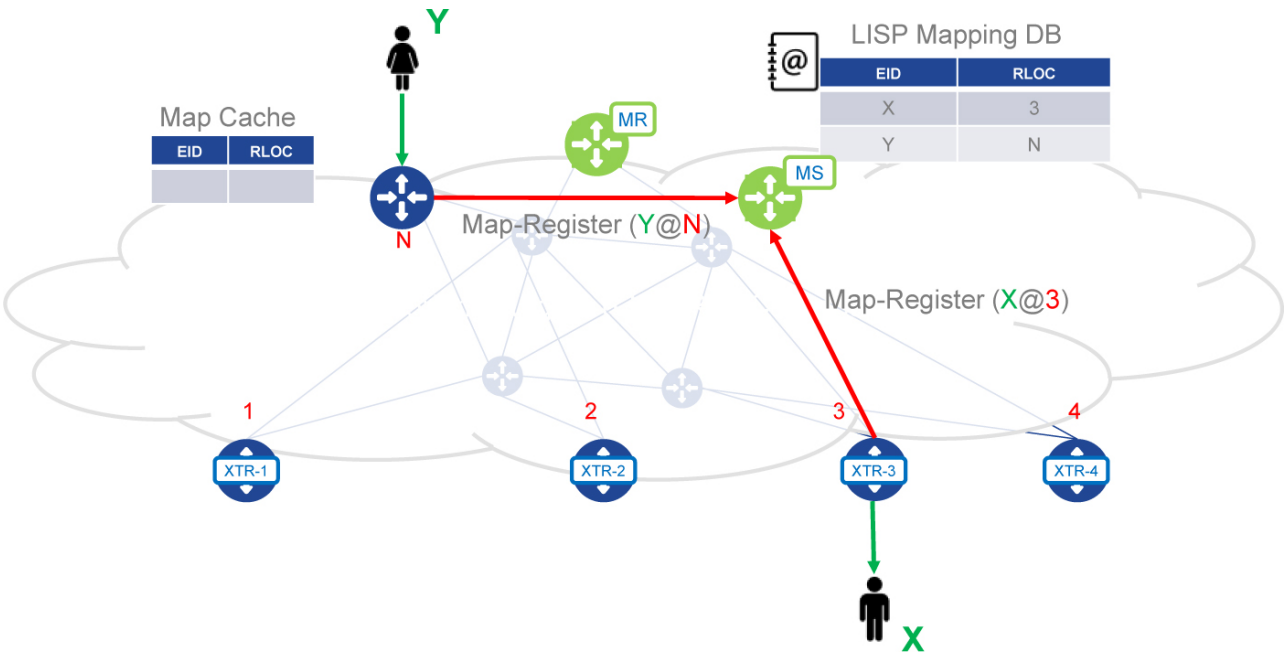


Figure 2-4 *Registration of EID to RLOC Mappings*

The Map-Resolver (MR) is responsible for servicing Map-Requests. The Map-Resolver provides the main interface between the Mapping Database System and the ITRs. As illustrated in Figure 2-5, when a Map-Resolver receives a Map-Request, it routes that Map-Request to the authoritative ETR, via the authoritative Map-Server, so that the ETR responds directly to the Map-Request. The mapping registration may indicate that the Map-Server needs to reply to the Map-Request rather than forward the request to the authoritative ETR.

https://t.me/learningnets

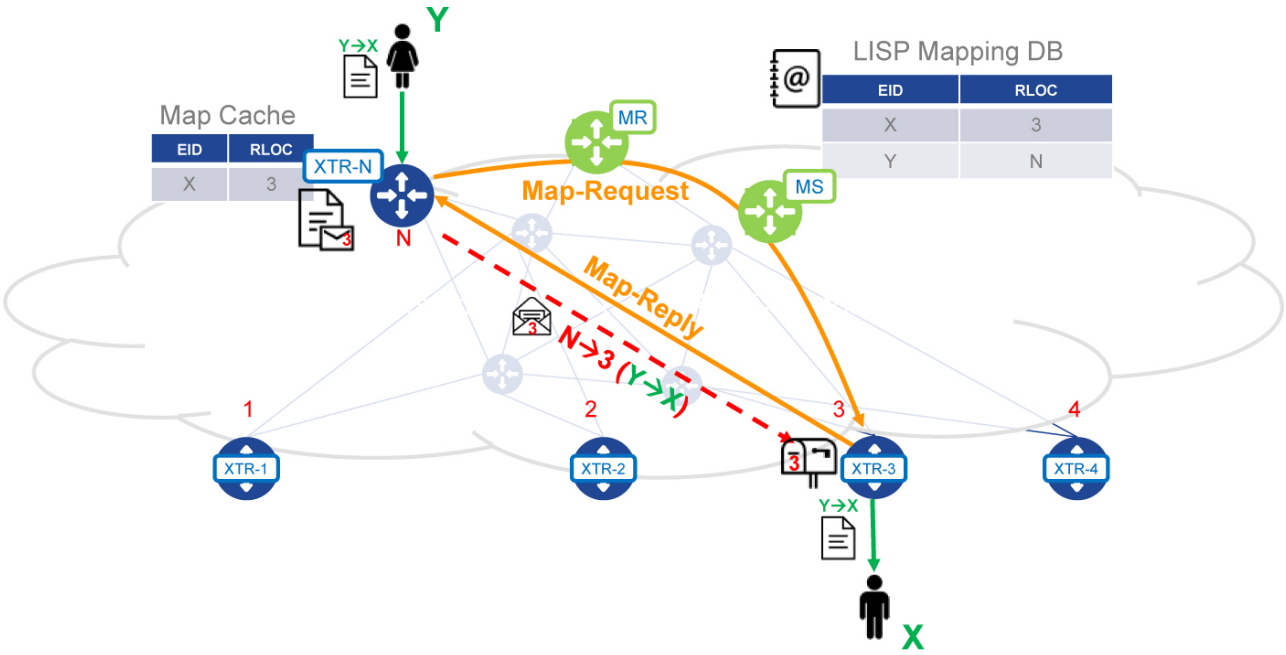


Figure 2-5 *Mapping Resolution*

Resiliency in the Mapping Database System is achieved without additional protocol messaging. The resiliency mechanism used in LISP is illustrated in Figure 2-6. ETRs may register to multiple Map-Servers and thus generate resilient state across more than one Map-Server. The Map-Servers synchronize their registered entries solely by receiving their information from a common source (the registering ETRs); a database synchronization protocol is not at play between the Map-Servers.

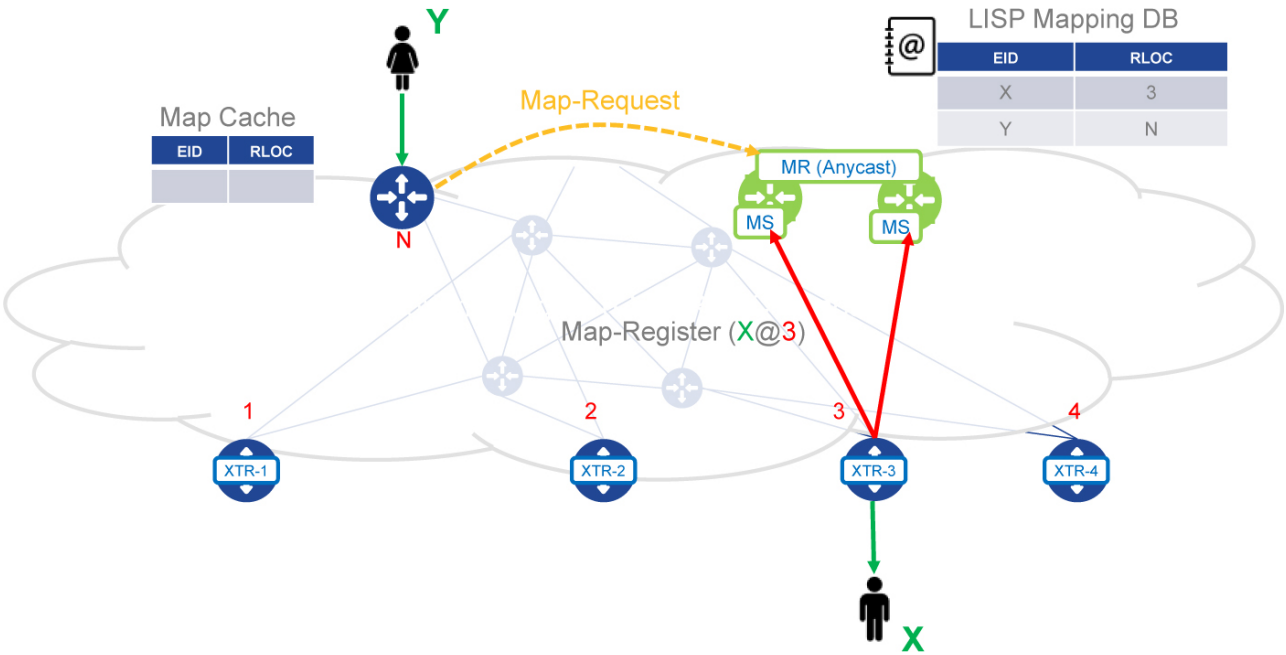


Figure 2-6 *Map-Server/Map-Resolver Resiliency*

> Note
>
> *An implementation could include database synchronization mechanisms and*

https://t.me/learningnets

Technet24

protocols among Map-Servers. This does not alter the way LISP works.

Multiple Map-Resolvers may share an anycast address. ITRs are configured to send their Map-Requests to this anycast address. The closest Map-Resolver to receive the Map-Request consumes the packet and services the Map-Request, providing Map-Resolver resiliency in a simple manner.

It is common to find Map-Resolver and Map-Server functionality co-located on the same device. In much of the literature, this combination is referred to as an *MS/MR*. Using the resiliency model just described, you can group MS/MRs to provide a resilient Mapping Database System node. Let's use the term *MS/MR node* to refer to a portion of the Mapping Database System that is authoritative for a finite set of EIDs.

LISP was originally conceived to address the scaling issues of the Internet. In such a role, a single node of MS/MR does not suffice. The Mapping Database System is designed to scale out by federating a multitude of MS/MR nodes. Different MS/MR nodes may handle different portions of the EID space. For instance, a large corporation may be organized in regions and have an MS/MR node deployed for each region. The MS/MR node for each region is authoritative for the mappings of the EIDs in the region as well as the LISP message exchange with the xTRs in the region. Distributing the EID mapping state in this way allows the Mapping Database System to scale in a virtually unlimited manner while providing adequate failure separation across the regions.

The separation of the MS and MR roles enables the distribution of the Mapping Database System across multiple MS/MR nodes while preserving the capability to communicate across the different regions that the different MS/MR nodes service. For instance, an MR in one region forwards Map-Requests to an MS in a separate region to resolve EIDs handled by that remote MS. This implies that the MS/MR nodes are part of some sort of referral system that allows an MR to determine which MS may be authoritative for a particular EID.

In the early days of LISP, MS/MR nodes exchanged EID information using BGP in what was known as the ALT topology (the Alternate topology). The challenge with this approach was that the ALT topology inherited the benefits and limitations of a traditional routed network. It became evident relatively quickly that a different mechanism for the federation of the MS/MR nodes was required to support an extensible EID namespace inclusive of segmentation and other semantics. The LISP working group at the IETF proposed the use of a Delegated Database Tree (DDT), which provides a tree structure that is traversed in a conceptually similar way to a DNS

https://t.me/learningnets

tree using iterative versus recursive lookups. The DDT is structured in a hierarchical manner with the leaves of the tree being the MS/MR nodes described so far. DDT introduces the concept of DDT nodes that form the branches and root of the hierarchical tree. Not surprisingly, there is a notion of a root DDT node, as shown in Figure 2-7. DDT also introduces a new LISP message known as a Map-Referral that is exchanged among DDT nodes to enable the navigation of the tree. When an MR needs to resolve an EID, the MR sends the Map-Request up the tree; this action then triggers an iterative process of Map-Referrals up and down the tree until the authoritative MS for the EID in question is identified. The MR receives a Map-Referral informing it where to forward the Map-Request.

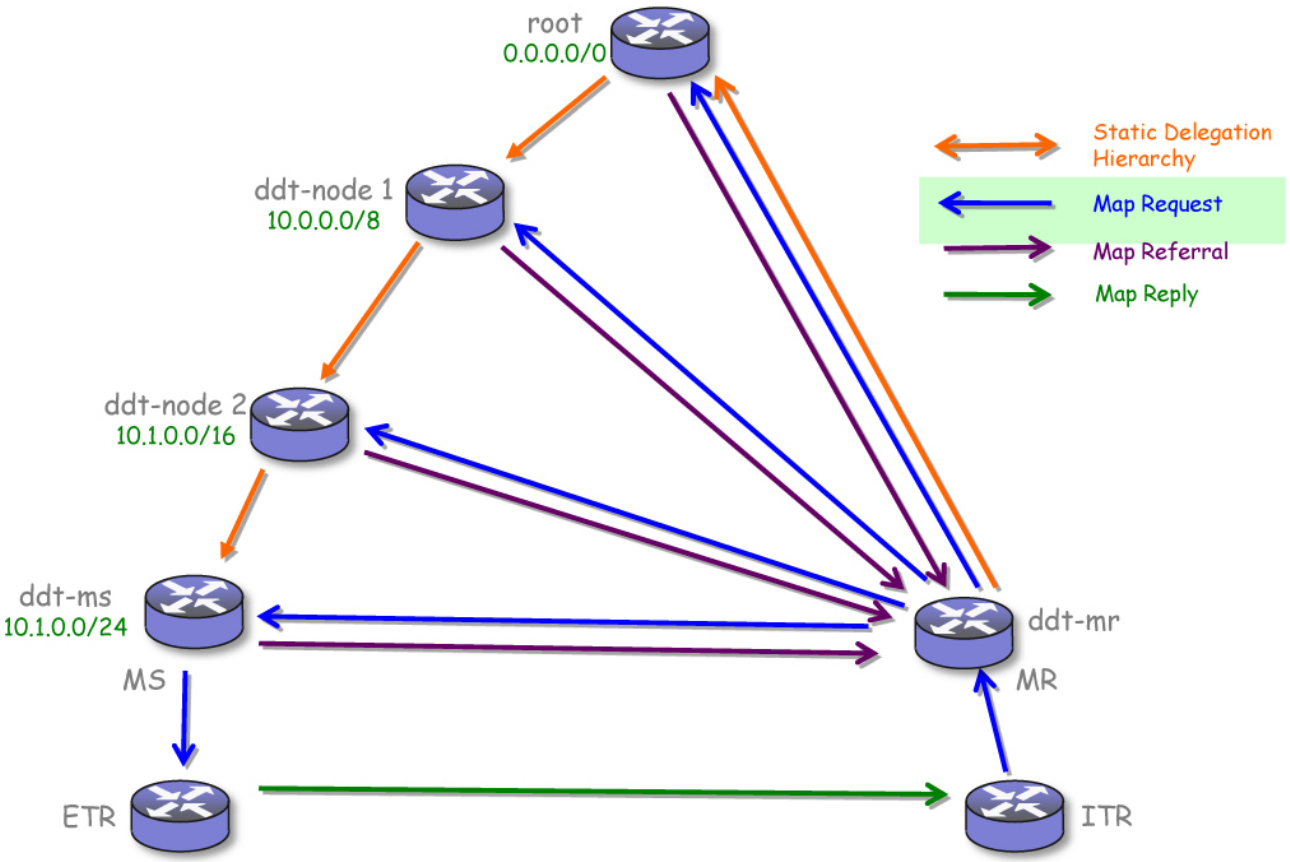


Figure 2-7 *Delegated Database Tree (DDT)*

DDT is independent of the EID structure, and although it could be organized around subnet summarization boundaries, it is often organized around other attributes of the EID space, such as the segment instance. As you can see, LISP handles information in a topology-independent manner and should not be subject to the limitations that topology awareness imposes on traditional routing protocols. Hence, you should not think of LISP as a routing protocol, but as an identity directory where the semantics of the identity are flexible and the scale of the directory is unlimited.

AN ASSET-CONTROLLED MAPPING DATABASE

The LISP mapping database naturally embodies the declarative model of object

https://t.me/learningnets
Technet24

interrelations that is at the heart of modern large-scale software architectures. In a declarative object model, agents are trusted to work autonomously, and they declare what they are willing and able to do instead of receiving a directive from the top-down stating what is expected from them. This capability leads to a scalable distribution of tasks and responsibilities in modern software architectures in which intelligence is a distributed attribute and trust relationships between agents enable the creation of efficient systems.

The concept of intent and declaration is intuitive and not exclusive to software architectures. Much of the social network (no, not the website where you post photos and comments) around which our lives are structured is based on declaration and trust. Simple interactions, such as having a neighbor water your plants while you are out of town, follow a model of declaration of intent. The neighbor declares an intent to water the plants every other day while you are gone and also declares a preference to do this in the mornings and skip the weekend. You, in turn trust, your neighbor to do as promised. There is an implicit assumption in this trust that the neighbor will do his best to take care of the plants and improve his own process of watering and care without your having to know or worry about how he goes on about things. Trusting the neighbor to be competent is important to the success of the system in making society scalable and productive. You, in turn, may be establishing other trust relationships and declaring intent to fulfill other tasks while you travel predicated on the time made available to you thanks to your neighbor's intent to attend to your domestic business.

In LISP, the ETRs are autonomous and trusted. The ETRs declare their intent to deliver traffic to certain EIDs by registering the EID mappings with the Mapping Database System.

> Note
>
> *In the LISP system, the establishment of this trust relationship is allowed within the scope of a policy that states which EIDs a specific ETR is expected to be authoritative for.*

In declaring their intent to forward traffic to the EIDs for which they are authoritative, the ETRs also declare their preferences in terms of how they'd like the traffic to reach them. LISP locators have associated priority and weight parameters that are set by the ETRs. The ETRs are trusted to the extent that they actually control the mapping

https://t.me/learningnets

database through the process of declaration of intent and preference. The ETRs are therefore assets that are listed in the database, and they actively participate in the database and control it by being the authoritative source of information for the database.

The distribution of tasks and state that results from this declarative model is instrumental in making the LISP architecture scalable, efficient, self-documenting, and, to an extent, self-healing. There is also a rather profound operational implication in that administrators of different ETRs can use declarative models to interact with each other and to model their communication policies. Declarative models are critical to the federation and agile definition of communication policies across administrative domains. The declarative nature of the Mapping Database System provides the necessary data model to enable the clean development of programmatic RESTful interfaces for communication with the LISP Mapping Database System and for communication between LISP administrative domains.

NETWORKING BEYOND TRADITIONAL ADDRESS TYPES

So far we've discussed EIDs and their mappings in the context of network addresses, but we can define EIDs more broadly to encompass much more than what a network address traditionally conveys.

In the mail system analogy, a person's identity may be qualified beyond a name to include details such as whether this person should be contacted for work purposes versus personal purposes, or how this person may be contacted after hours or during working hours. There could be further refinement of the identity specifying how to contact the person if the requestor is within the country or if someone is trying to reach the person from abroad. So, the directory may be able to provide the address for an individual based on a more detailed specification of the identity. The notion of identity is therefore extensible, allowing the directory to provide location information in the context of the intended communication policy. Furthermore, the directory may provide information in addition to the street address of the receiver; the obvious example is the phone number of the receiver in addition to street address. Thus, both identity and location naturally have extensible semantics.

In the networking context, the definition of EIDs and RLOCs quickly expands from traditional network addresses to more general data structures capable of incorporating rich information to define communication policy. For example, the EID may be expanded to include information such as the role of the EID (work or personal), a grouping of EIDs, or the time of day. An RLOC, in turn, may be expanded to include

https://t.me/learningnets

grouping information or even geo-coordinates.

The way these extensible types are encoded in the LISP control plane is defined by the LISP Canonical Address Format (LCAF) where a multitude of types is defined to allow the system to operate beyond the traditional network address types of IPv4, IPv6, and Ethernet to include much more flexible semantics capable of encoding composite names or even accommodate for previously undefined namespace types.

The potential of this extensibility is exploited immediately in the context of software-defined networking (SDN)–based applications. For instance, some applications leverage the encoding of geo-coordinates in the RLOC space to leverage the information in the network to enable location-tracking applications for entities that roam around a LISP-enabled network. In this example, the application queries the database for a particular EID, and the Mapping Database System replies to these queries with the IP addresses of the current RLOCs in the mapping. The reply also includes the geo-coordinates of the RLOCs, providing the application with coordinate information it would traditionally have procured from other sources but not from the network.

In looking at what the future may hold, notable research activity is a good indicator of where things may lead. There are efforts to formalize the notion of handling information in the network and make it a core element in the foundation of the Internet. One example is the Information-Centric Networking research group (icnrg) at the Internet Research Task Force (IRTF, the sister research branch of the IETF). This research group is looking at the implications of moving the focus of the Internet from nodes to information objects, which is at a high level moving from networking on IP addresses to names. The group centers its analysis on the use of name-based routing. The motivation and implications behind the work in this group are in line with the motivations and implications behind providing a network directory service capable of supporting extensible address types for both identity and location.

THE LISP DATA PLANE

LISP provides a data plane designed to enable optimal correlation of underlay and overlay information to aid the LISP overlay in making the best use of the underlying transport network.

A LISP ITR encapsulates the payload received from the EID space in an IP UDP header with source and destination addresses in the RLOC space referred to as the *outer-header*. The original header of the payload is preserved and is referred to as the *inner-header*. Between the outer UDP header and the inner payload header, a LISP shim

https://t.me/learningnets

header is included to encode information necessary to enable the forwarding plane functionality relevant to the use of an overlay.

The LISP data plane is designed to enable the following functionality:

• Tunnel entropy

• Segmentation

• Locator status validation

• Path reliability

• Confidentiality

• Authentication

Tunnel Entropy

When traffic is tunneled between an ITR and an ETR, the information in the outer-header of the tunnel may be the same for all flows between a specific ITR/ETR pair. Thus, many different flows may use identical outer-headers and therefore all be hashed to a single path in the underlying transport network regardless of the existence of other paths that could be used to balance the load in the transport network. Tunnel entropy refers to the ability to add entropy to the information in the outer-header so that different flows may be hashed to different paths and avoid the polarization of the tunnel to a single path. In the LISP data plane, this is achieved by using different source UDP port numbers in the outer-header for different flows in the payload. This way, different flows between the same source and destination locations use similar outer-headers except for the source UDP port that is different for different flows.

Segmentation

The ability to create a multitude of separate network contexts on a single network infrastructure is a common requirement in many environments. Generally, these separate contexts are realized in the form of a Virtual Private Network (VPN). Whether a service provider needs to deliver a VPN service to many independent customers or an enterprise maintains different parts of the organization segmented in separate virtual networks, network segmentation is a common requirement.

LISP EIDs and their mappings can be scoped in forwarding contexts such as VRFs or VLANs. To identify an EID as a member of a particular context, the LISP architecture

https://t.me/learningnets

Technet24

includes a context identifier known as an *Instance-ID*. The Instance-ID is coupled with the EID to expand the semantics of the EID to reflect the scope of a virtual network. The Instance-ID for a particular virtual network or segment is encoded in LCAF for the control plane to be able to do lookups and populate mappings in the correct virtual network context. The Instance-ID must also be included in the data plane so that forwarding decisions are made in the right virtual network context. For instance, when an ETR receives encapsulated traffic, it looks for the Instance-ID in the data plane to determine which VRF or VLAN to use to forward the received traffic after it de-encapsulates the traffic.

Instance-IDs are encoded in the LISP header, as shown in Figure 2-8. A flag is set in the header to indicate that the Instance-ID is present. When the flag is set, 24 bits are allocated for the encoding of the Instance-ID in the LISP header. The use of 24 bits enables a namespace slightly larger than 16 million identifiers. This provides a reasonable amount of flexibility in handling the segmentation namespace.

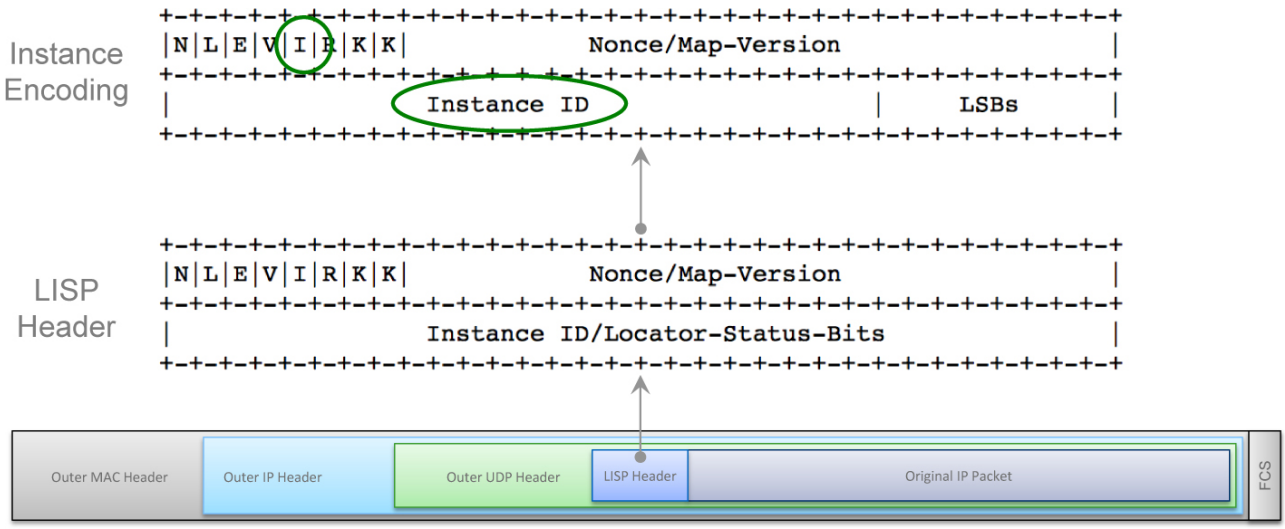


Figure 2-8 *LISP Header with Instance-ID*

Locator Status Validation

Because the LISP overlay control plane is independent of the underlay data plane, the LISP control plane does not convey information regarding the availability of remote RLOCs. Remember, LISP is not a routing protocol. Based on the mappings provided by LISP, an ITR may attempt to encapsulate traffic to an RLOC that is not reachable in the underlay. This may happen if the underlay routing hasn't converged or simply doesn't reflect the status of the specific RLOC due to summarization in the underlay. Nevertheless, LISP includes RLOC reachability mechanisms in its data plane that prevent an ITR from encapsulating to an unreachable RLOC.

So that you get a sense of the status of a remote RLOC, the LISP data plane includes a

https://t.me/learningnets

series of locator status bits in its header. Each bit represents the status of an RLOC in the local site. A bit set to 1 indicates the RLOC corresponding to that bit is up, and a bit set to zero indicates the corresponding RLOC is down. When an ITR encapsulates traffic, it sets the locator status bits according to the state of the RLOCs in its site. Assuming xTRs operate as both ITRs and ETRs, the ETR receives the locator status bits and uses this information when it acts as an ITR for the return traffic and decides whether an RLOC should be used or not to encapsulate return traffic. To set the bits, an ITR can infer the status of RLOCs in its site by local inspection of its interfaces and routing table.

The location of the locator status bits in the LISP header is shown in Figures 2-8 and 2-9. When the Instance-ID is present, 8 bits are available; when the Instance-ID is absent, 32 bits may be used to reflect the state of up to 32 RLOCs at the site. Each RLOC is assigned an ordinal between 0 and $n - 1$; the locator-status-bits are also numbered from 0 to $n - 1$ from the least significant bit in the field, where n is the number of RLOCs. The numbering of the RLOCs and LSBs is aligned to uniquely identify the state of a particular RLOC.

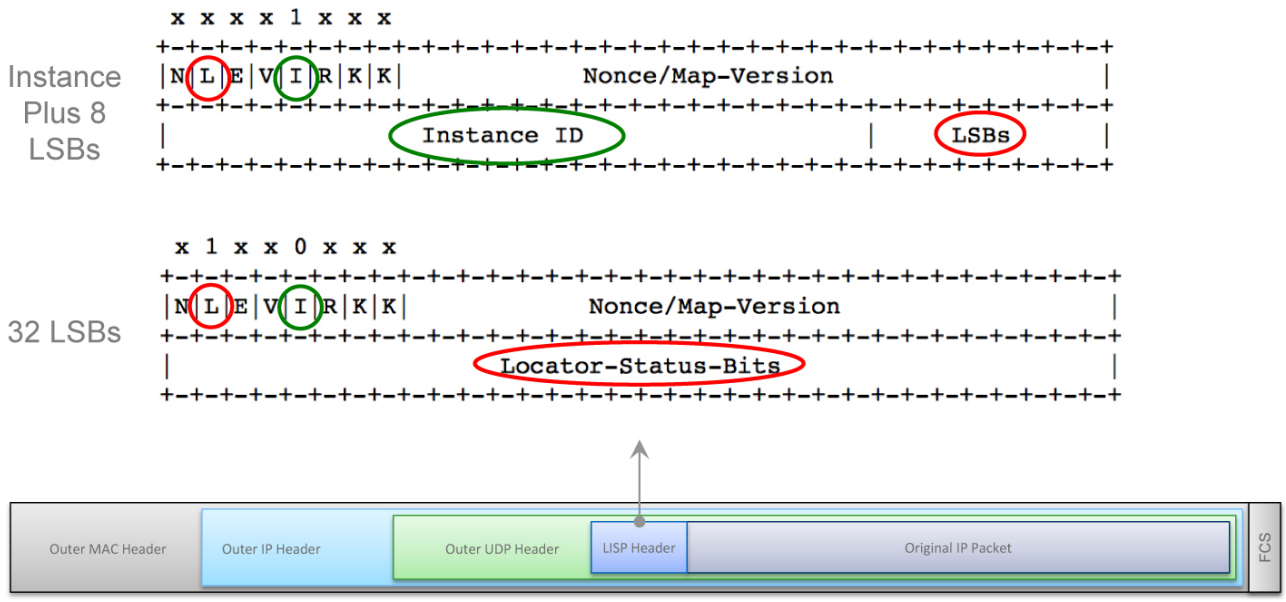


Figure 2-9 *LISP Header Locator Status Bits*

Path Reliability

The LISP data plane includes mechanisms to verify the integrity of the connection. The LISP data plane includes a nonce field that can be used to send a nonce and to verify that the return traffic can echo back the correct nonce. You can see the location of the nonce field in Figure 2-9; two corresponding flags (N and E) indicate whether the nonce is present and whether it is an original nonce or an echo. The LISP control plane also includes mechanisms to verify the reliability of a path. The mechanism is referred

https://t.me/learningnets

Technet24

to as *RLOC-probing*. When RLOC-probing, the ITR issues a Map-Request, with the probe flag set, for a particular EID. The Map-Request is sent directly to the RLOC(s) present in the Map-Cache for the specific EID. The ETR that hosts the RLOCs receives the Map-Request messages, verifies whether it can reach the EID, and sends a Map-Reply (with the probe flag set) to the probing ITR. The reply reflects both the most current mappings as well as the ETR's capability to reach the EID. Lack of a reply indicates a connectivity problem in the path to the RLOC; a reply with the R-bit clear for a particular RLOC indicates that the RLOC is reachable but the ETR cannot reach the EID post de-encapsulation. Figure 2-10 shows the probe flag (P-bit) as well as the R-bit that are relevant to the Map-Requests and Map-Replies used in RLOC probing. If you want more details on these mechanisms, look at the RLOC-probing algorithm definition in RFC6830bis and the specification for the control plane messages in RFC6833bis.

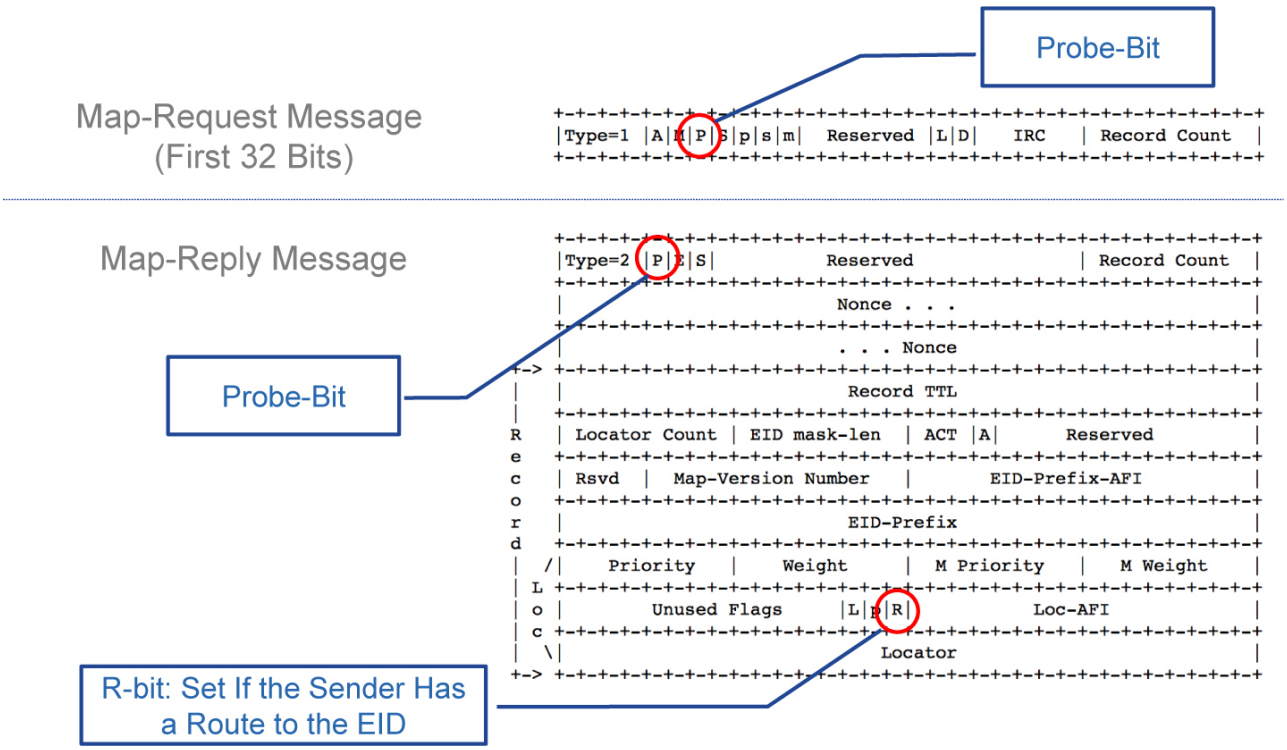


Figure 2-10 *LISP Message Fields Relevant to RLOC Probes*

The RLOC probing mechanism is robust and reliable; it handles failure scenarios within the underlay and also beyond the de-encapsulating ETR. The mechanism even includes rough round-trip time (RTT) estimates between locators, which is used as input for network management or performance-based traffic optimization. This versatility does come at the cost of bandwidth and processing cycles on the xTRs. In some scenarios, for the mechanism to be effective, the frequency of probes need to be high, which significantly increases the cost of processing and bandwidth.

Confidentiality and Authentication

https://t.me/learningnets

Encapsulated packets are encrypted by ITRs before encapsulation and decrypted by ETRs after de-encapsulation to provide privacy and confidentiality. Each ITR/ETR pair from source LISP-site to destination LISP-site use different keys, so they have pairwise security. Rekeying is exercised at high frequency while the packet stream stays secure.

LISP uses the latest ciphers that cryptography has to offer so that authenticated encryption can be performed. Therefore, when an ITR encrypts, the ETR knows the ITR is authenticated through the key exchange procedure performed by the LISP control plane.

Alternative Data Plane Formats

The control and data planes in LISP are loosely coupled. In general, the LISP control plane is used with different encapsulations and delivers most of its value. Depending on the encapsulation used, some of the functionality in the LISP data plane may not be available. In certain applications, the LISP control plane is used when there are no data planes at all. It is used as an inventory control database as well as an access control database.

One popular encapsulation supported in a wide range of switching ASICs is VXLAN. Some implementations of LISP use the VXLAN encapsulation in lieu of the LISP encapsulation, as already discussed. These implementations use the full functionality of the LISP control plane and compromise on some of the benefits of a full LISP data plane.

The VXLAN encapsulation is similar to the LISP encapsulation. Both data plane protocols encapsulate their payload in an outer UDP header, and the shim header of both LISP and VXLAN has similar flags and fields that are bit-by-bit compatible with each other. VXLAN could be seen as a subset of the LISP encapsulation. The similarities between the LISP and VXLAN encapsulations should not come as a surprise because the VXLAN specification evolved from the original IETF specification for Layer 2 LISP.

When you use VXLAN encapsulation, entropy and segmentation continue to be supported, but the semantics for locator-status, path reliability, and integrated cryptography are lost. Figure 2-11 shows the VXLAN header in contrast with the LISP header; note that the instance flag as well as the virtual network identifier (VNI) are in the same position as the instance flag and Instance-ID field in the LISP header. Also note that the flags are in the same position, but all flags in the VXLAN specification remain reserved, along with the bits in the spaces that correspond to the nonce and locator-status-bits fields in the LISP header. One way of looking at the VXLAN header

https://t.me/learningnets

Technet24

is that it is a LISP header with the nonce and locator-status-bits fields disabled. The VXLAN data plane encapsulation supports both L2 and L3 overlays with the same UDP port number, much like the LISP data plane can support both L2 and L3 overlays using different UDP port numbers for each service.

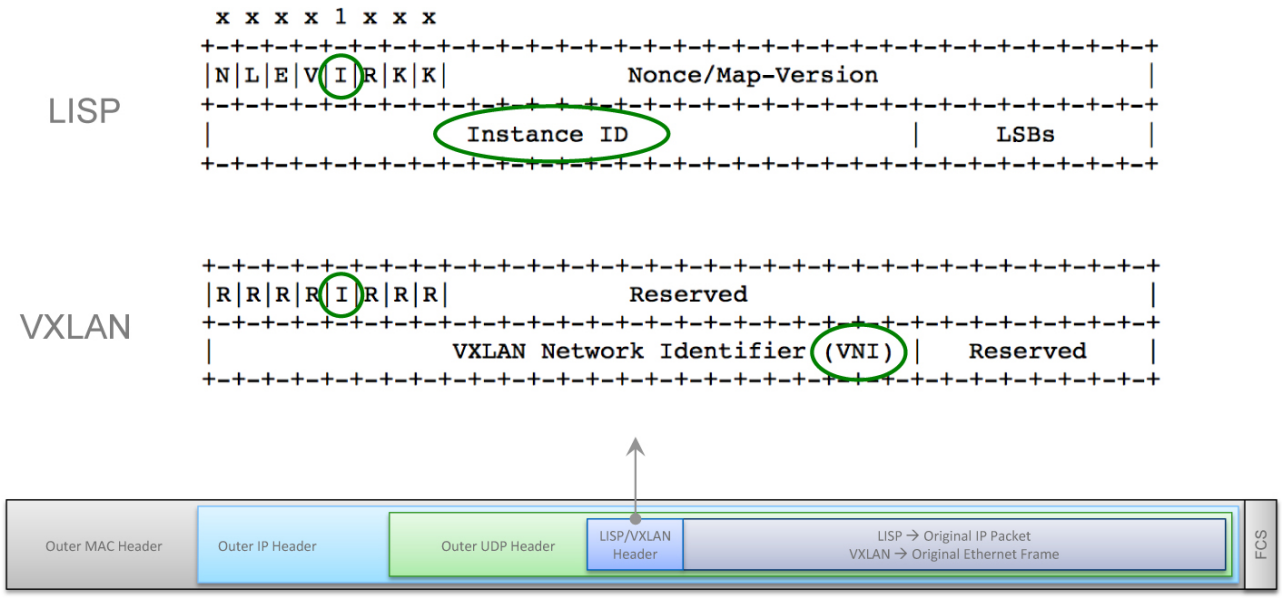


Figure 2-11 *LISP and VXLAN Headers Compared*

Some implementations include policy metadata in the VXLAN header by using some of the reserved bits to encode the additional metadata. Switching fabric implementations from Cisco, such as ACI, use a version of the VXLAN encapsulation that carries a 16-bit group tag in lieu of the LISP nonce. The extensions to the VXLAN header are documented in draft-smith-vxlan-group-policy. The group tag is used for the enforcement of group-based policies in fabrics such as the Application Centric Infrastructure (ACI). In this case, the nonce bit is set, and the 16 bits fall in line with the 16 bits originally intended for the nonce.

It is arguable how much metadata is actually required to be carried in the data plane. When a demand-based name resolution system such as LISP is in place, forwarding could be designed in such a way that metadata in the data plane may be of limited value because the metadata could be provided by the control plane at all relevant hops in the network. In any case, efforts are underway to generalize the mechanisms to provide extensible metadata in the data plane by introducing the network service header (NSH). This optional header allows the encoding of metadata without using the limited reserved space in the overlay headers.

An alternative use of the reserved header space is the inclusion of a protocol type to generalize the encapsulation to any type of payload. As of this writing, the LISP header assumes an IP payload follows, whereas the VXLAN header assumes an Ethernet payload. Neither header format has a next-protocol-type field. The Generic Protocol

https://t.me/learningnets

Encapsulation (GPE) specification devotes some of the reserved bits in the VXLAN header to provide a protocol field that allows a single header definition for L2 or L3 payloads. This is an effort to make the LISP and VXLAN headers as similar as possible and eventually converge in the use of a single header for all applications.

In spite of the benefits of the LISP header, there are many competing header proposals in the industry. VXLAN seems to have secured a broad footprint in ASIC implementations, and other options have also been hardware accelerated. Some examples include efforts such as Geneve. As this space evolves, the LISP control plane can leverage any of these data plane variations.

NAT TRAVERSAL

In many cases, especially when the RLOC space is IPv4, an xTR has RLOC addresses that are private. These addresses generally have to be translated by a Network Address Translation (NAT) device to achieve connectivity beyond the private address space. The use of Network Address Translation poses a challenge in LISP because ETRs are generally unaware of whether they are behind a NAT device or not. Because an ETR doesn't know whether it is behind a NAT, it may register its EIDs with private RLOCs that are not globally reachable.

For LISP to successfully function in an RLOC environment where Network Address Translation is at play, a handful of things need to happen:

• An xTR must determine whether it is behind a NAT.

• If an xTR is behind a NAT, any EIDs registered by that xTR must be registered using the global/translated addresses for its RLOCs.

• Forwarding state needs to be created in the NAT and the LISP data plane.

Note

EIDs can be private addresses. When they are, they must be registered within a VPN. That means the sites that use the private addresses must be LISP sites and can only talk to each other. When a privately addressed LISP site wants to talk to a host outside its VPN, it may need to have its address translated. Thus, if a LISP site is talking to a non-LISP site and the non-LISP site uses global addresses, a packet sent from the LISP site with a private address must be translated first and then encapsulated next. This function happens when the NAT and xTR are co-located and the NAT function

https://t.me/learningnets

happens first.

In the mail system analogy, the use of NAT in the RLOC space is similar to using a corporate address with mailstops to send mail to employees within a large corporate campus. The mailstops basically identify the building and floor within the campus. All mail is addressed to the main corporate address. For example, if you were to send mail to an employee at Cisco, you would send it to 170 West Tasman Drive, San Jose, CA 95134. You would further specify the mailstop for the recipient; for example, if someone's office is on the second floor of building 7, it would be SJ07/2. In this example, the corporate address is equivalent to the Global RLOC, and the mailstop is equivalent to the port number for a specific destination. The real RLOC is actually 425 East Tasman Drive, San Jose, CA 95134, and you could choose to specify the second floor as part of the address.

When you give your address to a sender or register it in a directory, you would give the main corporate address plus a mailstop; this is what a registering ETR must be able to do in the LISP system.

LISP NAT-traversal can support the following scenarios:

• An xTR behind a single NAT

• An xTR multihomed across multiple NATs

• Multiple xTRs supported behind a single NAT

All these scenarios interoperate with each other as well as with sites that are not behind NATs.

SUMMARY

LISP enables an extensible and highly scalable directory of endpoints. Although this way of thinking about a protocol is counterintuitive at first, LISP is not a routing protocol but a directory service. The LISP architecture follows a demand-based model similar to the Domain Name System and shares many of the scalability characteristics of DNS. The LISP architecture is succinct and simple, yet highly extensible. This extensibility is key in the enablement of advanced network functionality and sets LISP apart from traditional routing protocols, enabling network-centered services that would not be possible otherwise.

https://t.me/learningnets

Chapter 3. Data Center Trends

The infrastructure of the data center has evolved at a rapid pace in recent years. Compute management and orchestration have developed into a flexible and open ecosystem of functions that support the agile consumption of the compute infrastructure that powers large-scale cloud computing. At the heart of the agility provided in the data center is the network. The principles of separating location from identity have proven crucial to enabling the malleability and agility required from the network to support the requirements of modern cloud computing.

This chapter discusses the predominant trends of the data center and the role of LISP in enabling these trends. It examines how LISP and the revolutionary concepts introduced throughout its development have played a pivotal role in the evolution of the connectivity required in data centers to date. It also discusses the potential role of LISP moving forward.

A BRIEF HISTORY OF APPLICATION VIRTUALIZATION

The origins of operating system virtualization date back to the 1960s and 1970s. This was the age of the mainframe, and access to compute resources was not only limited but also a serial activity in which only one user at a time could access computers to run batch jobs. The benefits of high-capacity computation were evident in many circles, and demand for computing was on the rise, particularly in the research community; however, owning a dedicated computer was not viable for most people. The compute resource had to be shared.

On July 1, 1963, the Massachusetts Institute of Technology (MIT) announced project MAC. Project MAC was initially funded with a research grant from the Defense Advanced Research Projects Agency (DARPA) with the goal of advancing research in the areas of operating systems, artificial intelligence, and computational theory. The acronym *MAC* originally stood for Mathematics and Computation. Depending on individual vantage points, the project was renamed to Multiple Access Computer, Machine Aided Cognitions, or Man and Computer. The inception of project MAC at MIT, plus strong requirements from Bell Labs for time-shared access to compute

https://t.me/learningnets

Technet24

resources, triggered a series of events, including General Electric's commitment to build a time-sharing computer for MIT. Multiple users could access the computer at once and were allocated a slice of memory, CPU, and other system resources. The operating system handled the scheduling of these resources; the operating system used at the time was MultiCS. The experience was that of being able to run processes in a shared environment, with all the exposure to fate sharing and security vulnerability that such sharing implied. MultiCS was developed as part of project MAC and was further developed by Bell Labs into what eventually became UNIX.

Incidentally, professor John McCarthy was a member of the distinguished faculty at the Artificial Intelligence group within project MAC. Doctor McCarthy invented the Lisp programming language in 1958. McCarthy's Lisp family of programming languages is very different from the LISP networking protocol this book is about, but one could argue that both concepts are anchored on a notion of recursive operations and indirection.

IBM eventually responded to the requirement for shared computation with the release of the CP-67 system, which was the first commercially available mainframe to support virtualization. The system used the CP/CMS operating system. The Console Monitor System (CMS) was a small single-user interactive OS. It was coupled with the Control Program (CP), which was in charge of creating virtual machines. This was a big leap from what had been previously achieved with time sharing because now users had their own complete operating system (for them to run or crash without affecting other users), and the resources of the mainframe were more efficiently shared across virtual machines rather than statically split between users. CP/CMS was released in 1968 but did not get to a point of stability until 1972.

The virtual machine model that IBM introduced in the 1960s was similar to what is known today as virtual desktop infrastructure (VDI). In a VDI, each user has a dedicated operating system hosted centrally on a shared compute infrastructure and can access it remotely.

Following the introduction of virtual machines on mainframes, a lengthy process of decentralization of compute followed. This process was fueled largely by the increase in processing power and versatility of the personal computer. For some time, few applications required the processing power of the centralized mainframe, and more and more applications were run on dedicated hardware on personal computers. Applications got more sophisticated over time, of course, and more and more sensitive data was being handled digitally. Client/server applications required more and more centralized processing power at the data center, yet there was so much processing

https://t.me/learningnets

power on smaller normalized x86 systems that a return to mainframes was not an obvious choice. Horizontal scaling of distributed compute resources became the norm in large data centers, and with it, the management challenges associated with a large number of servers running distributed applications became a key problem to solve.

Virtualization had been somewhat dormant until 2007 when a multitude of software vendors (led by VMware and Microsoft) released to the market powerful tools to manage large numbers of virtual machines. The ability to treat servers as virtual resources and manage them at large scale made the use of distributed computing in large data centers viable. In some cases, these machines were used to host the desktops for end users. In most cases, however, they were used to host a multitude of complex multitier applications, and the use of virtual machines was mainly geared toward enhancing the manageability of the compute environment at very large scale. In fact, a vast majority of virtual machines today run on a dedicated host, which is a long way from where virtualization started with time sharing.

Since the refocus of the industry on operating system virtualization, data centers continued to grow, and the optimization of resource utilization has been the object of much thought and investment. In the quest for this optimization, it is natural that system administrators sought a more effective way to deliver the manageability and flexibility provided by virtual machines. Containers have emerged recently as a viable option to address the scale, flexibility, and agility required in large-scale data centers supporting DevOps practices.

Containers operate higher up the stack than virtual machines. You can think of them as much lighter weight than virtual machines because a container includes only the services and resources relevant to the processes that run on the container—nothing more and nothing less. Think of containers as mission-oriented processing units. The origin of containers can be traced back to the late 1970s in UNIX primitives such as chroot that set the foundation for process isolation. Process isolation further evolved in the early 2000s when the concept of jails in Linux and FreeBSD was introduced as a mechanism to partition a computer system into several independent, smaller systems (jails) with the ability to assign an IP address for each system and configuration. Containers continued to evolve outside the limelight. For example, Linux containers (LXC) were introduced in 2008, marking the inception of the first comprehensive container management solution. However, it wasn't until 2013, when Docker introduced a comprehensive container management solution with a complete ecosystem, that containers became relevant to addressing the large-scale requirements of cloud data centers. Today cloud-ready applications are designed and written to run on a container infrastructure in which they can easily be coupled with micro-services.

https://t.me/learningnets

Both virtual machine and container management systems include some level of networking that is particularly focused on optimizing the connectivity of virtual machines or containers running on the same host. Most of these networking models are limited to providing the equivalent of an L2 bridge between the virtual machines or containers without much focus on external connectivity. Other networking models are more flexible, and although they may work well within a host where interprocess communication channels abound, they are not easily extensible outside the host. Interoperability and other considerations make operators fall back to well-understood networking models to integrate the virtual infrastructure to the network.

MULTITIERED APPLICATIONS, VIRTUALIZATION, AND THE NETWORK

So how does this evolution of application virtualization impact the network? At first glance, the implication to the network is simple yet challenging. Virtual machines and containers create a large number of IP endpoints that may be instantiated anywhere in the network. This increases the scale requirements on the network as well as the entropy of the addressing space that the network must handle because subnet boundaries become meaningless. Furthermore, containers can be instantiated and brought down very rapidly, which makes the challenge even more interesting because the endpoints are disseminated across the network, and they also appear, disappear, and move at fast rates. But does the network actually have to deal with the connectivity of all virtual machines and containers that conform an application? Let's look into this issue in detail.

The architecture of most applications is broken down into three basic functional tiers:

• User interface tier

• Main processing algorithmic tier (the actual application)

• Functional tier in charge of handling the application data

More sophisticated applications require large amounts of processing power in each tier, which is achieved by horizontally distributing the application tier across multiple processors. Additionally, it is a common practice to develop the application tiers in a generic way so that certain components are reused and repurposed across applications to accelerate development. As a result, applications have evolved over time from self-contained monolithic designs in which all their components were bundled together in a single process to a multiprocess modular structure that is organized in tiers. A multitier application is illustrated in Figure 3-1.

https://t.me/learningnets

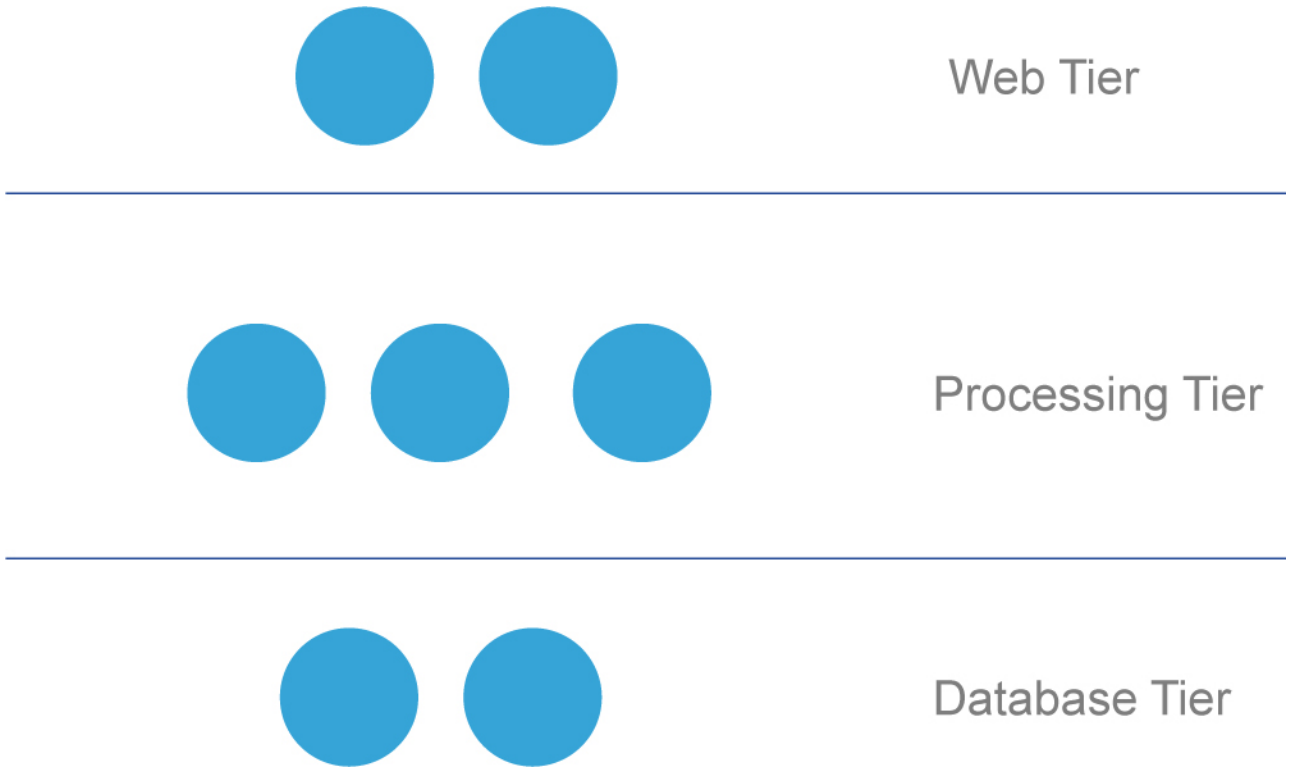


Figure 3-1 *Multitiered Application*

The processes in each tier are generally hosted on different servers. Therefore, the tiers of an application communicate with each other over the network. This intertier communication must be secured and optimized, so an integral part of an application stack is the services provided between tiers. The most widely used services are firewalls and load balancers. Firewalls secure the connections of the application, whereas the load balancers provide a key mechanism that many applications rely on to horizontally distribute workload processing within an application tier. Thus, firewalls, load balancers, and other network services have to be inserted between application tiers. For many years, and even today, the communication between tiers has used standard network mechanisms to steer traffic through the services between these tiers. In general, the mechanism used makes each application tier exist in one subnet and makes the network service node (firewall or load balancer) the gateway for the application subnet so that traffic to and from the application tier traverses the required network service (just as it traversed the router with the subnet's gateway). This mechanism for network service insertion is referred to as *VLAN stitching*. In VLAN stitching, the application servers are sandwiched between VLANs, and the VLANs are interconnected by the network services. Figure 3-2 illustrates a sample multitier application and how services have traditionally been inserted with VLAN stitching.

https://t.me/learningnets

Technet24

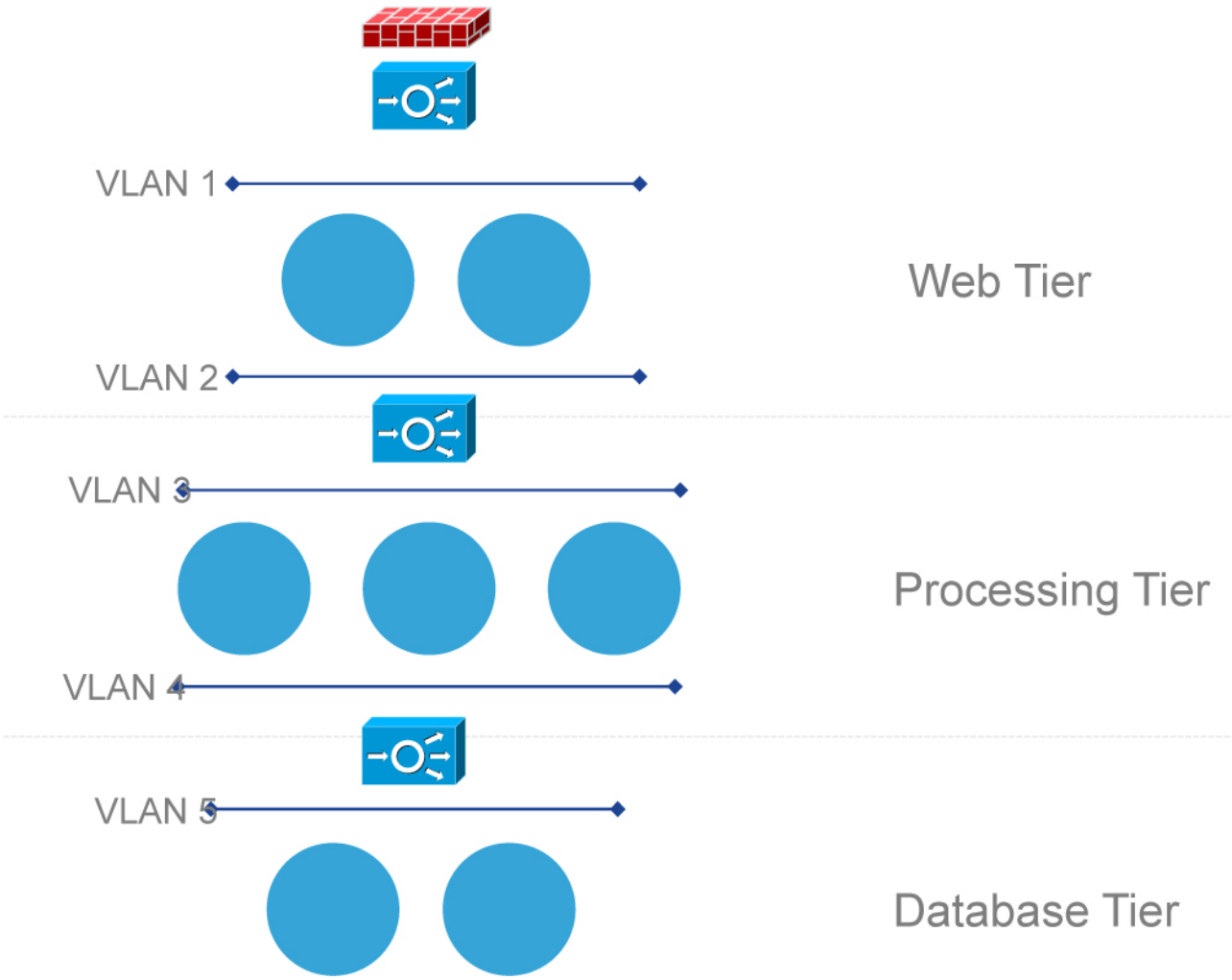


Figure 3-2 *Service Stitching in Multitiered Applications*

The subnet-centered structure of the multitiered application with VLAN-based service stitching made this a rigid and static structure. All VLANs, subnets, and port assignments had to be provisioned before the application was deployed onto the network. If the application nodes moved or new nodes had to be integrated into the application, the network had to be reprovisioned to accommodate the new structure of the application. The advent of agile application development practices, in support of business service velocity, meant that the manual process of provisioning servers, applications, and services in a rigid network structure had to be revisited. Around 2007, VMware, Microsoft, and the open-source community introduced new server virtualization hypervisors along with modern virtual machine management tools to optimize the use and management of the compute infrastructure. This renaissance of the use of compute virtualization marked the initial steps toward the creation of an infrastructure supportive of agile development practices and the start of the evolution in networking technology that has taken place since.

The renewed drive toward compute virtualization meant that full application multitier stacks could now be implemented with virtual machines. This included, in many cases, virtual instances of the network services used between the application tiers. In most

https://t.me/learningnets

cases, the virtual machines that made this application stack were distributed across multiple physical servers. The network had the responsibility of providing all the necessary VLANs, on the right ports, to enable the communication between the virtual machines in the application stack. The implication was that either the network became smart enough to automatically provision the right VLANs on the right ports as the applications changed, or the network could accommodate all VLANs on all ports. The industry chose to pursue the latter first, and the focus on networking shifted toward building data center networks to accommodate a large number of VLANs and MAC addresses. The vision was to create a virtual application stack and deploy it anywhere in the data center. The application stack was still based on VLAN stitching, and to deploy any application anywhere, the data center network needed to become a very large Layer 2 network. The experience had been partially improved from the compute management perspective, but the network was still a challenge. The fundamental network model in support of service stitching in multitier applications remained unchanged. The compute and service nodes in the multitiered application were virtualized, but the segments connecting the different nodes were not virtualized and remained rigid.

The need to make L2 networks more scalable and agile fueled a wave of network innovation that included technologies such as FabricPath, Trill, Overlay Transport Virtualization (OTV), and eventually virtual extensible LAN (VXLAN). VXLAN has an interesting story that is discussed later in this chapter. VXLAN became the response from the hypervisor vendors to virtualize the connections between the virtual nodes in the virtual application stacks. With VXLAN, the connections between the nodes became virtual wires that moved with the application anywhere in the data center. Furthermore, the hypervisor offered a virtual switch that created the VXLAN virtual segments directly between the compute nodes without any participation of the network, creating a host-based overlay network. The entire application stack was virtualized and made independent of the physical network. However, there was still much space for improvement. The fact that there wasn't a dependency on the physical network did not fundamentally simplify the problem of creating a network that supported the application stack. All that had been achieved was a complete virtual representation of the original multitier application stack, which was still predicated on segment stitching, although the segments were now virtual. Application developers still had to worry about putting nodes in the correct subnets and the virtual equivalent of the VLANs, which were the VXLAN virtual network instances (VNIs). It also became evident that the virtual application stacks required connectivity to the physical elements in the network because many application components were not virtualized. For the most part, network services like firewalls and load balancers remained physical, and the access to the applications was from a nonvirtualized network. The development of mechanisms

https://t.me/learningnets

to integrate the virtual connectivity of virtual applications with overlays that are built on the physical network still continues as of this writing. So, although promising in principle, the virtual machine model failed to truly make the applications independent of the physical network and didn't fundamentally alter the nature of the network operations required to support application deployment.

Another trend that has taken strength in the application space, particularly in the cloud space, is the use of containers. In recent years, significant investment was made in providing adequate management and orchestration tools for containers. This investment fueled the containerization of the different application tiers as well as the services that interconnect the tiers. The implementation of services in containers is widely known as *micro-services*. Micro-services get their name from the fact that they are lightweight processes dedicated to a single application component and therefore hold limited state. In the container model, each tier of the multitier application has the necessary micro-service containers attached to it, usually co-located on the same compute node where the application container runs. In this environment, intertier communication behaves much more as API-based interprocess communication and less as network communication. The promise of cloud-hosted applications and the strong DevOps culture of the cloud providers have fueled a strong wave of container-centered application development along with the required micro-services. These applications are commonly known as *cloud-ready applications*, and absolutely all of its components run in containers. The communication between containers has the opportunity to be totally independent of the network in ways that the inter-virtual machine communication did not. Container infrastructure such as that provided by Docker allows any model of container networking to be included in the infrastructure in an open-source crowd-sourced model. Conceivably, a container networking stack based on LISP is a distinct possibility.

EVOLVING SWITCHING FABRICS

Regardless of whether applications use virtual machines, containers, or a combination of the two, the front end of the application must connect to the physical data center network. Even if the intertier application connectivity is handled by direct overlay connectivity between the hosts, the network is tasked with handling connectivity to the host endpoints that are exposed by the applications as their front end. In many virtual machine–based deployments, the network is also tasked with securing connectivity to virtual machines and stitching physical services between application tiers. So, the data center network must be able to manage the mobility, scale, services, and segmentation required by the front end of the applications and in many cases by the entire application stack. Switching fabrics are the networking response to these requirements

https://t.me/learningnets

of mobility, segmentation, service chaining, and scale.

At a high level, a switching fabric dynamically provisions virtual networks and automates the connection of host endpoints to these virtual networks. In a switching fabric, different types of policy are also provisioned and enforced dynamically as host endpoints connect, move, and disconnect. This behavior is the same behavior you would expect from a host overlay created directly among virtual machines or containers.

Switching fabrics use network-based overlays to deliver a large number of virtual networks. The indirection between identity and location used in network overlays is key in supporting the required host mobility in the fabric. LISP includes the necessary mechanisms to support fast mobility of hosts while keeping both the mapping database and the Map-Caches up to date as hosts move around the fabric. The overlay data plane headers are leveraged to carry the metadata necessary for the segmentation of the network as well as the metadata required to enforce security policies and define service chains.

LISP is particularly well suited for the role of the overlay control plane in a switching fabric. The overlay control plane has a simple mission: maintain the mappings of endpoint-identity to their location. In the overlay control plane, there isn't a calculation of best routes, redundant paths, or loop resolution. These are all routing tasks that are fully realized in the underlay but are not required in the overlay. Furthermore, it is desirable to extend the database of endpoint-identity and location mappings to include policy, geo-location, and other information that may be relevant to the endpoints. All in all, the overlay control plane is in charge of populating and updating a directory of information that is relevant to the endpoints. At no point does the overlay control plane execute routing calculations. Therefore, it makes sense to use a protocol designed to handle such a directory versus trying to extend a routing protocol to do this database handling. The requirements of the overlay control plane call for a directory service such as LISP while all routing intelligence remains in the underlay, where well-established routing protocols like Open Shortest Path First (OSPF) and Border Gateway Protocol (BGP) excel at optimizing the reachability between routing locators (RLOCs). One salient characteristic of LISP in this area is the fact that it is demand based, allowing network nodes to request information only when they need it, and allowing the network nodes to be specific about the information they request. This capability has obvious scalability benefits but also leads to an interesting notion of contextual mapping database lookups in which the response from the LISP Mapping Database System varies depending on the context of the requestor. For example, if a mapping for an endpoint is requested from one location versus another, the reply may be different in

https://t.me/learningnets

either case. This notion of contextual lookups is discussed later in this chapter in the section “Policy: The Network as an Enforcer.”

Due to operational familiarity, the industry’s first wave of fabric development gravitated toward the use of traditional routing protocols for the overlay control plane. One popular overlay being promoted in the data center uses BGP with the EVPN address family to create virtual networks with mobility. As you will learn throughout the book, LISP presents clear benefits when used to create overlays in fabrics, and many implementations are tapping into these benefits. Cisco Application Centric Infrastructure (ACI) uses a mechanism with demand properties closer to those provided by LISP, and Cisco offers switching fabric implementations for the access networks in campus and branch that are fully based on LISP. Some of the benefits achieved by the demand-based model include scale, high rate mobility, embedded policy, and the simplification of the network control plane.

OPTIMIZING CONNECTIVITY TO THE DATA CENTER WITH LISP

LISP is frequently used to steer traffic to and from the data centers. It is common practice to deploy data centers in pairs to provide resiliency. When data centers are deployed in pairs, both facilities are expected to actively handle client traffic, and application workloads are expected to move freely between the data centers. The same situation arises in hybrid cloud deployments where workloads are relocated between different data center facilities. Figure 3-3 illustrates common multi-data center scenarios. Note that disaster recovery facilities also require traffic to be routed to them.

https://t.me/learningnets

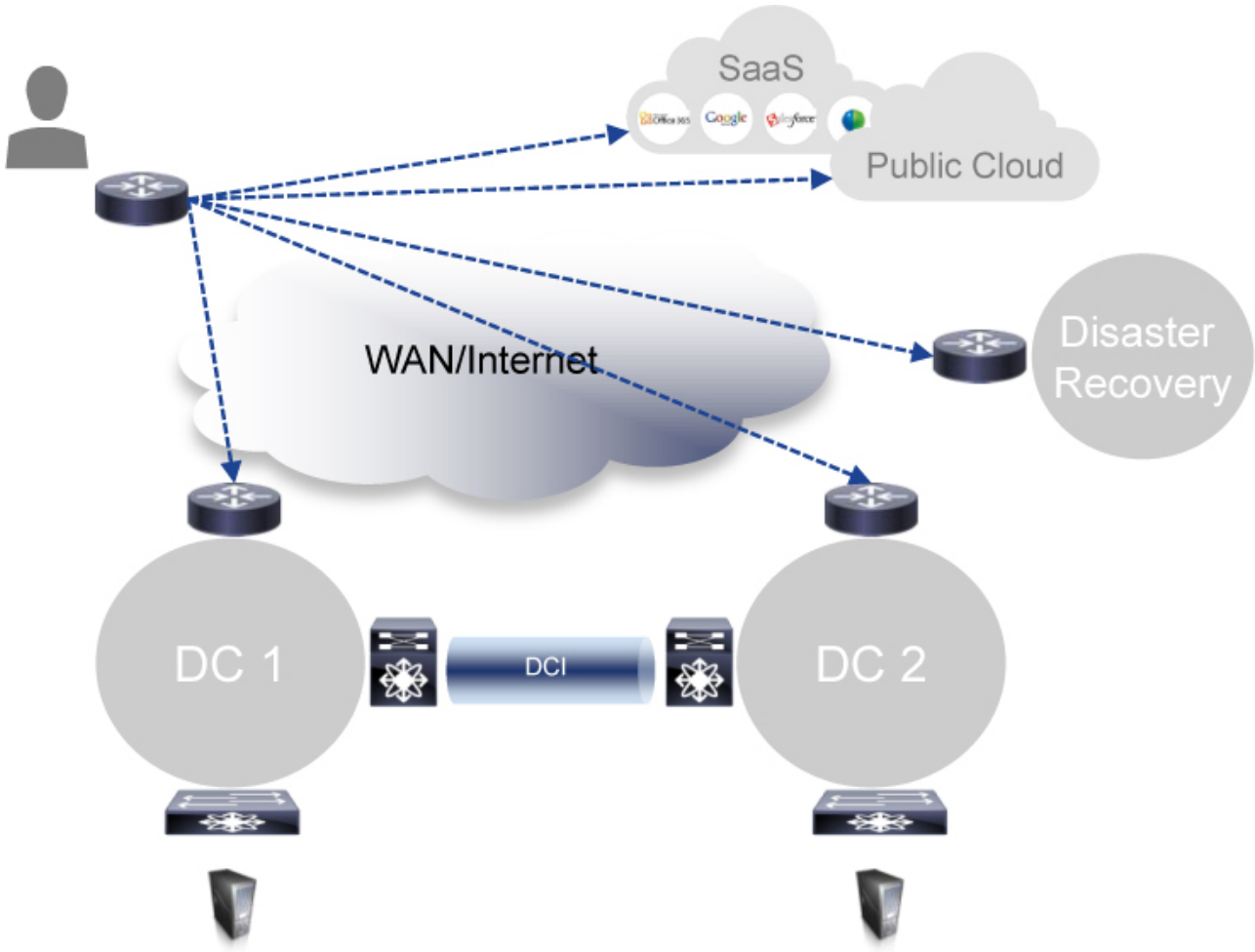


Figure 3-3 *Multi-Data Center Deployments*

Regardless of the type of multi-data center deployment, the IP address of the hosts for the applications does not change when the application workload is moved, so some hosts of a subnet are located in one data center and other hosts in another. This means that the IP subnet in which a host resides is no longer representative of one data center location or another. To maintain an acceptable application response time for users, the connection between the client and the application must follow the most direct path to the data center where the active host for the application resides. Figure 3-4 illustrates the optimal path to the active workload in contrast with suboptimal triangulated paths that result without host granularity in the routing.

https://t.me/learningnets

Technet24

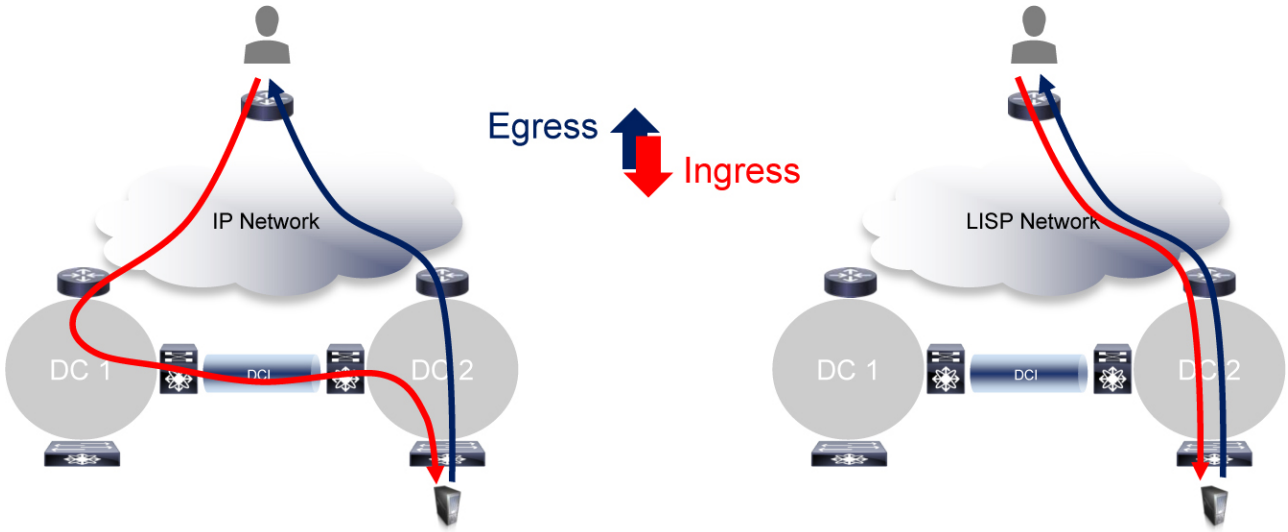


Figure 3-4 *Optimal versus Suboptimal Path to the Data Center*

The triangular pattern shown in Figure 3-4 assumes that the Layer 2 segments hosting the subnets are stretched across the different data centers. This configuration in itself poses an operational and design challenge that is addressed in the following section, where we make the case for mobility without subnet extension. Some network architects believe that the latency added by the triangular traffic path shown in Figure 3-4 may be negligible and therefore not worth addressing. In reality, the latency is the compound effect of the reachability at every layer of the application. Figure 3-5 shows the resulting traffic path when an entire application stack is not maintained in a common location. Figure 3-5 illustrates how the added latency from the triangular traffic path is multiplied. In a pair of data centers that are 10 ms apart, the latency impact on application response multiplies rapidly. This situation must be avoided, and to do so, you need to be able to move all members of an application stack to any data center without relying on triangulation to reach the workloads.

https://t.me/learningnets

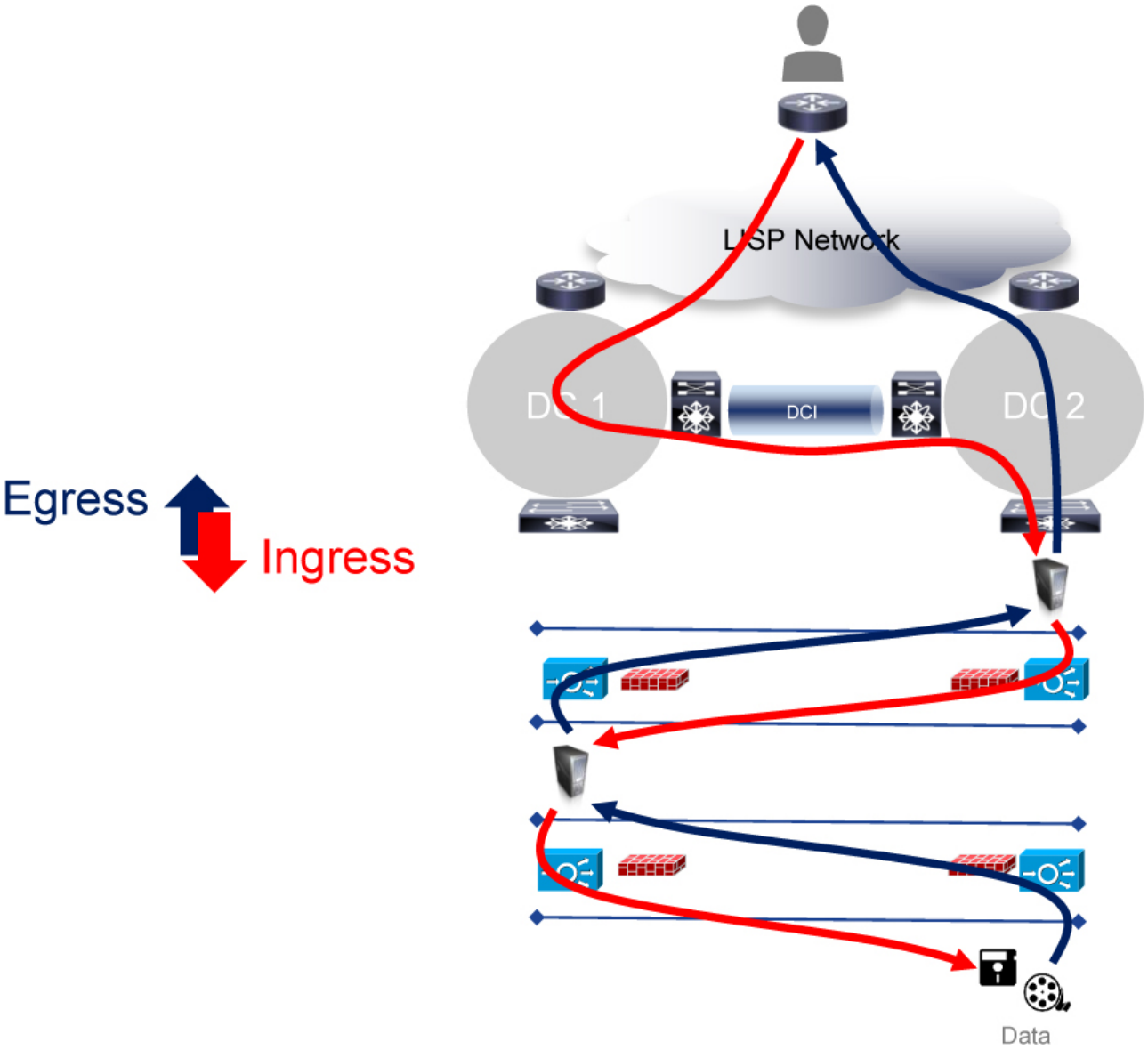


Figure 3-5 *Suboptimal Traffic Across Application Layers Between Data Center Facilities*

The implication is that traffic must be granularly routed to the different data centers based on host-specific location information. So, the question has been: how do you achieve host granularity in routing at scale? Traditional routing protocols require the dissemination of this host-level state across a large portion of the network, impacting scale and convergence severely. LISP provides the distinct advantage of handling granular host state for the applications in the data center by distributing the host state selectively only to the locations that need the state. By using LISP to handle host reachability outside the data center, address any scale and convergence concerns related to handling a large amount of host routes in the access and wide-area network while optimally routing to the data center where the application actually resides.

Some applications require very high availability. They are deployed in resilient data centers that are close enough to each other to allow the high availability models of the application to be deployed across the multiple data centers while adding resiliency to the network and facilities layer where the two data centers do not share any fate from

https://t.me/learningnets
Technet24

the networking perspective. As the active application instance moves from a host in one data center to a host in another, the sessions from the clients consuming the applications must be preserved before and after the move. So not only do you need to preserve IP addresses for the moving hosts, but you also need to provide consistent Layer 2 semantics across the different network locations so that the Address Resolution Protocol (ARP)/Neighbor Discovery (ND) state of the moving host remains valid after the move and connections are maintained. One simple way of doing this is to stretch the Layer 2 environments across the different locations, but let's look at how to avoid extending Layer 2 and simplify the way to handle moving hosts.

MOBILITY: SUBNETS REALLY DON'T WORK

As applications are migrated and load balanced within the data center or across multiple data centers, the IP address of the application hosts must be preserved. Thus, any IP address should be reachable anywhere it connects. With enough entropy, any IP address may show up anywhere, and IP addresses are no longer summarized into prefixes that align with the network topology. So, a switching fabric and the network connecting the clients to the data center must be able to handle a large amount of nonsummarizable host information. Furthermore, LISP allows the preservation of these IP addresses without requiring the extension of the Layer 2 domains that traditionally support the dissemination of these addresses to multiple locations.

The demand nature of LISP allows it to handle reachability state with host-level granularity while minimizing the required amount of forwarding table memory required on the LISP tunnel routers (xTRs). At the same time, the underlay remains stable and unaffected while all this mobility takes place. The large amounts of state required to handle mobility are maintained in the LISP Mapping Database System, which uses low-cost memory and scales horizontally as required. Forwarding memory at an xTR is used only to cache the reachability information relevant to flows established with its locally connected hosts.

To support mobility, a LISP-site must support the following:

• The moved host must be detected.

• The local first hop routing must be adjusted to handle the new location of the host.

• Registrations for the host in the LISP Mapping Database System must be updated with the new location of the host.

• The xTRs that have cached mappings for the endpoint identifier (EID) of the host

https://t.me/learningnets

must also be updated.

A moved host is detected through many mechanisms. In a LISP network, the local xTR must become aware of the move. When hosts are connected directly to the LISP xTRs, an xTR may glean data plane events and through a simple heuristic determine whether the packets correspond to a moved host. Alternatively, the host may be connected to a non-LISP fabric (for example, a BGP-EVPN fabric), and host routes may be reported to the LISP xTR that had not been previously reported to the particular xTR, thus indicating a move (or a new host). It is also possible to identify the mobility of a host by connecting to the APIs of the host orchestration software, which often signals the different stages of a host move.

When a host move is detected, the xTR registers the host with the LISP Mapping Database System. The LISP Mapping Database System also notifies the old xTR where the host used to be located. In any mobility solution, it is critical to signal the site from which the mobile host departed so that the site can clean up its forwarding tables accordingly and adjust its state to reflect the fact that a host that was otherwise local is now remote. In the particular case of LISP, the knowledge of the move at the old site is used to notify any sender who is still sending traffic to the old location that it needs to update its information. This is how LISP updates the Map-Cache of any active senders to the moving destination. In some of the existing implementations, the data traffic from the sending ingress tunnel router (ITR) triggers a solicit-map-request (SMR) message from the old egress tunnel router (ETR). This message tells the ITR to issue a new Map-Request for the destination and the lookup process eventually results in the refresh of the ITR's Map-Cache. Alternative mechanisms may be used in LISP implementations that allow subscriptions to mappings. In such systems, the Mapping Database System keeps a log of which ITRs are sending traffic to specific destinations and updates the ITRs if the registration for the destinations changes.

When a host move is detected, the xTR at the arrival site also installs a local route to the host so that traffic destined to the host and received over LISP is forwarded to the host after it is de-encapsulated. This route is a directly connected route or a route learned over a dynamic routing protocol, depending on whether the host is directly connected to the xTR or if the host was learned from an adjacent non-LISP fabric. Figure 3-6 illustrates this concept of directly connected hosts versus hosts connected to the xTR through a fabric.

https://t.me/learningnets

Technet24

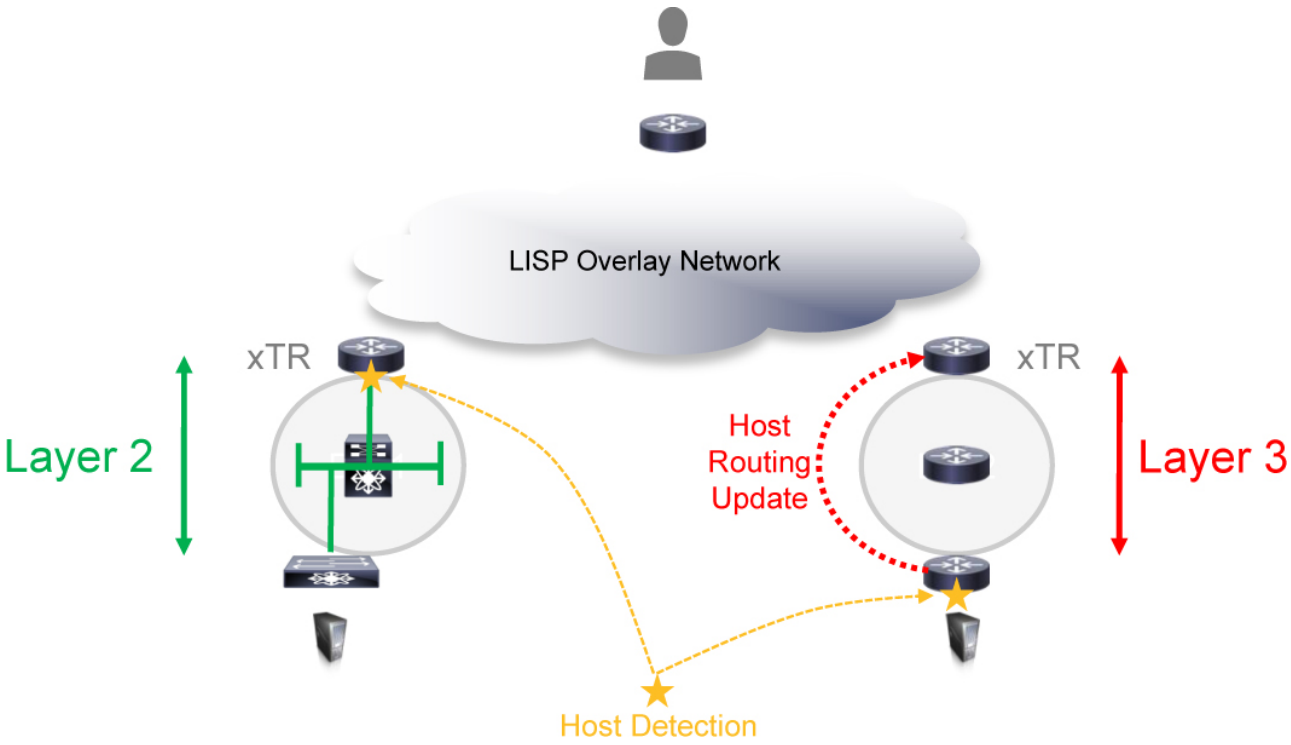


Figure 3-6 *Host to xTR Connectivity Options*

When the hosts are directly connected to the LISP xTR, the xTR may rely on a Layer 2 extension to address reachability of remote hosts on the same subnet as the moved host. This basically means that the traffic is switched into an L2 connection. This L2 connection may be in the form of one of the following:

- A VLAN
- An OTV extension
- A VXLAN tunnel
- An L2 LISP tunnel

Figure 3-7 illustrates the scenario.

https://t.me/learningnets

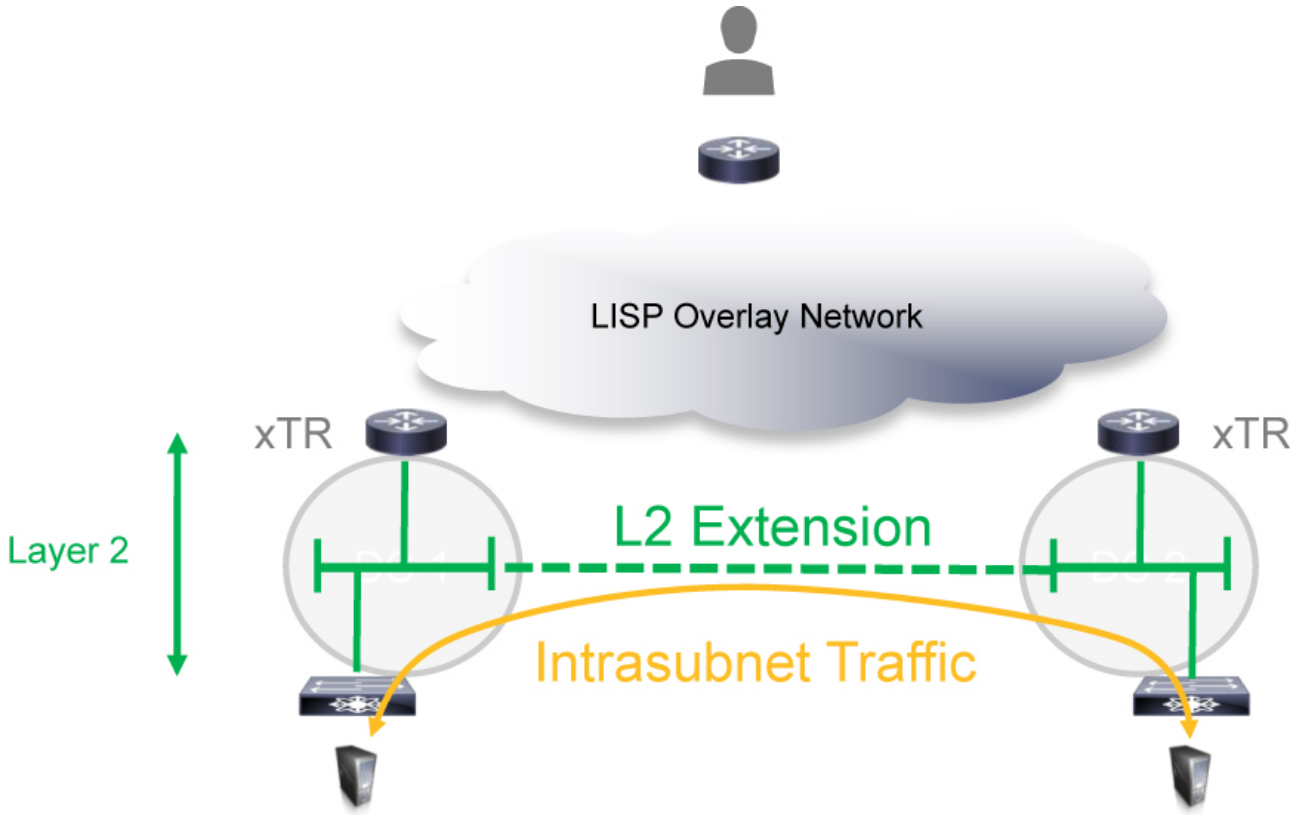


Figure 3-7 *Mobility and L2 Extensions*

In this scenario, ARP/ND Layer 2 resolution mechanisms take place over the L2 connection, and the multiple sites are susceptible to floods and broadcast storms because they are effectively connected to the same bridge domain. This L2 connection is there to accommodate for the traditional subnet structure. However, extending all this baggage of floods, broadcasts, and subnet boundaries is unnecessary. And, wouldn't it be great to get rid of all of this L2 machinery that many believe is terrible?

It's a long way from the days in which the guiding principle was "Switch when you can, route when you must." This recommendation was driven by better performance and cost on L2 switching devices versus the performance that was achieved on L3 routers. Today, switching hardware completes IP forwarding operations and L2 forwarding operations at comparable performance.

Unless non-IP hosts are in play, it is safe to assume that the vast majority of devices have an IP address on them, so why bother resolving an L2 address when you can simply forward based on the IP address of the destination? If you treat all routes as host IP routes and focus on routing those host IP routes rather than treating intrasubnet traffic different from intersubnet traffic based on what is actually a configured (and therefore arbitrary) subnet boundary, you could focus on a single address family, a single type of route, and a single type of lookup: a host IP lookup. The net effect of host mobility is that any IP address may show up on any xTR regardless of subnet boundaries, so pursuing subnet aggregation may not be possible, at least not how it was

https://t.me/learningnets

Technet24

pursued traditionally. But how is this possible? How can this scale? With LISP, the host granularity required to accommodate mobility is held in the Mapping Database System and not on the hardware forwarding tables of the xTRs. The xTRs cache only those host mappings they are using and nothing more. Because the Mapping Database System is detached from the topology, summarization continues to be achieved within the Mapping Database System regardless of the entropy of the hosts as they attach to different points in the topology. So maybe you don't need to adhere to the principles of subnetting after all and can also eliminate the need for switching and focus solely on routing to host addresses. Let's see how.

One view of how to get rid of L2 is to simply eliminate L2 from the host stacks and allow the host to just route. This approach requires customized stacks on the hosts, which not everyone has access to. So, the next thing to do is to take over the ARP process and force all hosts to talk to the router for both intrasubnet and intersubnet communication. If the xTR is topologically the first hop router that a host encounters—for the sake of simplicity, assume that the hosts are directly connected to the xTR—you can have the xTR always respond to any ARP requests issued by the host with its own MAC address. So, the host builds an ARP table that basically maps every single destination to the MAC address of the xTR. The xTR then routes every packet it receives based on the host mapping information in the LISP Mapping Database System. The ARP cache still exists on the hosts, but it doesn't really have any valuable information. No matter what the ARP request may be, the answer is always the MAC address of the xTR. So, you can provide intrasubnet and intersubnet communication without extending L2 and without managing subnets. This solution has implications on the way IP address management is done because a single large subnet is used for an entire network without having to do any of the tedious IP address planning that was done for years. Furthermore, in a world where the Internet of Things (IoT) promises to create a large IP address management challenge, IPv6 addresses are permanently assigned to devices and things, just like today you permanently assign MAC addresses to Ethernet interfaces.

Now that you understand why subnets aren't the answer, let's go back to the specific scenario of the structure of the multitier application. Could you think of a better way of providing the connectivity between the nodes in the fabric and shed the legacy VLAN and subnet-centered operations that a multitier application involved? After all, this subnet-centered model to building the multitier application stack was really the source of the problem. So, rather than just mimicking the VLAN structure by using VXLAN segments (the virtual equivalent to VLANs), could you treat the network as an interconnect capable of connecting any host to any host? And could the connectivity be governed by policy that focuses on the role the hosts play in the application rather than

https://t.me/learningnets

have to focus on the address of the host and the management of a multitude of segments that need to align with subnets and specific VLAN identifiers?

Because we've established that subnets aren't the answer and that their associated L2 boundaries can be eliminated, you need a mechanism to govern the connectivity of the different tiers of the hosts and ensure that traffic is actually steered through the right structure. Layer 3 overlays and demand/pull protocols such as LISP enable you to make the reachability of the hosts be subject to policies that reflect the structure of the multitier applications without the rigidity of stitching L2 segments or the stability and scale shortcomings of a Layer 2 network. The heuristic to follow is simple: if the web server must go through firewall 10 to communicate with the application server, then, when a packet from the web server triggers a Map-Request for the application server, the response is a path of locators that steers the traffic through the correct firewall. In LISP, this path of locators is known as an *explicit locator path* (ELP) and is basically an over-the-top engineered path. Because the LISP Mapping Database System provides a centralized interface through which it can be accessed, it is straightforward to have an orchestration system program those paths into the Mapping Database System. As a packet progresses through the path, the demand nature of the LISP requests and replies basically ensures that the right context is provided for the specific flow. Depending on the capabilities of the hardware, metadata identifying the particular application may be useful to have in the data plane, but regardless of these details, see how the combination of a pull and cache methodology, along with an extensible database that is not burdened by the responsibilities of routing, provides a viable mechanism for the creation of multitier applications without depending on subnets and L2 segments. This simple principle provides the basis for pervasive and scalable mobility. Consequently, a LISP-based networking stack for containers can be developed and contributed to the open-source community.

SEGMENTATION: 32 BITS NEEDED

As discussed previously, switching fabrics deliver a multitude of virtual networks. The term *segmentation* has been used to describe a network's capability to support multiple virtual networks. The virtual networks may be in the form of a routed virtual network or Layer 2 virtual network. The virtual networks are often referred to as *segments*.

The original goal behind segmentation in the data center fabric was to increase the number of segments that were available for the purpose of stitching the connectivity between tiers of an application. Switched networks that supported multitier applications for multiple tenants were running out of VLANs because many VLANs were required to stitch the services between the tiers of a single application. The

https://t.me/learningnets

reaction from networking operators and vendors was to pursue a larger number of segments to resemble the behavior of VLANs. The VLAN ID space is encoded in 12 bits, which represent a 4096 VLAN ID space. In the pursuit for more space, 24 bits in the LISP and VXLAN headers originated, which provides more than 16 million segment IDs to work with. Over time, segmentation has taken on a more meaningful role in the data centers, and the network in general, as a measure to keep the scope of connectivity for different activities constrained. When the scope of connectivity for hosts is limited to only the absolutely necessary, the attack surface available for malware to propagate is effectively reduced. More efficient mechanisms for building multitier applications are available, making security the leading driver behind segmentation in the data center and the network at large.

In the winter of 2010, Dino and I were in Barcelona, Spain, at Cisco Live Europe when we discussed the need for segmentation in LISP. The goal was to create L3 segments but be ready to support L2 segments if they were necessary in the long term. We went back and forth over lunch about how much space was required. We wanted 32 bits, which would give us more than 4 trillion segment IDs, but the switching hardware would not perform well with those 32 bits. Therefore, we ended up with 32 bits in the control plane but 24 bits in the data plane. We named the parameter an *Instance-ID*. The packet format as well as the control plane behavior were first documented in RFC6830bis (at the time draft-ietf-lisp-07) and RFC8060 (back then draft-farinacci-lisp-lcaf-00) in April 2010. In March 2011, a formal specification of the LISP Instance-IDs and the LISP header for the creation of Layer 2 segments was published as draft-smith-lisp-layer2-00. Many that were pursuing overlays over IP networks saw this header as a good path forward. Based on the header defined in the lisp-layer2 draft, a lot of negotiation across vendors took place in 2011, and the result of those negotiations was the VXLAN specification published in August 2011 (RFC7348). The VXLAN specification basically introduced the term *VXLAN* in lieu of Layer-2-LISP and dropped the use of some of the fields from the Layer 2 LISP specification. The subtle differences between the L2-LISP and VXLAN headers are illustrated in Figure 3-8. Any fields that didn't carry forward from L2-LISP to VXLAN were reserved for future use.

https://t.me/learningnets

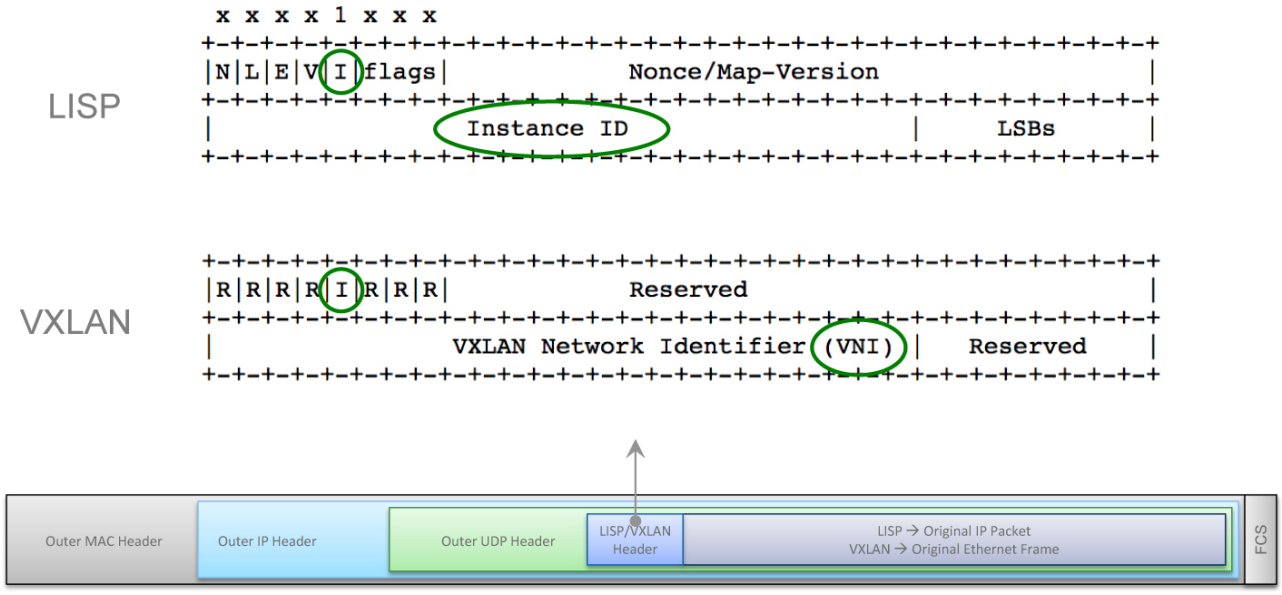


Figure 3-8 *L2-LISP and VXLAN Headers*

So VXLAN really existed 18+ months before its initial publication. VXLAN was really the original L2-LISP header. Even the UDP port number used for VXLAN was for a long time borrowed from OTV. The timeline illustrated in Figure 3-9 shows the sequence of events that led to the VXLAN specification.

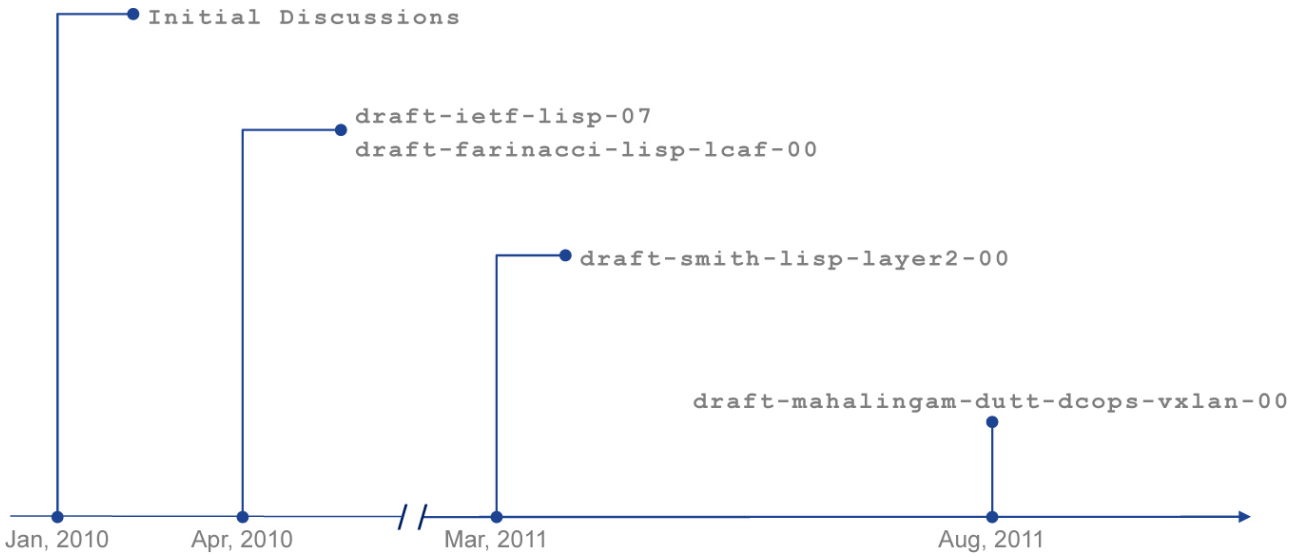


Figure 3-9 *VXLAN Header Specification Timeline*

The good news is that these headers are all bit-by-bit compatible, which makes it possible for a LISP implementation to use VXLAN as a data plane option if desired. The Cisco Campus Fabric is an example of an implementation of the LISP control plane with a VXLAN data plane.

The mechanisms by which segmentation is provided in LISP are rather straightforward and are documented in draft-ietf-lisp-vpn. The gist of the segmentation mechanisms is that both the control plane messages as well as the data plane are tagged with an Instance-ID. The Instance-ID is considered to be part of the EID because it basically

https://t.me/learningnets
Technet24

qualifies the EID's routing/forwarding scope. EIDs that are qualified with their Instance-ID are referred to as extended EIDs or XEIDs. Because both the control and the data plane are tagged with Instance-IDs, the Mapping Database System must accommodate its registrations within the scope of different Instance-IDs, and the xTRs must also maintain different forwarding tables to cache the mappings that were looked up for forwarding. For an L3 overlay, the segmented forwarding tables are virtual routing and forwarding (VRF) instances, and for the L2 overlay, the segmented forwarding tables are in the form of bridge domains or VLANs. The operations of LISP do not change at all, but they are now scoped by Instance to one set of tables within the Mapping Database System and the xTRs.

Let's look at the segmentation mechanisms in a little more detail. Segmentation is broken down into three main functional areas:

• Device segmentation

• Control plane segmentation

• Data plane segmentation

Device Segmentation

To segment the network, the forwarding devices segment their forwarding tables into multiple forwarding contexts. A VLAN is an example of a forwarding context where only the MAC addresses learned on ports that belong to that VLAN can be reached by hosts that are connected to that same VLAN. A VRF is another example of a forwarding segment. In the VRF case, only the IP addresses that are learned on ports that are connected to the VRF can be reached from a host in the VRF. So, a forwarding context is a forwarding table, and you assign hosts to a forwarding context by assigning the ports they use to connect to the network to the particular forwarding context. Figure 3-10 illustrates this concept of contexts and port membership.

https://t.me/learningnets

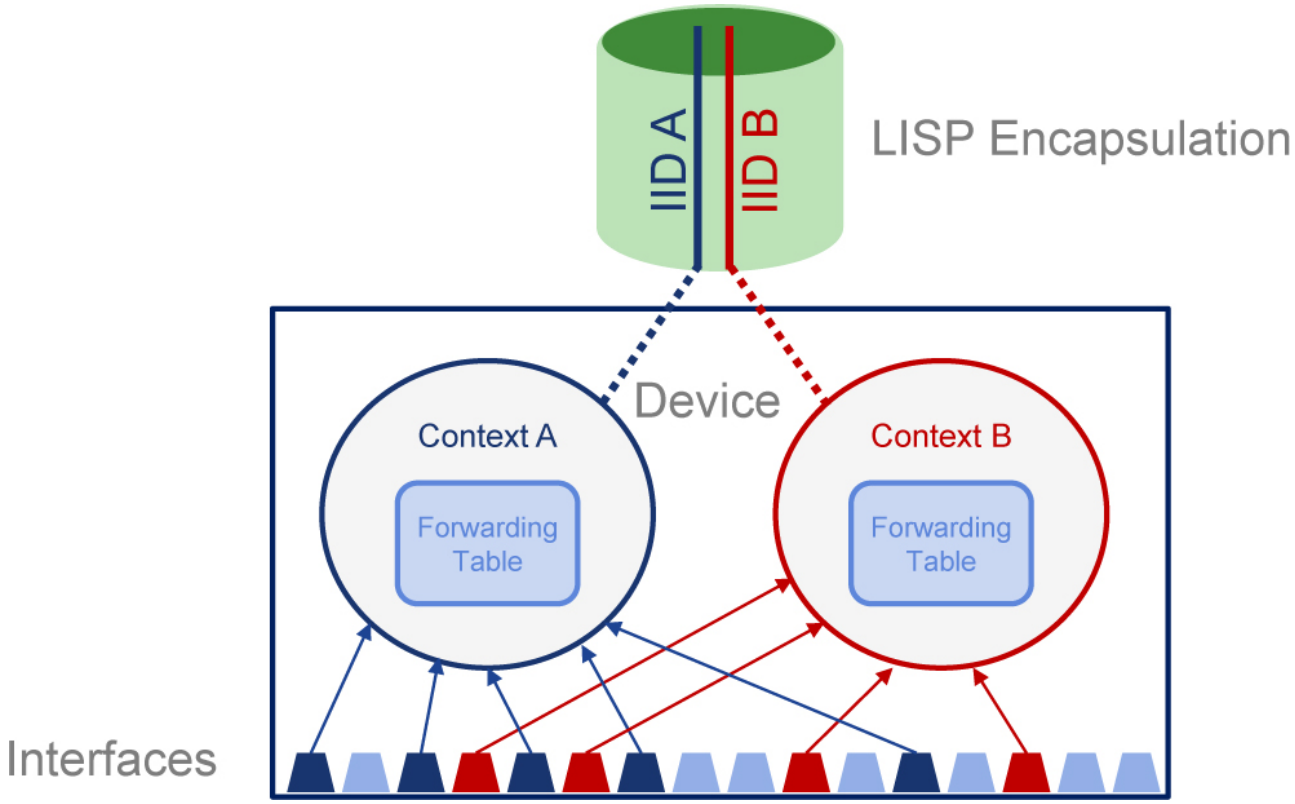


Figure 3-10 *Forwarding Contexts*

Certainly, a network is made of a multitude of devices. The forwarding contexts maintained in a device also include addresses (and routes) for hosts that are connected to other devices in peer contexts. So, the path between devices must also be segmented to preserve context information as control and data traffic travels from one device to another. To maintain this context information across devices, both the control and the data planes between the devices must be segmented. This is necessary so that the separation provided by the forwarding contexts is effectively extended across multiple devices.

Control Plane Segmentation

LISP achieves control plane segmentation by tagging all control plane messages with an Instance-ID and by scoping the tables in the Mapping Database as well as the Map-Caches on xTRs into a separate table per Instance-ID.

In LISP, the Instance-ID is considered an attribute of the EID. With segmentation enabled, LISP simply operates on an extended EID or XEID, which is the tuple of Instance-ID and EID. The process of forwarding table lookups, Map-Requests, Map-Replies, and Map-Caching happens as described in previous chapters, but all operations are done within the scope of a particular Instance-ID. The Instance-ID to add to a Map-Request is determined at an ITR simply by the forwarding context in which data traffic is received. Thus, a VRF is associated with an Instance-ID, and interfaces are associated with the VRF (an interface would belong to only one VRF).

https://t.me/learningnets
Technet24

When traffic is received on any of the interfaces associated with the VRF, the ITR tries to find a route to the destination on the VRF (forwarding context) that the ingress interface is associated to, and if there isn't a route or a LISP mapping cached, a LISP lookup is initiated with the Instance-ID that is assigned to the VRF. For LISP to provide an L2 segmentation service, a similar process takes place for L2 lookups in the context of a VLAN. The Map-Replies, and any other LISP signaling, include the Instance-ID for the EID so that any LISP elements (Mapping Servers or xTRs) know which Instance to use for its local operations.

Data Plane Segmentation

The control plane ensures that mappings are handled in the context of the different contexts. The net result is a segmented Map-Cache and routing table at the xTRs. When an ITR obtains a mapping and caches it in the right context (VRF or VLAN), the ITR must encapsulate the traffic to the ETR with the RLOCs in the mapping. The ITR includes the Instance-ID for the XEID when it encapsulates the traffic to the ETR. On the receiving end, the inclusion of the Instance-ID in the data plane header allows the ETR to know which forwarding context (VRF or VLAN) it must use to look up the destination EID to forward the traffic once it is de-encapsulated.

Extranet VPNs

Now that you understand how segmentation is done in LISP, let's explore some interesting things to bring unique flexibility to a segmented network. One good example of the flexibility of the LISP demand-based directory approach is how extranets are procured in LISP. It is a common requirement of many segmented networks to have the ability for multiple virtual networks to access one or more shared virtual networks where common services are hosted. In these scenarios bidirectional communication must occur between the shared segment and all other segments, but the shared segment should not serve as transit to provide connectivity between the segments that are connecting to it. For example, two customer virtual networks, A and B, access a Shared virtual network. EIDs in virtual network A may talk to EIDs in virtual network Shared, and EIDs in Shared may be able to connect back to EIDs in A. Virtual networks B and Shared may also have bidirectional communication, but EIDs in virtual network A cannot communicate with EIDs in B via Shared. So, the segmentation between A and B is maintained while they both access the Shared segment. The way traditional routing protocols achieve this is by leaking routes from A to Shared and from Shared to A, similarly between B and Shared, with the restriction that Shared does not leak routes learned from B to A and vice versa. The result of doing this is that all routes in A and B are replicated at least once in Shared, and all routes in Shared are replicated as many

https://t.me/learningnets

times as there may be virtual networks subscribing to Shared. This results in a large amount of replicated state, which is further exacerbated by the fact that for this to work, all VRFs must be instantiated at every router (edge routers in the case of overlays). So, all routers must be able to accommodate multiples of the number of routes that exist in the overall network. This state problem is solved elegantly in a demand-based system such as LISP.

If you look at the problem of the extranet, what you are mainly trying to accomplish is enforce a reachability policy that indicates a hub-and-spoke relationship between the Shared virtual network and any virtual networks requiring access to it (A and B in this example). Furthermore, the policy restricts the hub virtual network from acting as a transit for connectivity between the spokes. So, wouldn't it make sense to simply state who the hub is and list its spokes? And could you make the responses from the Map-Server be based on such policy? In fact, that is what the extranet functionality defined in draft-ietf-lisp-vpn actually does. It has the great advantage that routes from one VRF are not copied into another on the xTR forwarding tables, and lookups are expanded to go across multiple segments as long as the segments are included in the policy. So, when a host in virtual network A issues a Map-Request for a Shared host, the ITR tries to resolve the Shared destination without any knowledge that the destination is in a different segment, and it is up to the Mapping Database System to expand the lookup beyond segment A and into the Shared segment. The Mapping Database System is programmed with the extranet policy to understand that this should be done, and the Mapping Database System replies with the locator for the destination as well as the Instance-ID to be used to reach that destination. When the requesting ITR receives the reply, it caches the mapping in the forwarding context for virtual network A with a flag that indicates that a specific Instance-ID is to be used to encapsulate traffic toward the destination it resolved. As you can see, the mappings are only cached when needed and where needed (just like all mappings in LISP), and the ITR does not need the Shared VRF to be instantiated locally (nor does the receiving ETR need an instance of the VRF for virtual network A). So, VRFs (or VLANs) are instantiated only at the xTRs where they have connected hosts. Therefore, there is no need to create additional state in the network because the forwarding policy now includes extranet type connectivity. This statement can be extended to all connectivity policy, so the LISP directory actually allows you to make the network truly programmable without impacting the amount of state or behavior of the forwarding plane. The savings in state and costly forwarding memory are significant. Also note that when you explore multicast, extranets can be provided in LISP indistinctly for unicast or multicast traffic. If you've ever tried to share multicast traffic across VPNs in an IP/BGP Multi-Protocol Label Switching (MPLS) segmented network, you are likely intrigued at this point because doing this in BGP is

https://t.me/learningnets

intellectually painful, to put it mildly. If you have never done such a thing in BGP, then life is good because you may never need to go there; you can use LISP to solve the problem in a much easier manner.

POLICY: THE NETWORK AS AN ENFORCER

The data center network is increasingly focused on the delivery of security policy. As businesses transition to a fully digital model, all information and most processes are now hosted in the data center, which makes the data center the target for attacks that attempt to compromise and illicitly profit from the information required to run these digital businesses. You've also seen how the data center network has had to evolve to a fabric that enables any device to connect essentially anywhere. The network is in a unique position to provide location information that is critical in the enforcement of the security and process policies designed to support the digital business environment. Because of the mobility and variability of the hosts involved in the digital business, the enforcement of policy is of a transactional nature. A demand-based control plane is transactional in nature and, therefore, as you may suspect, a good fit to complement the policy enforcement mechanisms required to secure the digital business.

This chapter covers how segmentation is achieved in LISP and how policy exceptions to the segmentation are put in place seamlessly to provide secure extranet connectivity. The way LISP provides extranet connectivity is a great example of a transactional mechanism that enforces a connectivity policy through the control plane of the network.

In a system where policy is defined and passed to the LISP control plane for enforcement, the responses of the Mapping Database System vary depending on the context of the requestor. The Mapping Database System can thus tailor its replies based on the access control policy, and the enforcement of the policy is done purely in the control plane by influencing the mappings provided for different connections. The richer the information included in the Mapping Database System, the more precise the access policies that are enforced in the control plane are. For example, an implementation of LISP could include Layer 4 port numbers or group information as part of the semantics of the EIDs registered. In other words, the EIDs are extended to include the additional information, and all EID-related operations are executed on the extended EIDs. In this way, an access control policy that matches the extended semantics to take simple actions of permit or deny are enforced in the control plane by providing different replies to Map-Requests subject to the policy.

Depending on the kind of information that is to be added to the EIDs, some data plane

https://t.me/learningnets

assistance may be required. For example, a policy inclusive of Layer 4 semantics requires the Map-Cache used by the forwarding plane of the ITRs to include entries based on the extended EID—that is, not just the destination IP address, but the tuple of IP plus port number. This leads to a high level of granularity in the forwarding tables and scaling challenges. Some policy enforcement systems abstract the policy from the IP addresses altogether to aid the scaling of the policy enforcement state. One interesting example of how the LISP control plane assists in the rendering of policy at large scale is the implementation of group-based policies in the Cisco Campus Fabric solution. In the Cisco Campus Fabric, group-based policies are rendered by evaluating scalable group tags (SGTs) rather than IP addresses. This mechanism is geared toward simplifying access control lists (ACLs) by abstracting the groups involved in the ACL with a level of indirection from the IP addresses and thus reducing the number of entries that an ACL has. The enhanced ACL is often referred to as a scalable group ACL (SGACL). For a policy to be enforced, both the source and destination scalable group tags must be known at the policy enforcement point. Each edge device knows how its locally connected IP endpoints map to scalable groups. To make the handling of the group tags scalable, policy is enforced at egress where the scalable group tag for the destination endpoint is locally known, while the SGT for the source endpoint is obtained from the data plane header as the ingress device (which knows the SGT for the source endpoint) includes the source SGT in the data plane header when sending traffic. The mechanism simply leverages the data plane to communicate the source group tag to the enforcement point without requiring a separate protocol and without requiring the creation of an IP endpoint to SGT mapping state at the policy enforcement point.

The SGT egress enforcement mechanism works well for access control policies as long as the destinations are directly connected to the policy enforcement point. In some cases, however, you want to enforce the policy on a device that is not directly connected to the destination endpoints. Say you wanted to enforce the policy at ingress instead of egress. You may want to enforce at ingress simply to reduce the consumption of bandwidth in the network or to use the scalable group model to enforce a quality of service (QoS) policy that is needed to be enforced from ingress onwards. In these scenarios, the ingress device has to learn which IP addresses belong to each scalable group and maintain a table of these mappings and translate the group policy to granular IP-based rules. This means the benefits of group-based policies of scale and structure are, to a large extent, lost.

LISP provides a simple mechanism that allows an ingress node to obtain the scalable group information for those remote endpoints toward which it may be sending traffic. When an ETR registers a host with LISP, the registration includes the SGT value that

https://t.me/learningnets
Technet24

corresponds to the host being registered, so that the Mapping Database System now includes not only the locator for the host but also the host's SGT. When an ITR issues a Map-Request for a particular host, the Map-Reply includes the host's SGT, and the ITR then enforces policy at ingress because it has both the source and the destination SGTs for the flow. This simple mechanism also enables the scalable enforcement of group-based policies in firewalls and other devices that are not directly connected to the destination hosts.

When the policy enforcement point is a firewall, usually the firewall is not directly connected to either the source or the destination endpoints. Thus, even if a firewall is capable of parsing scalable group tags, the full table of IP to SGT mappings must be communicated to the firewall for it to classify destination endpoints into SGTs. Because the firewall inspects traffic in both directions, all endpoints are destinations in some direction of the flow and a full table of IP to SGT mappings is required, which means there is little gain in receiving the source group tags in the data plane. The obvious solution is to include both the source and the destination SGTs in the data plane, rather than only the source SGT. However, this means that the ingress network device actually knows which scalable group the destination endpoint belongs to so it can impose the destination SGT label in the data plane. The LISP demand-based mechanisms described provide a scalable manner for an ITR to procure the destination endpoint SGT information and include the destination SGT in the data plane header for enforcement by a firewall downstream. An even better scenario is one where the data plane is not required to carry any group information and the firewall simply procures the IP to group mappings it needs from the LISP Mapping Database System directly.

> Note
>
> *The mechanisms for encoding destination group information in the data plane are defined in several encapsulation proposals to the standard bodies. The Service Function Chaining workgroup at the IETF does a good job of documenting the problem and potential solutions.*

THE HYBRID CLOUD AND CARRIER NEUTRALITY

The increasing use of cloud services for the hosting of critical business applications has challenged the networking industry in interesting ways. Most businesses find themselves in an environment where they maintain some assets in a private data center and other assets are maintained at a cloud provider's data center. Both these data

https://t.me/learningnets

centers leverage cloud models and are referred to as *private cloud* and *public private cloud*, respectively. Assets are moved between these data centers dynamically as the criteria by which an asset is required to reside in one type of facility versus another is wide ranging and constantly changing. The combination of private and public cloud facilities is known as a *hybrid cloud*.

In connecting the business to the hybrid cloud, the network must provide seamless mobility, scale, and security. It is also extremely important for businesses to have the flexibility of adding or removing cloud providers seamlessly. This has many implications, but from the network perspective, the implication is that a common connectivity model must allow connectivity to any cloud service. By providing an over-the-top connectivity solution with large-scale mobility, LISP is an interesting cloud connectivity option.

The two main models to connect a hybrid cloud are

• Use co-locations as connectivity hubs for all public and private cloud services.

• Create an overlay network with presence in the different cloud facilities.

When using the co-location model, you deploy a high-capacity switching fabric within the co-location facility where the fabric has direct connections to all the different fabric facilities involved. The switching fabric is responsible for connecting the business to the different cloud facilities and is also responsible for inserting security service nodes that serve as centralized policy enforcement for the connectivity to the data centers. Figure 3-11 illustrates this connectivity model. The cost of using the co-location facility to host this fabric is an important consideration. The switching fabric uses LISP, as described earlier in this chapter, to deliver mobility and segmentation and to enforce policy. LISP is also used to provide secure connectivity with mobility and segmentation to the co-location facilities, as illustrated in Figure 3-11. The two key advantages of this model are the quality of the connections between the co-location and the cloud providers and the fact that the policy enforcement points are centralized into the co-location facility and made agnostic to the different cloud providers.

https://t.me/learningnets

Technet24

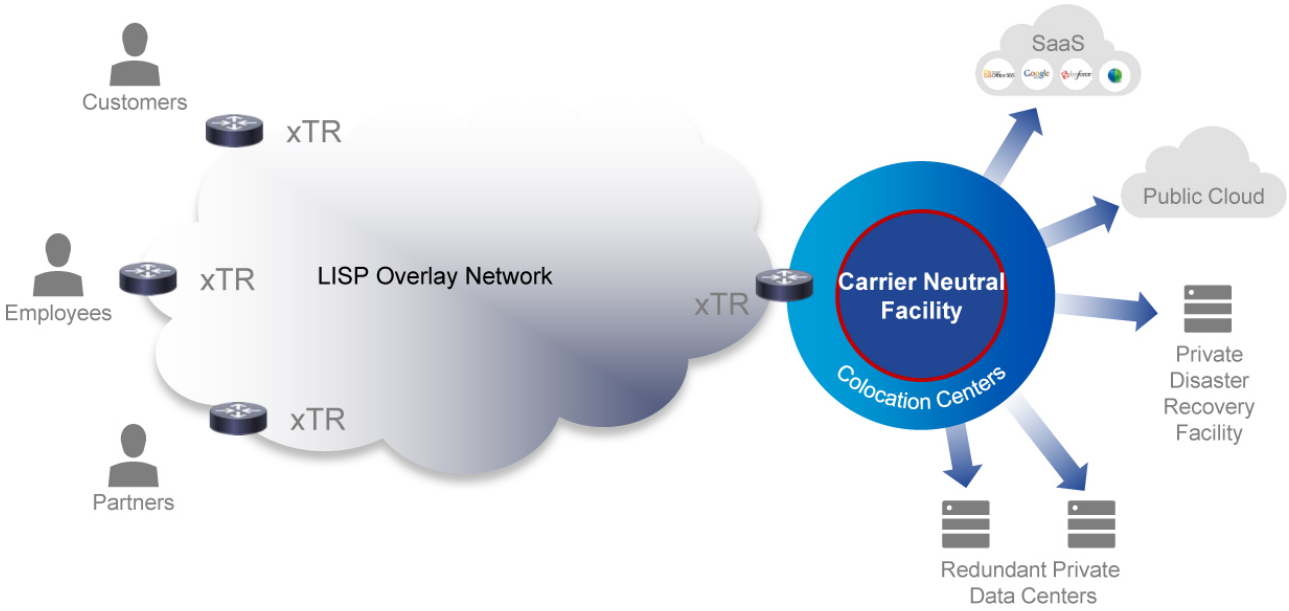


Figure 3-11 *LISP Connectivity to Cloud Neutral Facilities*

Alternatively, the overlay model that does not leverage the co-location facilities may be more cost-effective. Through the use of network function virtualization and automation of policy, a network overlay between the enterprise and the cloud facilities still provides the benefits of a provider-agnostic policy deployment. The overlay model assumes that each cloud provider has an overlay tunnel router and that the overlay virtual network extends into the enterprise. There are a series of advantages to using LISP for such an overlay. Figure 3-12 illustrates how an overlay network is used to provide normalized access to the different data center locations that conform a hybrid cloud.

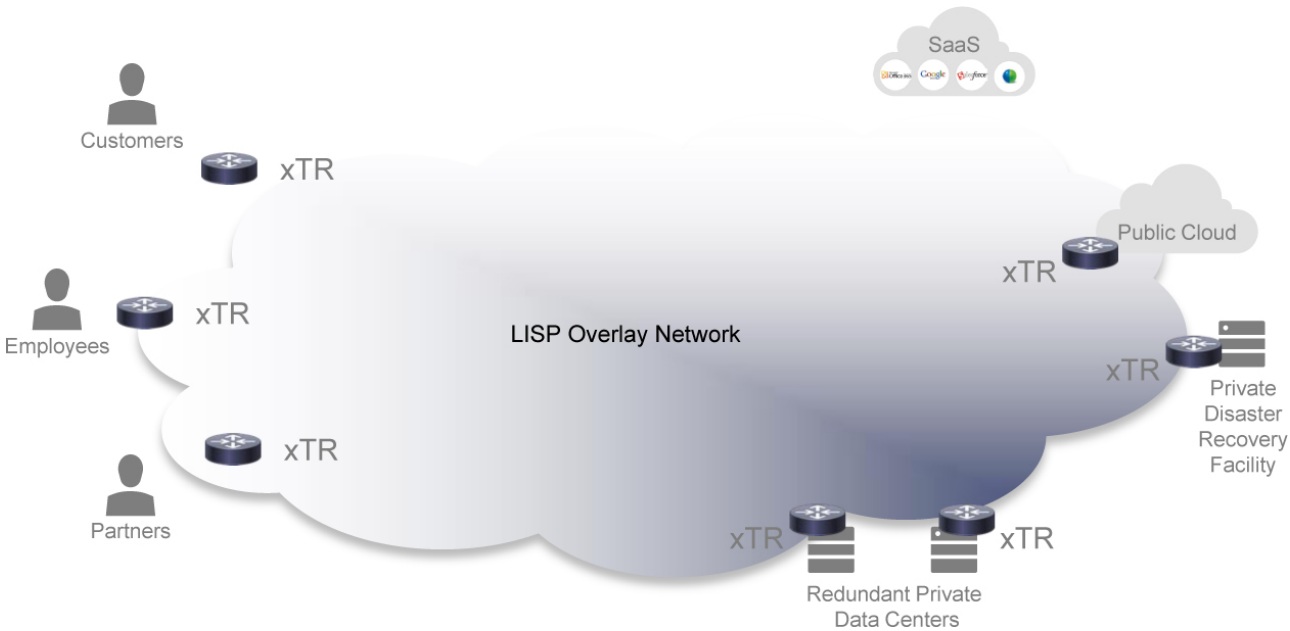


Figure 3-12 *LISP Hybrid Cloud Connectivity*

LISP provides the mobility required for cloud connectivity with host granularity and across any number of cloud facilities. The scale benefits of a demand directory service like LISP are hard to match when mobility across cloud facilities requires maintaining

https://t.me/learningnets

reachability at a host level without summarization. LISP is particularly good at handling this type of state because Map-Caches are populated only with relevant entries and the registered hosts may still be summarized within the Mapping Database System while their mobility is not constrained by topological subnetting boundaries.

Cryptography is integrated into the LISP overlay in a scalable and efficient manner. This capability to deliver cryptography is key to the connection between the enterprise and the cloud. Chapter 5, "Mega-Scale Access Networks: LISP, User Access, and the Internet of Things," explores the details of how LISP provides cryptography.

LISP may steer traffic through particular service nodes so that the security policies for data center access are properly enforced. This is particularly interesting in the co-location fabric where sophisticated traffic steering may be required. It is also relevant in the overlay cloud access model, although for the most part it is expected that virtual micro-services are deployed right in front of the applications, making the need for network-based service insertion obsolete in some instances.

SUMMARY

LISP enables a directory of extensible and highly scalable endpoints. It has laid out the key principles for the design of network overlays in the data center. Switching fabrics are slowly catching up to the potential of what LISP could enable them to deliver. LISP is also widely used to steer traffic to a particular data center based on location of particular hosts, enabling the large-scale mobility required to access the different locations of a hybrid cloud. The capability to provide host-based mobility at virtually unlimited scale and the capability to extend the set of information handled by LISP to include elements of policy and geo-location are critical to the next wave of functionality for the data center network and a must for the hybrid cloud access network.

https://t.me/learningnets

Technet24

Chapter 4. The Wide-Area Network: Bringing Traffic from Access to the Data Center

Wide-area networks (WANs) evolved at a rather conservative pace over the years. The recent adoption of overlays in areas of the network adjacent to the WAN has fueled an appetite for a simpler and more efficient WAN design.

This chapter discusses the challenges encountered in WANs and how networking technology evolved to meet these challenges and enable alternative designs to the WAN. LISP plays a pivotal role in the technological evolution of the WAN and is the focus of this discussion.

MODERN WAN SERVICES

For many years the wide-area network was designed as a collection of hub facilities that aggregate traffic from a relatively large distribution of locations. Each hub services the traffic for a multitude of locations within the same geographical area. Hub facilities connect to a backbone network that provides connectivity between the hubs and also connect to one or more data center facilities. Figure 4-1 illustrates this hub-centric network design.

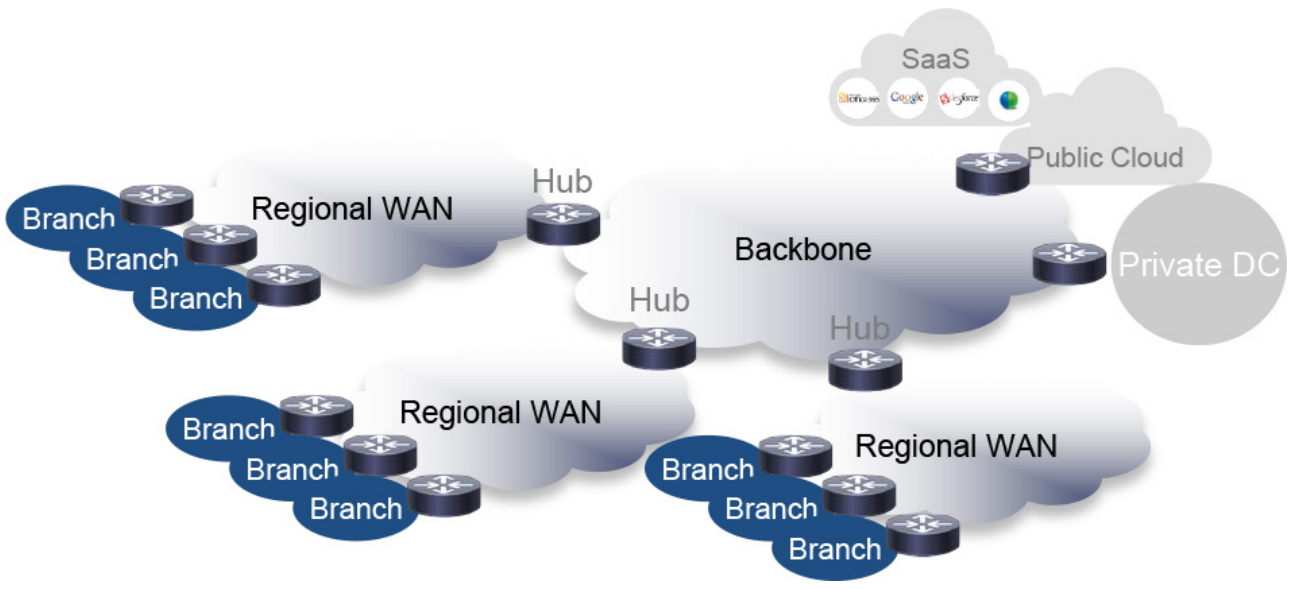


Figure 4-1 *Topology-Constrained Wide-Area Network*

https://t.me/learningnets

The hierarchical design of the WAN was anchored around the scale characteristics of the networking technology available at different points in time. Cost, distance, available bandwidth, and latency were other aspects that led to hierarchical WAN designs. Aggregating traffic at different hubs was also leveraged as a way to insert policy enforcement devices such as firewalls and intrusion prevention/detection systems in the flow of traffic. In a modern WAN, the technologies involved are multiple, and their combination is operationally complex.

The rich history of the WAN goes back to the use of point-to-point circuits during the 1970s and 1980s. At that time, running a WAN meant running a dedicated last-mile interface and circuit between any pair of locations. During the 1990s, Frame Relay lowered the cost of the WAN significantly, allowing the expensive last-mile physical connection to be shared across multiple remote connections. This capability lowered the cost of the service and its operations significantly, as well as the cost of the router connecting to the service. Frame Relay became pervasive quickly, but it was still a circuit-switched protocol, and for all practical purposes, this mandated a hub-and-spoke design of the Frame Relay circuits. The convergence of voice and video into the data network required unconstrained any-to-any connectivity, which was offered by a connectionless protocol such as Multiprotocol Label Switching (MPLS). In the 2000s, Frame Relay gave way to MPLS networks for provider-delivered WAN services. This marked a big transition from hub-and-spoke services to any-to-any services delivered using MPLS on provider edge (PE). Some operations still required point-to-point Layer 2 services, particularly in the metropolitan area. Carrier Ethernet Services became popular in the early 2000s and are still used in some applications. Nevertheless, the bulk of the industry eventually moved toward any-to-any MPLS virtual private networks (VPNs) for the WAN.

In parallel to the evolution from private circuit networks to connectionless MPLS networks, broadband Internet access was introduced in the 1990s and evolved rapidly to offer much higher bandwidth capacity at a fraction of the cost of an MPLS service. Many enterprises started leveraging broadband access to provide best-effort backup paths for their main WAN connections. These backup connections used the public Internet combined with IPsec VPNs to maintain data confidentiality over the public network.

Today the MPLS WAN service is the de facto private WAN service used by enterprises. MPLS WAN services command a hefty premium over the cost of Internet bandwidth. This is evidence of the balance between cost and quality that enterprises must strike. For most businesses, service levels must be guaranteed, even at the expense of using a lower bandwidth service at a multiple of the cost. MPLS networks are engineered in

https://t.me/learningnets

such a way that the service provider can give assurances around the service levels delivered. The Internet, on the other hand, is made of an intricate mesh of cross-organizational network peering relationships that make it difficult to provide any sort of service guarantee.

MPLS VPN network services do not provide encryption, and they offer customers little flexibility. Customers supplement the MPLS service with over-the-top (OTT) services delivered at the customer premises equipment (CPE). One example of such an OTT service is the cryptography delivered by IPsec VPNs, Dynamic Multipoint VPNs (DMVPN), or Group Encrypted Transport VPN (GETVPN). Other examples include virtual routing and forwarding (VRF) based segmentation of traffic. Thus, customers must manage their own CPE and OTT services or purchase a managed OTT service from the provider. In the case of IPsec VPNs, the tunnels are point-to-point and resemble the circuits of the Frame Relay era. With DMVPNs, many hub-and-spoke dependencies are partially reintroduced, although spoke-to-spoke connections can be created on demand.

Different OTT WAN services have limitations around scale and the capability to simultaneously leverage transport from different providers. Most importantly, before the advent of modern software-defined WAN (SD-WAN) solutions, manageability of these OTT VPN services was challenging, and none of the services offered a good way of leveraging the Internet while maintaining an acceptable service level guarantee.

Since the early 2010s, there has been a high demand for a better solution to the WAN problem. Specifically, there is a strong desire for an OTT solution that supports the following attributes:

• Hybrid WAN transport connectivity (the ability to use multiple transport services simultaneously)

• Scale

• Direct site-to-site communications

• Pairwise cryptography

• Segmentation

• Application optimized routing

• Manageability

https://t.me/learningnets

• Multipathing

• Path entropy

This evolution of the WAN OTT space was dubbed the SD-WAN. As with the original software-defined networking propositions, the SD-WAN is built on an overlay virtual private network (VPN), and a large part of its value derives from the mechanics of such an overlay. The principles of the SD-WAN were inspired by LISP and the advanced functionality LISP made possible. As a multitude of SD-WAN start-ups emerged with the goal of creating better overlay networks, the networking industry had at its disposal years of research on overlays completed by the LISP community. In the creation of an optimal overlay that fulfills the requirements of the SD-WAN, a demand-based approach such as LISP makes a big difference. Following sections in this chapter are devoted to each aspect of the SD-WAN evolution and how LISP plays to each one of them.

LISP natively supports the concept of a hybrid WAN, which entails the use of multiple disparate underlay transport networks over which the overlay will run simultaneously. LISP offers tunnel router (xTR) multihoming, which is described in detail later in this chapter. At a high level, LISP registers endpoint identifiers (EIDs) with routing locators (RLOCs) that belong to multiple different transports. This happens naturally in the protocol.

LISP is designed as an any-to-any stateless service and represents a leap in scalability for this type of service given its stateless and demand-based approach. Because of its demand-based resolution of reachability information, LISP provides an ideal framework to enable pairwise security key exchange negotiation. In LISP, security key exchange is integrated with the mechanisms used for the resolution of reachability information. The scalability implications of an integrated pairwise key exchange negotiation and the simplification of the stack that this integration brings are a welcome development in the space of OTT WAN services, which are explored in a later section.

Previous chapters discussed in detail how LISP brings segmentation and multihoming to the switched fabrics. The WAN benefits from the same LISP behavior. The next few sections describe in detail how LISP delivers each one of these attributes of the software-defined WAN.

HYBRID WAN: EFFICIENT XTR MULTIHOMING

A hybrid WAN is a WAN service made of multiple transit networks. The main benefit of

https://t.me/learningnets

Technet24

a hybrid WAN is that the multiple transit networks are handled as a pool to balance different workloads and optimize the utilization of the combined transport services. Thus, the resulting bandwidth of a hybrid WAN may be roughly compared to the aggregate of the bandwidth of the underlying transit networks.

LISP simplifies the connection to multiple WAN transports by separating the customer prefixes from the provider routing space, streamlining multihoming operations and providing embedded traffic engineering policies. Figure 4-2 illustrates a hybrid WAN; although the underlay is realized over multiple transports here, the overlay is abstracted from this multitude of networks and sees it as one. The variety of transports can be a mix of private MPLS WAN services, metro networks, and the public Internet. None of these transports are exposed to the customer prefixes, nor is the overlay aware of the different transports. LISP separates the underlay and overlay networks seamlessly.

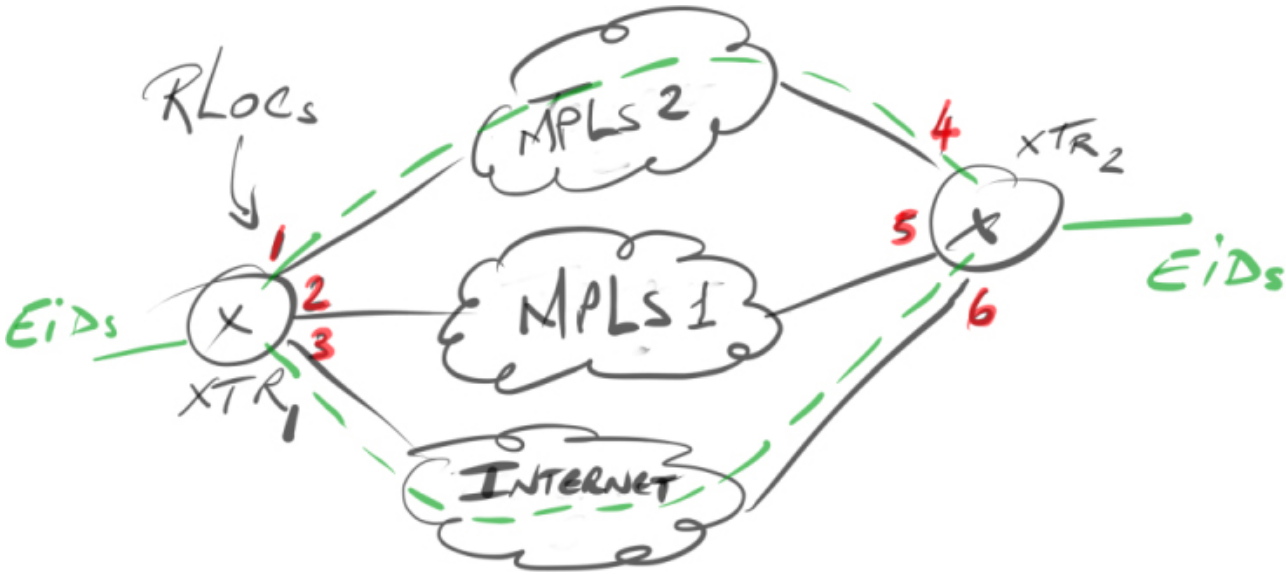


Figure 4-2 *An Overlay-Enabled Hybrid WAN*

When the WAN uses a LISP OTT service, customer prefixes are not advertised into the provider transport networks, but the customer EID prefixes are handled in the LISP Mapping Database System. This means that the customer EID prefixes are totally independent from the provider, allowing the customer the flexibility to add or remove provider services without any impact on addressing or policy. Furthermore, because customer EID prefixes are not advertised to the provider transport networks, the network-to-network interface (NNI) between the customer xTR and the provider edge (PE) can be simplified significantly to a minimalistic approach focused on providing underlay connectivity for the RLOCs. As shown in Figure 4-3, in the underlay NNI, the xTR simply has a default egress route leading to the PE; for the traffic inbound to the site, the RLOCs connecting to each PE are part of a directly connected prefix at the PE. In other words, the routing at the NNI is static and prescriptive.

https://t.me/learningnets

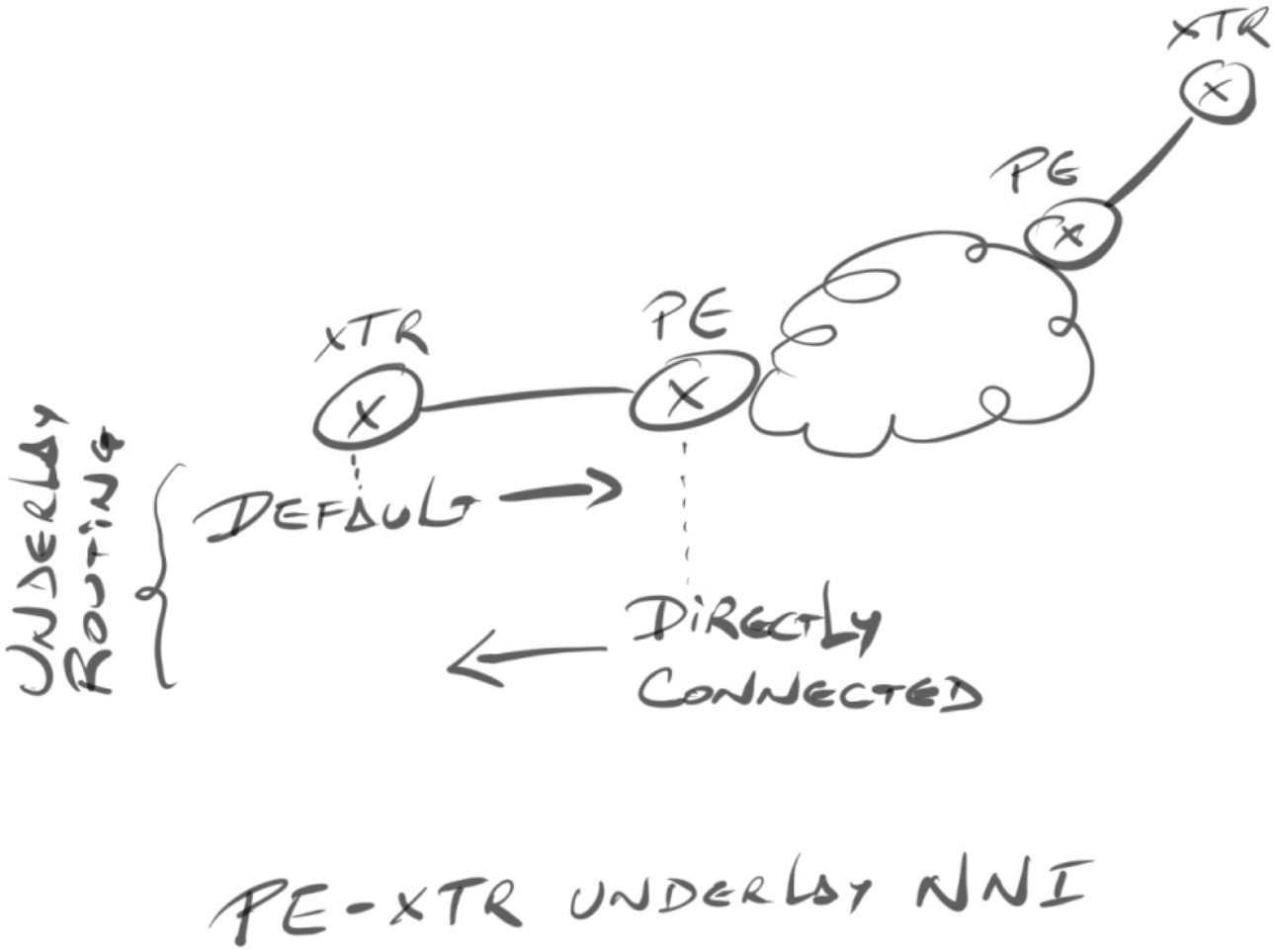


Figure 4-3 *xTR to PE Underlay Network-to-Network Interface*

Arguably, customer route suppression and the simplifications highlighted so far are properties of CPE-deployed overlays in general. The benefits of LISP as a protocol for such an overlay become evident when you understand how the LISP model seamlessly handles multihoming. LISP multihoming is free of complex routing policy such as that required when using traditional routing protocols like Border Gateway Protocol (BGP) to attach to multiple networks. A simple model of EID-specific priorities and weights allows the LISP operator to manage multihoming policies in LISP without the burden of handling complex route-maps and policies constructed around a nonintuitive hierarchy of metrics and attributes such as Autonomous-System-paths (AS-paths), Multi-exit Discriminator (MED) values, and community attributes.

Traffic is distributed over the different transit networks in a hybrid WAN based on a variety of criteria. In some instances, you may prefer to use a specific transit network for traffic corresponding to a particular application or destination, leaving other transit networks available as a fallback for that particular application.

In the LISP protocol definition, priorities and weights indicate the receiver side preference for use of the different RLOCs in a mapping. When different RLOCs are connected to different transport networks, as illustrated in Figure 4-4, LISP natively supports the hybrid WAN scenario.

https://t.me/learningnets
Technet24

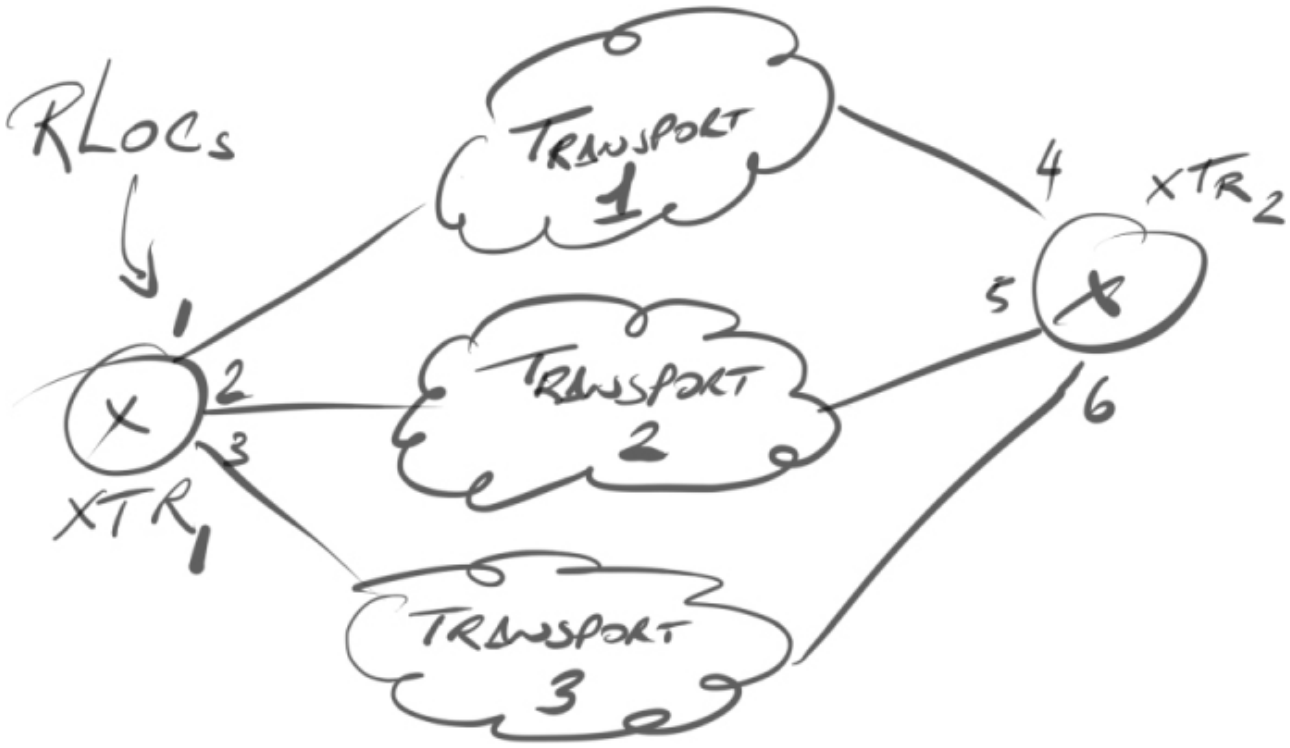


Figure 4-4 *Connecting an xTR to Different WAN Transports*

LISP uses a two-level prioritization policy articulated as the combination of priority and weight. Each RLOC in an EID-to-RLOC mapping is given a priority and weight that is relative to the priority and weight of other RLOCs. At the top level, there is an absolute priority, in which only available RLOCs with the highest priority are used. RLOCs with a lesser priority are not used unless all RLOCs with better priority values are unavailable. For instance, in a system with RLOCs at priority levels 1 and 2, as long as there is a viable RLOC with priority 1, only the priority 1 RLOCs are used. If all priority 1 RLOCs are unavailable, the next level of priority is put into use—in this case all available RLOCs with priority 2. Among RLOCs with the same priority, weights influence the balancing of the traffic across the different RLOCs. A set of RLOCs with equal weights for each RLOC results in an equal distribution of flows across the RLOCs in the set. For instance, a set of two RLOCs assigned weights of 75 and 25, respectively, results in one out of each four flows using one RLOC and the remaining three using the other RLOC.

With this combination of priorities and weights, users can articulate absolute preference as well as weighted load-balancing preferences. When each RLOC in a set is connected to a different transport service, this simple mechanism is used to express preference among underlay transport services.

The policy that specifies which underlay transport is preferred is driven from the receiving site, where priorities and weights are set per EID. Each EID-to-RLOC mapping has its own set of priorities and weights. So, different destination EIDs at the same location may have different reachability policies, although they are connected to

https://t.me/learningnets

the same set of egress tunnel routers (ETRs). For example, Figure 4-5 illustrates a data center site reachable via three different transport networks from the main enterprise campus. In the data center site, two applications are hosted. Application A uses prefix A as its EID (10.10.10.0 /24), and application B uses EID prefix B (10.20.20.0 /24). The data center connects to the WAN using a pair of xTRs: xTR1 and xTR2. Both xTRs are connected to every transport network; thus, each xTR has at least 1 RLOC per transport. xTR1 has RLOCs 11, 12, and 13 (in this example, the first digit identifies the xTR, and the second digit identifies the transport, so 12 is the RLOC that connects to transport 2 on xTR1); xTR2 has RLOCs 21, 22, and 23, connecting to transports 1, 2, and 3, respectively.

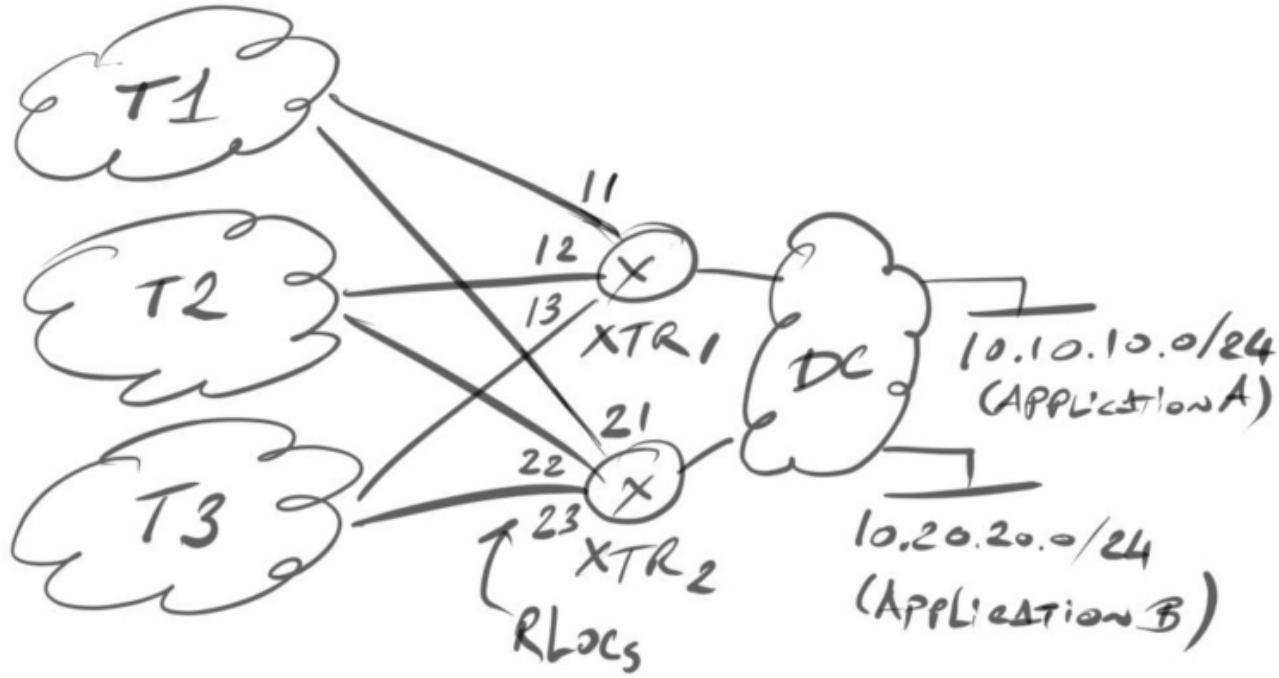


Figure 4-5 *Applications in a Multihomed Data Center*

Each xTR registers the application EIDs with all RLOCs, and each registration has its own independent set of priorities and weights. Say that the desired policy states that Application A should always use transport network 1 unless it becomes totally unavailable and that Application B should be balanced across transports 2 and 3 at a 1:2 ratio and use transport 1 only as a backup to 2 and 3. The resulting priorities and weights for EID prefixes A and B in this system are illustrated in Table 4-1.

https://t.me/learningnets

Technet24

EID	RLOC	Priority	Weight	Policy
Prefix A (10.10.10.0 /24)—Application A	11	1	50	Use only transport 1 unless unavailable; balance equally between the 2 xTRs.
	21	1	50	
	12	2	50	Fall back to transports 2 and 3 with equal balancing across xTRs and transports.
	13	2	50	
	22	2	50	
	23	2	50	
Prefix B (10.20.20.0 /24)—Application B	12	1	5	Place 1/3 of the flows on transport 2.
	22	1	5	
	13	1	10	Place 2/3 of the flows on transport 3.
	23	1	10	
	21	2	50	Use transport 3 only as a backup.
	11	2	50	

Table 4-1 *Ingress Traffic Engineering Policy*

Up to this point, we've discussed a static policy. The criteria for traffic distribution are augmented with dynamic information, such as the measured performance of the different transport networks. LISP's logically centralized architecture is ideal for the implementation of appropriate application programming interfaces (APIs) to allow the dynamic redefinition of the reachability policies discussed. The ability to program and reprogram the network dynamically through APIs is a cornerstone to any software-defined networking system. In the SD-WAN, a common requirement is to measure performance and reroute traffic based on measured performance. In combination with the multiple transport networks, the measurement of performance opens the opportunity to incorporate Internet bandwidth in the mix of the hybrid WAN, while still preserving service guarantees by virtue of rerouting traffic based on performance measurement. The priorities and weights in the LISP mappings are adjusted programmatically to route traffic via different underlay transports. This allows different performance measurement systems to drive network behavior in the LISP network.

We've also discussed policies that are centered around destination prefixes. Such policies assume that different applications are delivered by dedicated IP addresses, which is true for the majority of deployed applications. However, in the software as a service (SaaS) cloud model of application delivery, different applications may share an IP address. In the SaaS model, applications are differentiated from each other by Uniform Resource Locator (URL) rather than IP address. In this scenario, the SD-WAN edge device must decide how to route traffic on a per flow basis, and destination IP is not a sufficient condition to classify traffic into distinct flows. Deep packet inspection is used in these scenarios to identify flows and map them to a tuple of source and destination IP parameters that uniquely identify a flow. To enforce any policy that is flow specific, you must enhance the ingress tunnel routers (ITRs) to override the traditional destination prefix-based behavior and augment it to be based on the full tuple of Layer 3 and Layer 4 parameters in the IP header. In a LISP system, you achieve

https://t.me/learningnets

this by augmenting the forwarding logic in the ITRs to allow an override of the preferred RLOCs based on EID source and destination parameters up to the L4 header (User Datagram Protocol [UDP] or Transmission Control Protocol [TCP] port numbers). Source parameters are evaluated at forwarding time by using additional forwarding tables, such as those used for the enforcement of access control lists (ACLs). These tables use more expensive memory and are therefore smaller than the main forwarding table. This is a form of policy-based routing (PBR) that has as an action the choice of specific RLOCs for the steering of traffic over a particular transport. The forwarding tables in the ITRs are altered programmatically based on performance measurements, similar to how it is suggested that you can alter the main prefix-based mappings. This simply offers an additional level of granularity in the policy. This behavior is possible in any overlay system and is not a specific function of LISP. In the specific case of a LISP-based overlay, the action is simply to override the preferred RLOCs and encapsulate traffic that matches the source plus destination criteria to the overriding RLOCs.

An alternative approach to programming a local PBR table to override the traditional destination EID mappings is to extend the definition of the EIDs that LISP is handling and integrate the per flow mappings into the LISP control plane. EIDs are extended to be the full tuple of source and destination IP parameters rather than just the destination IP. When you do this, the specific mappings for a specific flow are registered in the LISP control plane dynamically. The extension of the EID definition is easily achievable in the control plane using the LISP Canonical Address Format (LCAF), which effectively enables the LISP Mapping Database System to act as a general key/value store. The logic for Map-Registrations and Map-Requests has to be enhanced to concurrently handle destination IP EID mappings and “flow” EID mappings (inclusive of source and Layer 4 information). The ITRs also must be enhanced to make forwarding decisions on an extended tuple of source and destination parameters or data plane labels (for example, IPv6 flow-labels, quality of service [QoS] Differentiated Services Code Points [DSCP], type of service [ToS], or class of service [CoS] tags) that identify the flow. The enhancement of the forwarding plane is probably the tallest order in this list of enhancements because it basically implies that the ITRs are flow routers, and currently, these are expensive devices to build. The benefit of integrating the flow policies in the LISP control plane is that it consolidates all forwarding behavior into the network control plane versus using a controller to override the base network behavior.

SCALE CONSIDERATIONS

WANs interconnect a large number of sites. In an OTT WAN service, a given site

https://t.me/learningnets

Technet24

participating in the WAN can have a large distribution of connections. The large distribution of connections requires the WAN edge devices to support a large amount of adjacency state. This adjacency state is one of the main limiting factors in scaling the control plane of an OTT WAN service. As illustrated in Figure 4-6, to achieve full any-to-any connectivity, each participating router in the OTT WAN keeps at least one connection for each remote site. An alternative to this approach is also illustrated in Figure 4-5 and uses a partial hub-and-spoke topology to limit the large-scale requirement to fewer edge routers at a few hub locations.

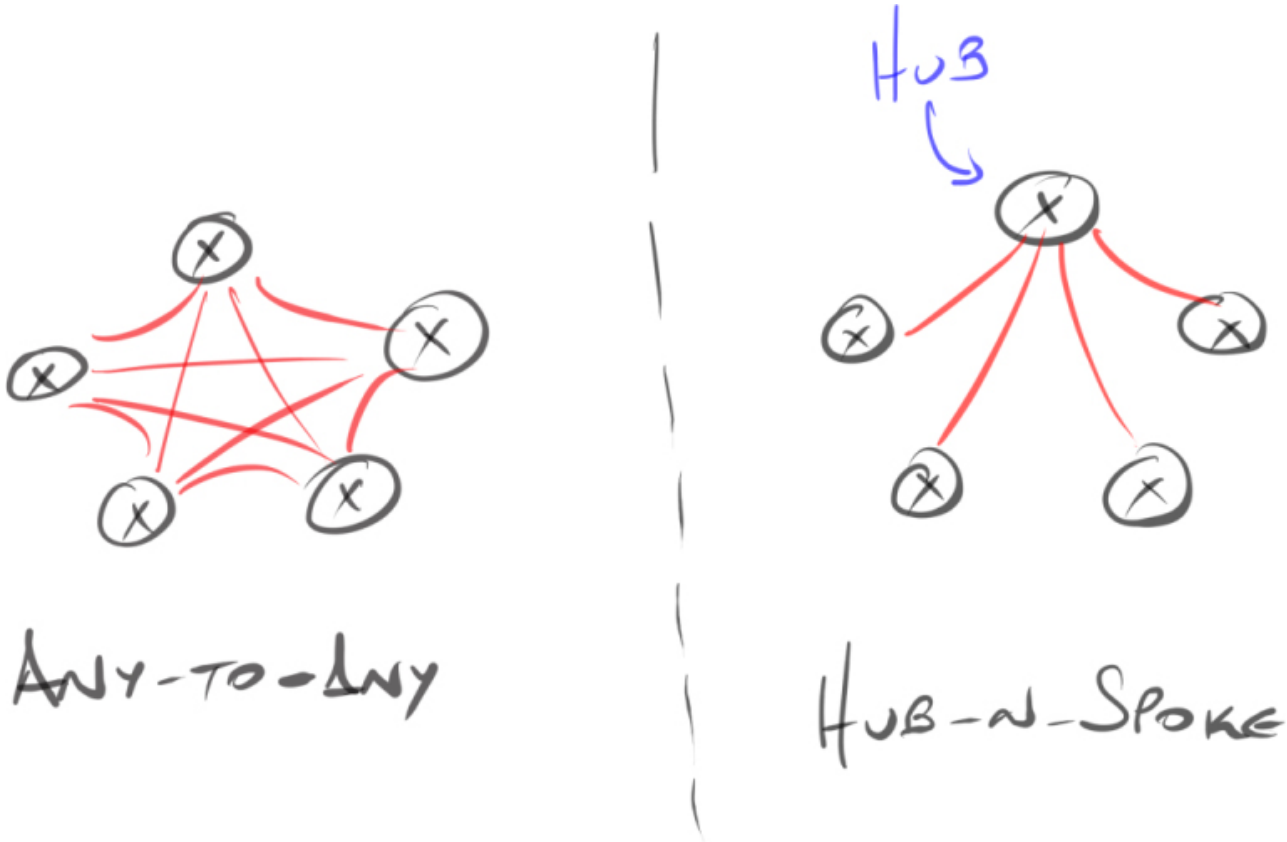


Figure 4-6 *WAN Distribution Scenarios*

In a system using traditional routing mechanisms, the scale challenge stems from the need to maintain routing adjacencies with a large number of peers. Each adjacency has associated state information and requires periodic processing. At large scale, this has a high cost in terms of memory and compute capacity on the routers. Scale for routers with good capacity is around the hundreds of adjacencies. The scale implications are amplified when VRF-based segmentation is included as part of the solution. Each VRF has its own set of adjacencies that count against the overall scale achievable. For instance, if a router supports 500 adjacencies, this translates to 500 sites when there is a single VRF but only 250 sites if a second VRF is supported, or 50 sites when 10 VRFs are required. Basically, each virtual private network requires its own distribution of adjacencies. Multiprotocol-BGP (MP-BGP) uses the virtual private network address family identifier (AFI) to reduce the number of adjacencies. Although this helps the

https://t.me/learningnets

scaling issues, processing overhead is still associated with the multiplexing and demultiplexing of updates into different VPNs. The net effect is similar to that of having an adjacency per VRF.

In contrast with this scenario, LISP does not maintain adjacency state among its xTRs and is therefore able to accommodate distributions of virtually any size. We previously discussed in detail why LISP is not considered to be a routing protocol but a directory service. Because the control plane does not maintain a connection with each xTR, a LISP domain may have thousands of xTRs without them being aware of each other. State is created at an xTR only when that xTR needs to service traffic to a destination attached to another xTR. Even for established connections, the forwarding state required on the xTR imposes a minimal burden on the xTR's processing and memory resources. Furthermore, the handling of multiple VRFs does not require more or less resources in the control plane or the forwarding plane. Because LISP is a directory service, the VRF identifier is simply handled as an attribute of the EID. Refer to the discussion on extended EIDs (XEIDs) in Chapter 3, "Data Center Trends," which describes the concept of XEIDs for the creation of multiple virtual private networks. Thus, multiple VRFs do not burden the control plane any more than multiple EIDs. So, a demand-based control plane such as LISP removes some of the architectural constraints that traditional routing protocols had imposed on OTT logical topologies for WAN use and allow the logical networks to scale many orders of magnitude more.

It is also important to note that the xTRs need nothing more from the underlay core than simple packet delivery. So, in a segmented network, the underlay does not need to participate in the segmentation. That is, no VRFs are instantiated in the core.

LOGICAL TOPOLOGIES: PEER-TO-PEER CONNECTIVITY AND SERVICE INSERTION

The scale considerations discussed so far steered the industry toward hub-and-spoke topologies in the WAN. In addition to scale, another consideration in pursuing these hub-and-spoke topologies was the ability to insert network services at the hub. For example, it may be desirable to drive all traffic through a firewall. Placing firewalls at the hub is a way to guarantee that traffic is steered through the firewall without having to add more sophisticated logic to engineer traffic through these network services. However, this static logical topology constrains what can and cannot be done in the WAN.

Most traffic in the WAN is traffic from connections between a user site and a data center, and when all paths to the data center are via the hub, you could argue that a

https://t.me/learningnets

hub-and-spoke topology works relatively well. Hub-and-spoke topologies have been the de facto design for the WAN, yet they compromise the ability to utilize more direct and efficient paths between the users and the data centers. The hub-and-spoke model also forces peer-to-peer user traffic to traverse the hub, utilizing expensive hub bandwidth and router capacity to forward peer-to-peer traffic that could have gone directly between the WAN sites rather than traversing the hub.

As you move into a more decentralized world, spoke-to-spoke or peer-to-peer becomes more pervasive. In a modern WAN, traffic should not be constrained to a hub-and-spoke topology unless this is indeed the desired traffic behavior. The overlay used in the WAN must allow any-to-any connectivity at scale. As discussed, the connectionless and demand-based approach that LISP follows allows the LISP overlay to scale to a large number of sites. This scale is independent of the desired logical topology, which, by default in the LISP overlay, is an any-to-any topology. So, LISP delivers an any-to-any logical network, unless the logical topology is further constrained by policy.

One of the salient benefits of a software-defined networking approach is the ability to programmatically define network behavior. The LISP Mapping Database System provides a logically centralized control point to enable the creation of logical topologies in the WAN. If you want to constrain connectivity to a subset of the WAN sites and emulate a hub-and-spoke topology, you can program the LISP Mapping Database System to deliver such a topology. The logic to achieve constrained virtual topologies is centered around enforcing lookup policies in the Mapping Database System. For instance, if you want a hub-and-spoke logical topology, the policy in the Mapping Database System simply checks which RLOC the Map-Request is being sourced from and issues a Map-Reply that steers the traffic via the logical topology. For example, in the hub-and-spoke scenario, a Map-Request from a spoke RLOC, for an EID at a different spoke, results in a Map-Reply with the hub RLOCs, making the ITR send the traffic to the hub rather than directly to the destination spoke RLOC. A Map-Request from a hub RLOC results in a Map-Reply with the RLOCs of spoke, where the destination EID is indeed connected to the LISP network.

Although the flexibility to create logical topologies is interesting, much more compelling is the idea that you can use the same mechanisms you use to steer traffic via a logical topology to dynamically steer traffic via service function nodes such as firewalls, regardless of their location. Rather than focusing the design and operation of the network on creating a logical topology, you can focus it on optimal connectivity and the dynamic insertion of network services. As network services become virtualized, the use of hardware appliances to deliver these services in a central location gives way to a set of distributed virtual instances of the service. As services are virtualized, the service

https://t.me/learningnets

instances are distributed across the network and instantiated dynamically so they can service the connections that would naturally flow closest to the virtual service instances. Because the service enforcement points are distributed, centrally deployed high-performance appliances are less relevant, and the presence of the service becomes ubiquitous across the network. Additionally, removing the dependency on a hub site for the insertion of services significantly improves the resiliency and reliability of the network by eliminating the hub as a single point of failure. The static logical topologies used previously are now replaced by flow-specific paths through virtual service instances. These paths can be made optimal by integrating the LISP control plane with the service orchestration software and make the insertion of a virtual service instance simply part of the path provided by the LISP overlay to get traffic from source to destination. Figure 4-7 illustrates the workflow by which services are dynamically instantiated and inserted on a path by the LISP control plane.

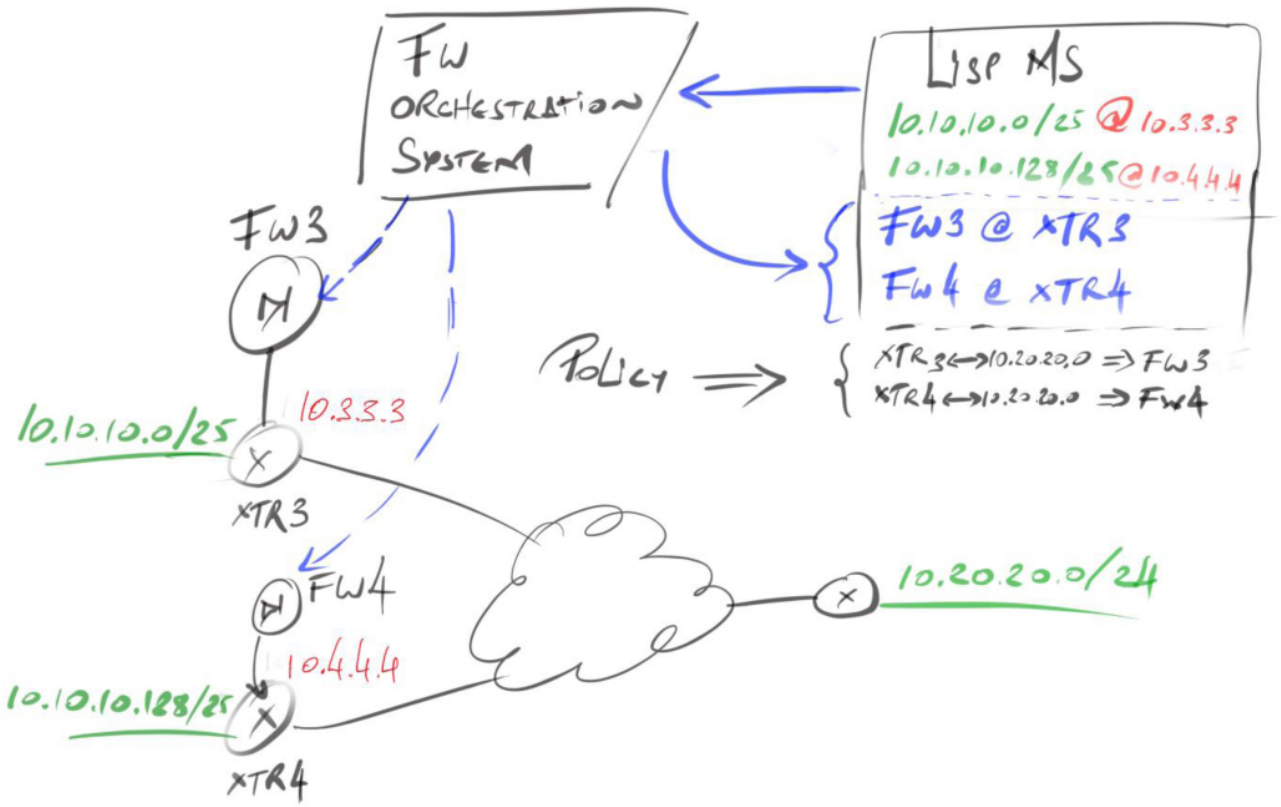


Figure 4-7 *Service Insertion Workflow*

In the system shown in Figure 4-6, a WAN is running LISP, and a virtual firewall system can instantiate virtual firewalls on the WAN edge routers. It all starts with a policy. In this example, the policy is to allow email traffic to the data center and also allow peer-to-peer voice traffic; all other traffic is not allowed. The email traffic is required to traverse a firewall, while the voice traffic is not. In this example, the policy is articulated in terms of prefixes, although in some implementations, prefixes are abstracted as groups of endpoints. Say that all users use IP prefix 10.10.10.0 /24 and that the email servers are all in prefix 10.20.20.0 / 24. Table 4-2 illustrates the

https://t.me/learningnets

Technet24

intended policy in terms of classifiers and actions. The classifiers further specify the type of traffic, and the actions express the actual policy (permit through a firewall, deny, and so on).

Source	Destination	Classifier	Action
10.10.10.0 /24	10.20.20.0 /24	email (SMTP)	Permit Stateful
10.10.10.0 /24	10.10.10.0 /24	Voice (SIP, UDP-x)	Permit Stateless
Any	Any	Any	Deny (stateless)

Table 4-2 *Connectivity Policies*

As EIDs register with the LISP Mapping Database System, there is awareness of where endpoints are located. Virtual service instances may be instantiated in preparation for future flows based on that location awareness. To do so, the network management system may request the service orchestration system to instantiate virtual services close to (or at) the routers with the RLOCs for the EIDs involved in the policy. In this example, routers 3 and 4 register EIDs in the 10.10.10.0 /24 range with RLOCs 10.3.3.3 and 10.4.4.4, respectively. The location information for the EIDs is used to inform the services orchestrator where to instantiate virtual services. When the orchestrator instantiates the services, it responds to the network management system with the location (RLOC) for the instantiated services. This allows the management system to include these service instances in the service chaining policy of the LISP Mapping Database System. The service chaining policy defines different service insertion RLOCs for different requesting RLOCs. The flows initiated from RLOCs 10.3.3.3 and 10.4.4.4 are chained through FW instances 3 and 4, respectively. Firewall 4 in Figure 4-7 illustrates how virtual firewalls are instantiated at the xTRs themselves because some routers offer the ability to host applications locally. The latter simplifies the chaining of services because a redirection of the traffic by the overlay is not required.

Virtual service instances may also be instantiated and inserted on demand as connections are established. This requires small changes to the workflow described for the preprovisioning of services.

With the ability to provide any-to-any connectivity at scale and dynamic service insertion, the LISP overlay enables the flexibility to access any data center or Internet egress point without having to predefine a rigid topology.

SECURITY: CONNECTION INTEGRITY AND CONFIDENTIALITY

Integrity and confidentiality are important aspects of the WAN service, particularly for those connections leveraging the public Internet. A wealth of OTT solutions leverage point-to-point IPsec tunnels for cryptography over public networks; these solutions are

https://t.me/learningnets

limited in their scale for similar reasons to those that limit large distributions of traditional routing adjacencies. In the case of cryptography, the gating factor is the number of security associations that must be maintained at every edge router. One proposal geared toward alleviating these scale limitations uses group keys to reduce the number of required security associations. An example of a solution using shared keys is GETVPN, where a Group Domain of Interpretation (GDOI) centralized key server is used to distribute a shared key that all participants in the VPN use. GETVPN is usually run only over private networks for two main reasons: (1) The source and destination addresses for the tunnels must be distributed into the transit network; and (2) the security level of a shared key on the public Internet is questionable. LISP can be combined with GETVPN to address problem (1); however, the industry requires much tighter security than shared keys, and current security practices require pairwise keys that are unique to any combination of RLOCs involved in a connection.

LISP offers an elegant solution to the problem of exchanging information for the generation of pairwise keys and scaling the security association state required to provide pairwise cryptography in the WAN. The security key exchange negotiation available in LISP is fully decentralized. There is no need for a third party to partake in realizing key management, and keys are not stored anywhere, providing easier rekeying and better avoidance of key cracking.

The mechanics of LISP-based cryptography are explored in detail in Chapter 6, "Security." RFC-8061 describes the procedures that integrate cryptographic key material negotiation into the LISP control plane. The mechanisms simply enhance the map-resolution process to allow the exchange of cryptographic key material during the map-resolution process. Because the map-resolution process involves the ITRs and ETRs, which are the endpoints to the connection, private key calculation can take place at the connection endpoints. Cryptographic keys are thus calculated on demand, as the establishment of new connections triggers requests for EID resolution. The amount of cryptography state created is strictly the state required to support established connections between the relevant xTRs; this demand-based model for pairwise key calculation is rather unique to LISP.

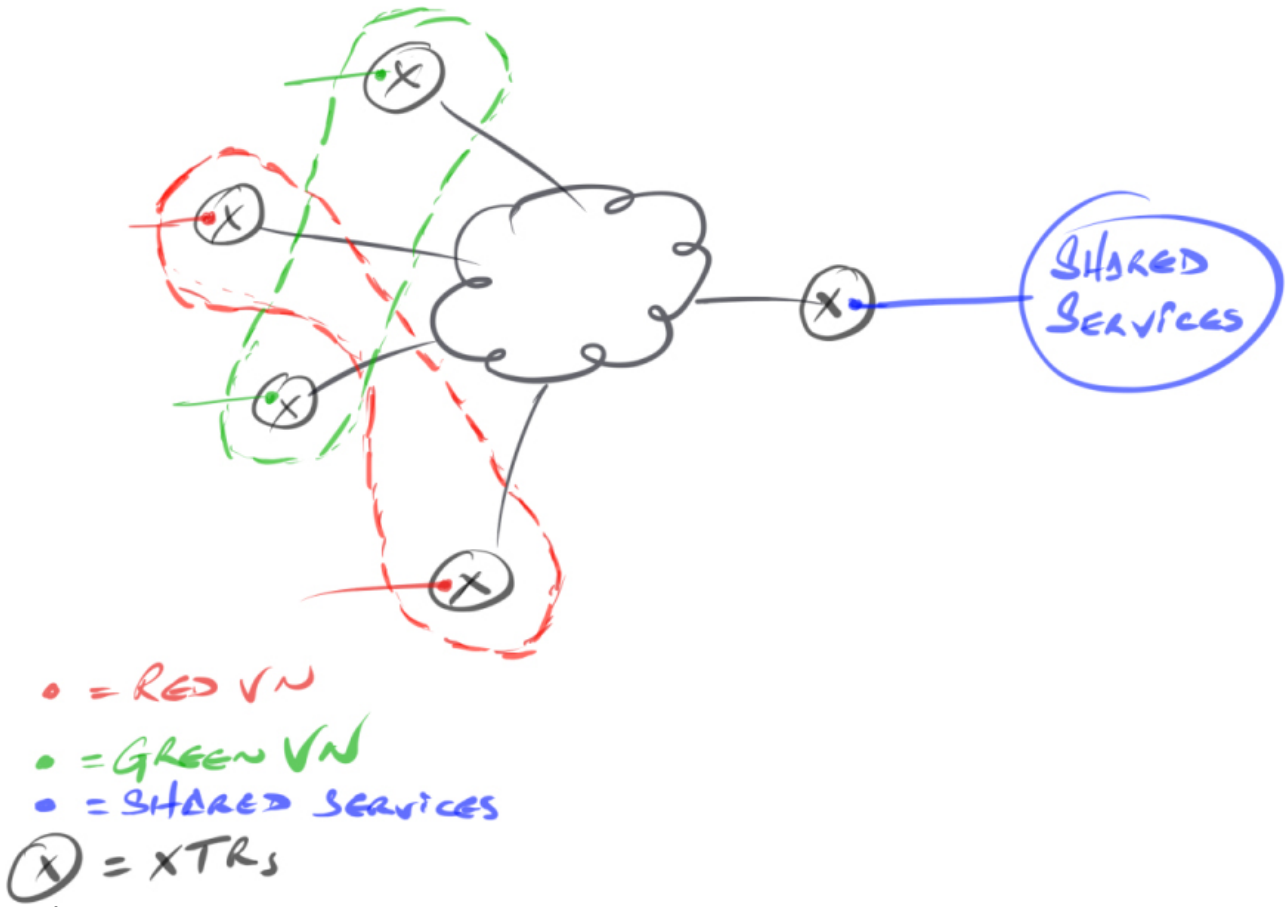
SEGMENTATION

Chapter 3 described in detail the mechanisms by which segmentation is provided in a LISP network. Basically, the xTRs are segmented into different forwarding contexts (using VRFs or bridge-domains, for instance), and EIDs are extended (XEID) to include an Instance-ID (IID) that associates an EID to a particular forwarding context. When you use the XEID in all control plane messages and include the IID in the data plane

https://t.me/learningnets
Technet24

headers, both control and data plane are scoped to a particular VRF (or bridge-domain) to effectively create network segments. The LISP-enabled WAN leverages these mechanisms to deliver segmentation and create virtual private networks (VPNs).

In the WAN, the scope of a particular VPN is often limited to a subset of the WAN sites. In many cases, a site participates in a single VPN while other sites participate in other VPNs, so each site has to hold state only for the VPNs that it participates in. The different VPNs may coincide at certain sites, particularly at sites where shared services are accessible. The LISP segmentation architecture provides an efficient mechanism to provide connectivity to shared services in a system with multiple VPNs. The mechanism, referred to as extranet VPN, is described in detail in Chapter 3. One key advantage of the LISP extranet VPN functionality is that it allows a VPN to access resources in a shared VPN without requiring the instantiation of the shared VPN at the requesting ITR. In the case of the WAN, for sites participating in a single VPN, this implies that the xTRs continue to handle a single VPN, even if they need access to shared services that exist in a different VPN. The way LISP provides extranet VPN services is in contrast to how currently cross-VPN communication is handled in protocols like BGP, which require that all VPNs be instantiated at all sites from where they may need to be reached. Furthermore, it requires that routes be replicated across the VPNs. In a way, the LISP extranet VPN functionality could be considered stateless. Figure 4-8 illustrates the footprint of the VPN across the WAN in a LISP overlay network.



https://t.me/learningnets

Figure 4-8 *Virtual private network Footprint and Extranet Access to Shared Services*

THE ACCESS NETWORK: MULTISITE CONSIDERATIONS

To connect users to their applications, multiple networks must be stitched together. The wide-area network must interface with the access networks built at the different campus and branch sites that the WAN interconnects and must provide clean and effective handoff points to the data centers where applications and data reside. Some may argue that this is all really one large network that includes user access, WAN services, as well as the insertion of security services and handoff to the data center. Most agree with this view, and furthermore, using LISP across all these domains offers distinct advantages. LISP's holistic architecture allows the normalization of all the different network domains to a single technology for the delivery of overlays and all the necessary functionality those overlays must deliver. Historically, there were different fragmented solutions for premise-based overlays, SD-WAN overlays, cloud VM, container overlays, and possibly the same for Internet of Things (IoT) overlays. In the future, you will need only one technology to bring all these disparate domains together. The same can be observed in the cellular and future 5G network, where the carriers have their own overlays based on technologies such as the General Packet Radio Service Tunneling Protocol (GTP).

Chapter 3 discussed how a switched network uses LISP to provide optimized segmentation and mobility at scale. The requirements for segmentation and mobility are not unique to the data center but are prevalent in the campus and branch access network. In fact, supporting user and device mobility in the access network may be more challenging than supporting mobility of data center workloads; more on that in Chapter 5, "Mega-Scale Access Networks: LISP, User Access, and the Internet of Things." We also described the different models utilized to connect the access network to the data center locations in Chapter 3. At the heart of connecting users to data center-hosted applications is the wide-area network. There are distinct advantages to the use of LISP both in the site networks built in campus and branch as well as in the wide-area network that connects those sites to the data centers. The largest benefit is realized when both access and WAN utilize LISP in a synergistic way.

Beyond the obvious benefits of normalization of operations that are derived from using a common protocol throughout the network, critical functionality is enabled to help the network evolve and support the next wave of requirements. An end-to-end LISP network enables global mobility while maintaining scalability and fate independence across the different sites and regions that conform the global network. This could be

https://t.me/learningnets

Technet24

architected as a single LISP network where traffic is tunneled from the access switches in the campus/branch directly to the routers that connect to the data center, making the entire structure of campus and WAN transparent to the overlay. To achieve the desired scale and fate independence, the LISP specification provides the Delegated Database Tree (DDT) mechanism as a way to scale out the Mapping Database System and scope the authority for certain EID prefixes to specific Mapping Database Systems and thus achieve fate independence across regions. DDT is specified in RFC 8111, where the mechanisms by which a Mapping Database System may be constructed with a multitude of servers organized in a tree are defined. The structure and navigation of the tree are similar to that of the Domain Name System (DNS) tree. Each leaf of the tree is responsible for a particular set of EID prefixes (the tree could also be organized per virtual private network, which in LISP is known as Instances or, more informally, Instance-IDs). DTT is not a replicated database; state is only synchronized locally at the leaves of the tree, which enables the desired fate independence across sites but also reduces the exposure of the Mapping Database System to distributed denial of service (DDoS) attacks. The tree is navigated from the leaves, up the branches, toward the root in search of the branch that leads to the authoritative leaf node for the EID being looked up. Once the leaf is identified, the Map-Resolver forwards the Map-Request directly to the authoritative Map-Server (the authoritative leaf in the tree), and the map resolution process takes place as usual. The use of DDT presents a good way to scale out the network to a broad footprint, and effort has been put into making sure all features in the LISP architecture are supported when the Mapping Database System is structured in a Delegated Database Tree. A multisite network using DDT is illustrated in Figure 4-9.

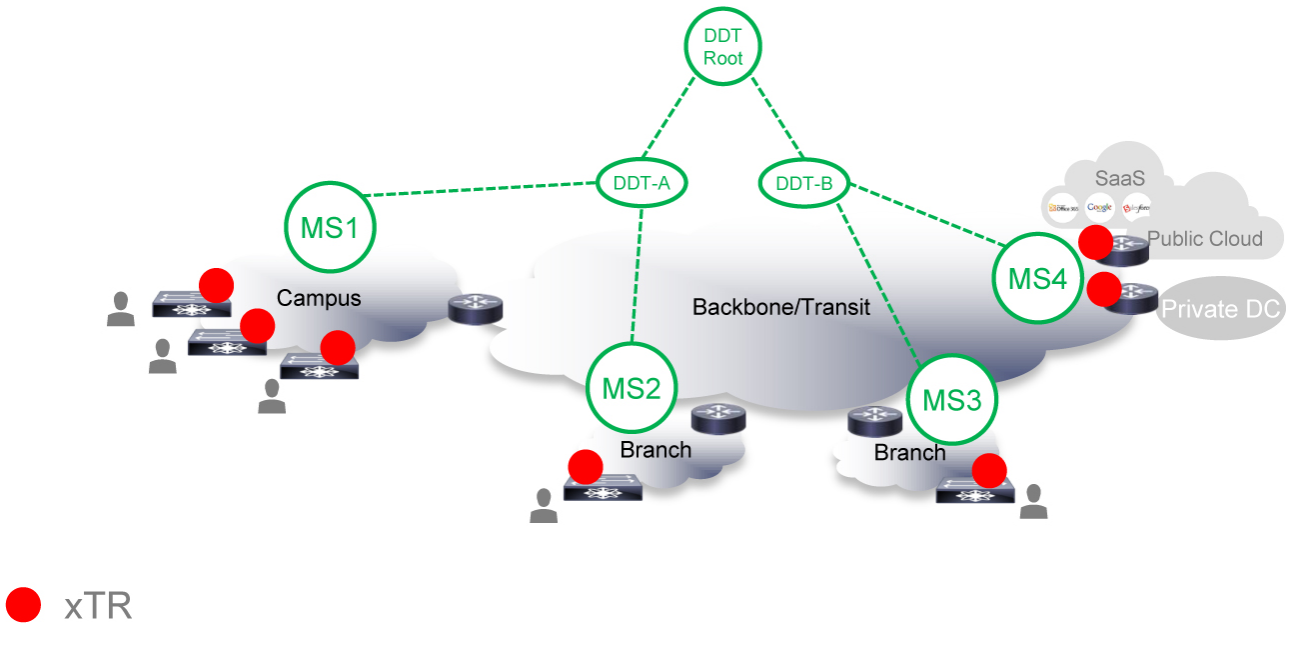


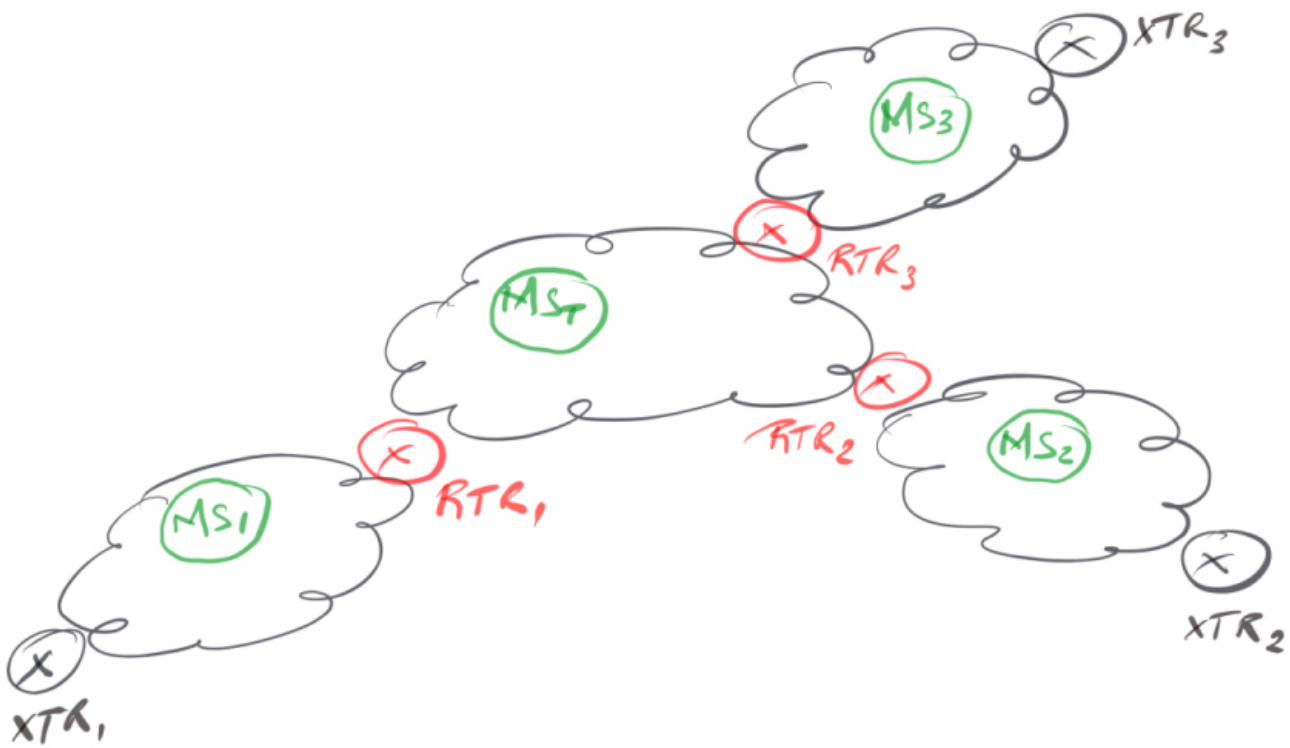
Figure 4-9 *LISP Multisite with a DDT Mapping Database System*

When you are using the DDT model, it may not be possible to support mobility across

https://t.me/learningnets

sites because this would imply variations in the delegation criteria of the DDT tree that are not defined in the specification, nor are they easy to achieve. Mobility in a DDT environment is possible only if the mobile hosts always register with their authoritative DDT node. However, the goal is to allow mobile EIDs to move from the scope of authority of one Mapping Database System node to another and create clearly delineated survivability zones. An alternative design that supports the desired scale, survivability zoning, and also includes mobility is a federated design where multiple LISP domains are connected to each other. This design is structured in a hierarchical manner where the different sites are independent LISP networks and are all interconnected by one or more transit areas, which are LISP networks in themselves. The benefits of such federated design include mobility at large scale with failure containment, as well as the ability to focus the operations of certain functions at the right places in the network.

Thus, a LISP network can be organized as a collection of sites interconnected by one or more transit areas. Both the sites and the transit areas are LISP networks. The sites share re-encapsulating tunnel routers (RTRs) with the transit network, as illustrated in Figure 4-10. RTRs are simply routers that receive LISP-encapsulated traffic, de-encapsulate the traffic, issue a new lookup, and re-encapsulate the traffic. RTRs are instrumental in interconnecting disjoint portions of a network.



LISP MULTI-SITE FEDERATED NETWORK

https://t.me/learningnets
Technet24

Figure 4-10 *LISP Multisite Federated Network*

Each domain (site or transit) has its own Mapping Database System. The RTRs between domains therefore connect to one or more Mapping Database Systems. Mappings are redistributed between the domains at the RTRs to enable EIDs to be reached across domains. These mappings are held and redistributed in the RTRs' soft memory and are not installed in the hardware forwarding tables of the RTRs until an active connection requiring the programming of the mapping into the forwarding tables is initiated. When these networks are treated hierarchically, the sites may function as stub LISP domains, and the transit network acts as the core transit LISP domain. If the sites are indeed stubs, the amount of information exchanged with the sites may be minimized because the sites could simply use a default path rule to steer traffic toward the RTR connecting to the transit LISP domain to reach other sites. In other words, the sites assume default routing toward the transit and do not need the transit prefixes to be registered in their Mapping Database System. Conversely, the prefixes in the different sites do have to be registered to the transit Mapping Database System. The EID prefixes at each site can usually be registered as summary prefixes in the transit LISP domain. When cross-site mobility is at play, the mappings registered in the transit Mapping Database System may be in the form of mappings for host EIDs and therefore numerous. Because the information across domains is redistributed at the border RTRs, the local site Mapping Database System must provide the border RTR with a list of all registered EIDs. Publish-subscribe mechanisms defined in the LISP architecture are designed for these purposes. Basically, the site Mapping Database System publishes all its registered EIDs to the border RTR.

When federating LISP domains via RTRs, the RTRs are connected to multiple Mapping Database Systems. Thus, the RTRs must be able to determine directionality of the traffic flows being serviced so that they can consult the Mapping Database System that is on path to the desired destination. Different implementations vary in how they achieve this notion of directionality. One simplistic implementation assumes the RTR has knowledge of the prefixes that are present in the site and directs its requests to the transit Mapping Database System only for destinations that it doesn't have listed as being in the local site. Another approach is to send a Map-Request to all Mapping Database Systems to which the RTR is connected and decide which Map-Response to use based on the EID prefix length of the response. This assumes that the Map-Server on path to the destination always has a more specific match (longest prefix) for a mapping to the destination EID. If two or more Mapping Database Systems return a similar length prefix, the responses should be merged into a single mapping with multiple paths as they indicate that all the responding domains have a path to the destination requested.

https://t.me/learningnets

The federated multisite model allows cross-site mobility and scales well in particular scenarios, such as when the sites are stubs and don't require specific mappings to external destinations. In the case where the sites are stubs, the Mapping Database System in the stubs may have to handle only EID prefixes that are locally connected, which is an equivalent scale profile to that of a DDT Map-Server. The Mapping Database System in the transit area, however, has all EID prefixes for all sites registered to it. This is often a manageable number of summary EID prefix mappings from the different sites, but the number of mappings registered in the transit Mapping Database System may require a more scalable Mapping Database System design if mobility between sites is at play and the mappings registered are not for summary EID prefixes, but host mappings.

Because the Mapping Database System in the transit area is independent of the Mapping Database Systems at the different sites, the Mapping Database System for the transit area could be implemented as a DDT in the cases where scale beyond the capacity of a single Mapping Database System node is required. This DDT Mapping Database System has specific prefixes or virtual private networks anchored to a particular Map-Server node, so that the node has to be capacity engineered for all possible host routes in the prefixes (or VPNs) for which it is authoritative. As more prefixes are added, new Map-Server nodes can be added to scale out the capacity of the Mapping Database System in the transit network. The transit network, although using DDT, does not get the benefits of fate isolation that DDT for localized EID prefixes normally delivers but is structured to scale horizontally as the network grows. This hierarchical design is illustrated in Figure 4-11. Note how EIDs for a particular prefix are always registered to the same Map-Server node in the DDT of the transit area, regardless of which site they are in. Also note that the Mapping Database System in each different site handles information only for the local site.

https://t.me/learningnets

Technet24

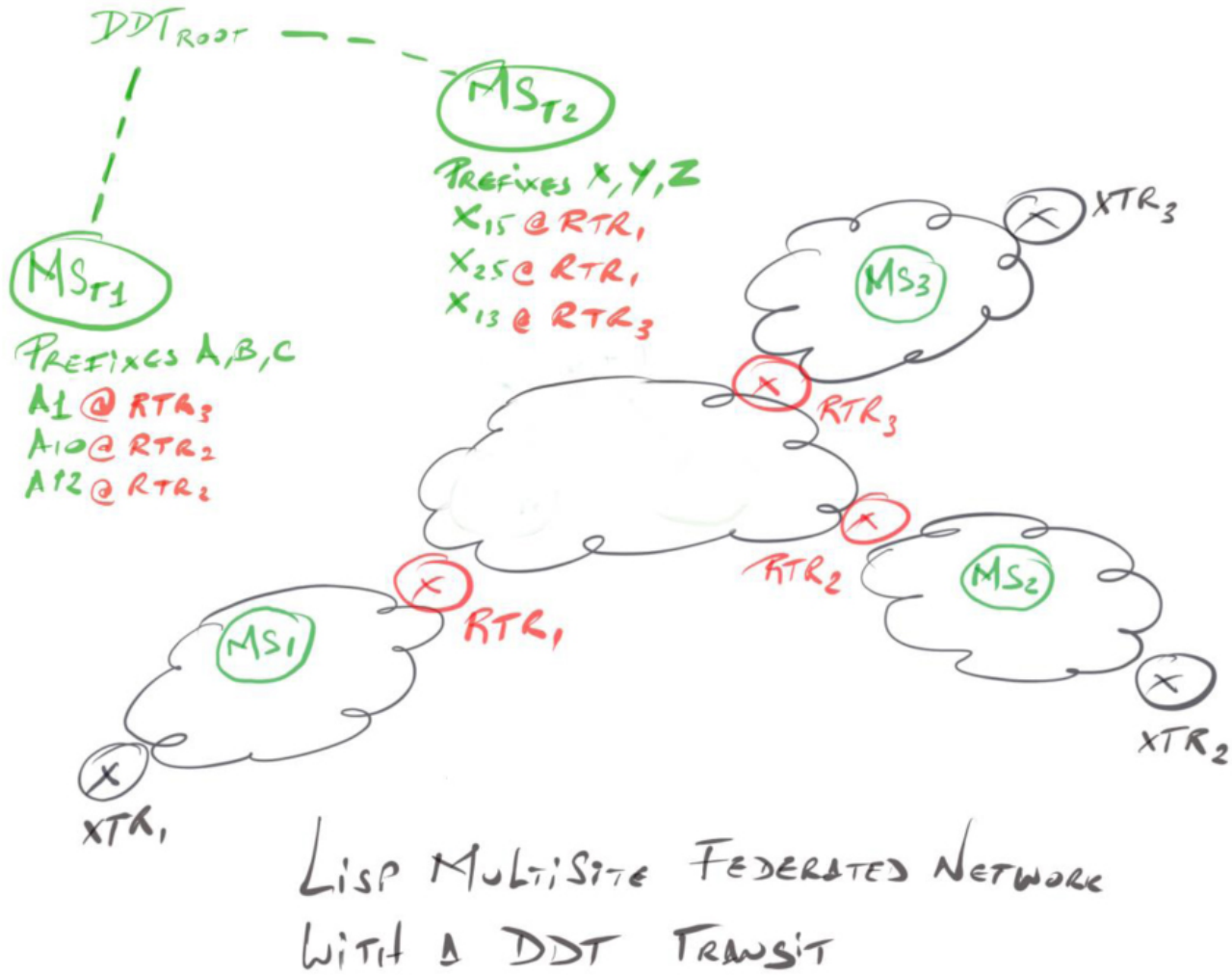


Figure 4-11 *LISP Multisite Federated Network with a DDT Transit*

When the access network is created, it must be extended to the edge of the data center. Chapter 3 discussed connecting to the data center via a cloud neutral facility or installing an overlay endpoint at the cloud facility. Whichever model is used to transport the traffic to the edge of the data center, the connections must be subject to inspection and control so that policy can be applied to these connections accordingly. In most cases, this means that a firewall must be inserted in the flow before handing off the traffic to the data center. If you think of the portion of the network that connects to the data center as a LISP site, one of the services delivered by the LISP network at that site is the insertion of services such as firewalls. Earlier in this chapter, we explained how LISP can insert these services and thus complete the end-to-end solution for large-scale connectivity of the access network to the data center. LISP delivers a single network to manage from Access all the way into the data center, whether it is run privately or sourced from a public cloud service.

MANAGEABILITY

One key requirement is that all these solutions are manageable. Furthermore, the solutions are expected to be highly programmable. LISP provides benefits in two fronts: (1) it simplifies the functionality being deployed by focusing on a directory function for

https://t.me/learningnets

resolution of locations of EIDs, rather than solving a routing problem; and (2) LISP provides a logically centralized point of control in the Mapping Database System that can be accessed programmatically to effect change or obtain information and visibility.

As far as the topics discussed here around the WAN and campus/branch access networks, the consolidated end-to-end network is to be managed as one, without a division between access and WAN. Although the site and transit networks are somewhat independent, from a management perspective, they are kept together as elements of an end-to-end service. Some features and functions may be relevant only to the WAN portion of the network, and others may be relevant only to the access portion of the network, but the management ownership of the devices should be ideally maintained under a single authority. The LISP architecture is structured to support this notion of unified management. A normalized set of policies for traffic steering, logical topology creation, and service insertion allows a common management model to be leveraged across the entire end-to-end span and provides a comprehensive interface for both programmability as well as visibility of the end-to-end network behavior and state.

SUMMARY

LISP intrinsically provides most of the benefits expected from a software-defined WAN. The technology enables some fundamental shifts in how networking is done in the access and wide-area networks. These foundational changes are the cornerstone of new functionality that allows the network to deliver end-to-end segmentation at a large scale inclusive of mobility and encryption in a model that is architected for easy programmability. LISP enables the network architect to roll out what is effectively one network to bring connectivity all the way from the access to the data center.

https://t.me/learningnets

Technet24

https://avxhm.se/blogs/hill0

Chapter 5. Mega-Scale Access Networks: LISP, User Access, and the Internet of Things

Since the early 20th century, the notion of automating mundane tasks by having things of moderate "intelligence" interconnected evolved from being just an idea to becoming a practical reality. The pervasiveness of wireless network connectivity and the availability of affordable processing power in a small form factor enabled implementations of a growing number of automation solutions. Some of the leading applications in this area include building automation, industrial control systems, connected vehicles, asset tracking, item inventory, wearable sensors, and the list goes on. The connected mobile computing footprint that enables these applications was dubbed the *Internet of Things* (IoT). Because the Internet is best known for its public footprint, and these applications take place over both public and private networks, a more accurate moniker is probably the *Network of Things*. Regardless of the affordability of processing power, these continue to be "things." As such, not only are they many, but they also need to remain cheap, and *affordable* processing power does not mean *free* processing power. The network plays a key role in supplementing these things with connectivity, privacy, and even location services. Anything that can be securely offloaded from the connected things will save costs and therefore be a candidate for handling elsewhere. The guiding principle is that things are dumb, which requires the network to be smart.

In parallel to the Network of Things, mobile computing is now the norm. The average person owns and operates between three and four different mobile devices in the form of phones, tablets, or personal computers. All these mobile computers are connected to the data network, and most of them have more processing power than the super computers of the late 20th century. On May 19, 1997, IBM's Deep Blue supercomputer defeated Garry Kasparov, the legendary world chess champion, marking a historic milestone in the evolution of artificial intelligence. Deep Blue wasn't the most powerful computer at the time, but it was the most famous; it had a compute power of 11.38 Giga-FLOPS (floating operations per second). By 2014, an off-the-shelf smartphone delivered more than ten times this performance (some smartphones had a processing output of 142 Giga-FLOPS). As of this writing, smartphone performance has more than

https://t.me/learningnets

doubled from the 2014 reference. Whether you look at a phone that has more computing power than the conglomerate of mainframes that supported NASA's lunar mission in 1969 or simply look at a sensor, the practical applications that leverage the intelligence of all these things depends mainly on the ability to have all these things connected. This dependency on connectivity puts the network at the heart of the automation of this world. In addition to basic connectivity, the network can offload important functions such as privacy and location services from the processors on the connected things. Furthermore, the requirements for connecting mobile computing nodes and things are similar enough that a single access network can be used to support user and device access alongside the connection of things.

With a world population of more than 7 billion people, and an average estimate of 3.5 compute devices per IT user, although the entire world is not connected to the network, the number of mobile computing devices is extremely large. In addition to the large number of mobile computers connected to the network, the number of things connected to the network is expected to be in the tens of billions. Some expect the numbers to grow into the hundreds of billions. A large majority of these devices will connect to the network wirelessly, yet the number of wired connections cannot be neglected. Wired and wireless networks must converge.

The converged wired and wireless access network that is to support this massive number of mobile computers and things must be architected to support mobility at unprecedented scale. Traditional IP networks are not particularly optimized for mobility; the entire scalability model in a traditional IP network is built around IP address summarization. When hosts are mobile, IP addresses move across these summarization boundaries, making the scale model based on IP summarization impractical. Chapter 7, "LISP and the Next-Generation Mobile Network," explores a multitude of scenarios in which IP addresses move across the summarization boundary and how LISP enables this mobility. Some examples include connected cities hosting

- Connected vehicles
- Connected aircraft
- Portable field operations units
- Orbital compute networks
- The future 5G mobile network

The constraint of mobility around the IP summarization boundaries drove operators to

https://t.me/learningnets

Technet24

design mobile networks as separate and independent networks from the IP-based Internet. The mobile networks and the IP networks connect at specific handoff points and remain separate. These models must converge, and the Locator/ID Separation Protocol (LISP) offers the necessary mechanisms to enable such convergence by enabling large-scale mobility independent of summarization boundaries.

An interesting aspect of the required connectivity among the things that make this world is that, although they may need to have the option to connect to each other, this option is not usually exercised. Things usually connect to other things and compute resources that are relevant to the task they automate, but they don't usually connect to everything. The implication on the network is rather interesting because the network must be ready to service any connection but would need to limit the state it handles to only active connections. Traditional routing protocols would propagate all information for all possible connections to every network device, making the state for all connections existing everywhere. LISP, in contrast, will allow network devices to request connectivity information only when they need it.

LISP's demand model is key in allowing the convergence of the mobile and IP networks at the scale levels required to connect the things in this world. This pervasive connectivity is the key to automate the repetitive, gain insights into our environment, and delegate mundane work tasks to robots and things rather than people.

ACCESS NETWORKS USING LISP

The access network must support an unprecedented number of endpoints. The endpoints will connect wirelessly or will be hardwired. The treatment of these endpoints should be indistinguishable whether they are wired or wireless. The endpoints will be highly mobile and, in many cases, will be basic and lack sophisticated software or even a rich networking stack. Endpoints will range from capable mobile computers (such as smartphones) to basic utility elements such as lightbulbs or sensors where high volume dictates a strong appetite for reducing cost, which tends to result in a reduction in processing power and sophistication. A small OS footprint on the connected devices is also one possible way to limit the attack surface available in such devices. An extreme case would be the production of a connected lightbulb where even the IP stack of the device is hard-coded to avoid having to run protocols such as the Dynamic Host Configuration Protocol (DHCP) or the Address Resolution Protocol (ARP) that are designed to enable IP address variability. The right access network would not require hosts to handle this IP variability, but it would dictate that the network adapts to hard-coded addresses. Simplifying the connectivity of the host to this level will have a significant impact on the operations of the access network and the cost

https://t.me/learningnets

of running these networks. So, the task for the access network is one of supporting a massive number of endpoints, allowing the endpoints to be mobile, and offloading network intelligence from the connected devices so that the devices and their network management can be simplified as much as possible. As for how the devices connect to the network, whether the interface is wired or wireless, management, security, and policy enforcement should be holistically common regardless of how devices are connected. It is also worth noting that these devices will be multihomed because the data they are reporting may be mission critical, and because their storage capacities are limited, they need network access.

LISP Access Network Design

Traditionally, access networks were designed around the constraints imposed by the technologies and protocols used to build the network. Furthermore, networks must be structured to accommodate any mobility required within the limitations of the networking technologies used to build the access network. By using a LISP-based overlay, the topological design of the network is made independent of the scope of endpoint mobility as well as the properties of the technologies in use.

Access networks were traditionally built with a combination of Layer 2 technologies, such as Spanning Tree and Link-Aggregation, and Internal Gateway Protocols (IGPs) such as Open Shortest Path First (OSPF), Intermediate System to Intermediate System (IS-IS), or Enhanced Interior Gateway Routing Protocol (EIGRP). In general, the network is structured in a series of distribution blocks that are interconnected by a core. This dictates the topology as well as the protocol layout of the network. The topology also dictates the scope of mobility that is available to endpoints. Figure 5-1 illustrates a typical Layer 2 access design that aggregates onto a routed core. Note the combination of technologies and the restricted scope of mobility in these networks. Moving forward, mobility and address tracking will be critical to many applications. In such environments, addresses cannot be topologically assigned, and traditional network models may not fulfill the requirements.

https://t.me/learningnets

Technet24

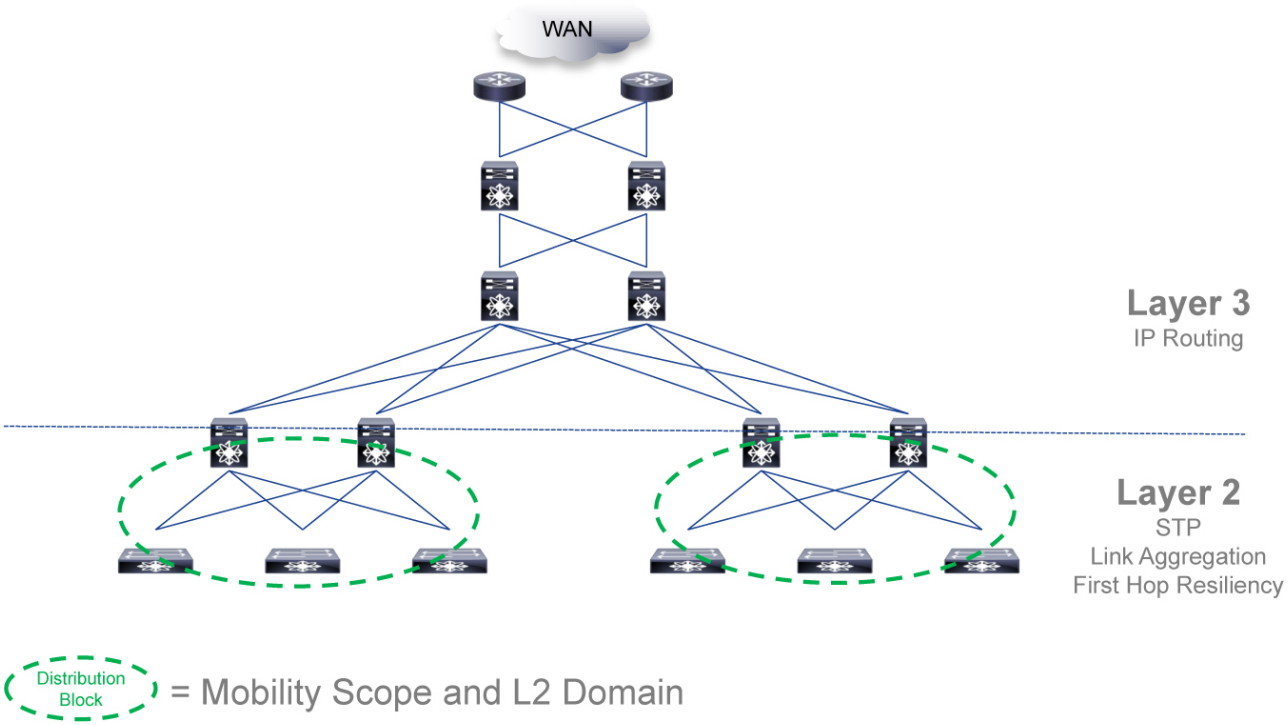


Figure 5-1 *Standard Layer 2 Access Network Design*

The combination of protocols complicates operations, and the exposure to Layer 2 failures impacts stability. An alternative design uses a routed access. In a routed access, all tiers are routed, and exposure to Layer 2 failures is eliminated. Figure 5-2 illustrates the routed access design.

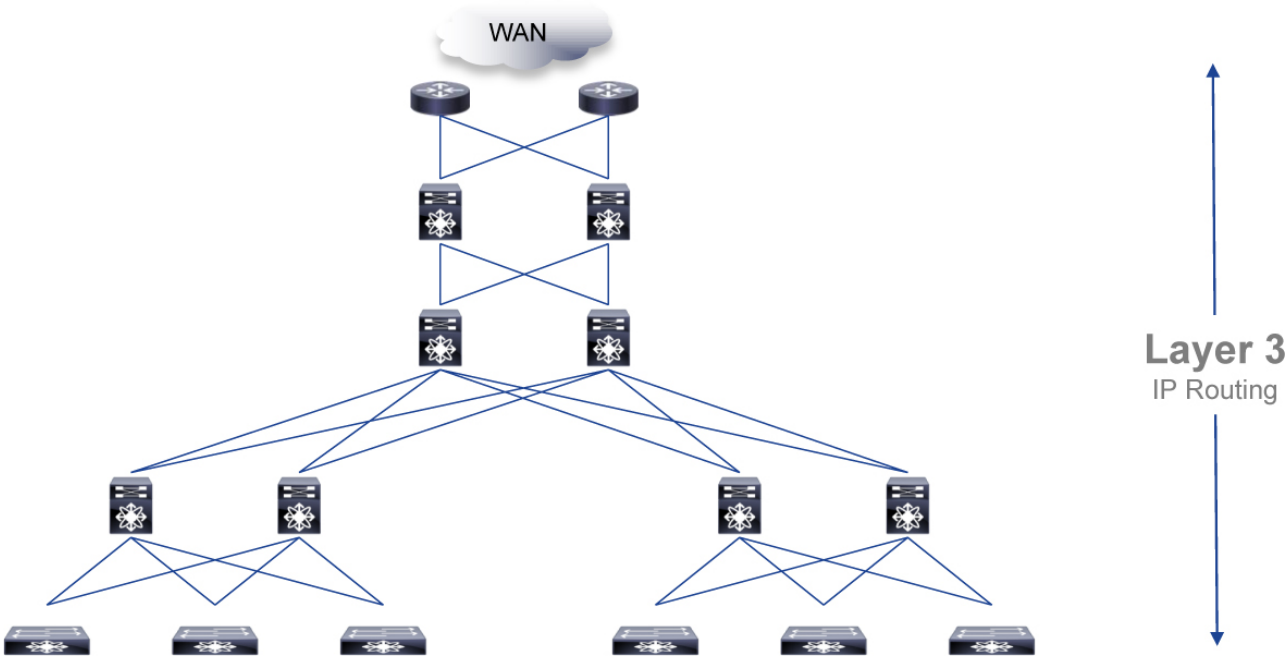


Figure 5-2 *Routed Access Network Design*

Without any enhancements to the existing Layer 3 protocols, the routed access design does not provide any support for mobility. Traditionally, in routed access networks, mobility was supported exclusively for wireless endpoints and in the form of a Layer 2 overlay. This overlay is, for all practical purposes, a separate network from the routed

https://t.me/learningnets

access network. Because the routed access network does not support mobility and all nodes behave similarly, no topological restrictions constrain the deployment of a routed network. There are, of course, best practices focused on providing symmetric redundant paths and convergence optimization through summarization, but these practices are independent of any service requirements such as mobility.

To support mobility, improve scalability, optimize network stability, and deliver network segmentation, you can use a LISP overlay network in the access network. The LISP overlay network leverages a well-designed routed access network as its underlay network to maximize stability, provide resiliency, and optimize convergence. However, the design of the overlay network is independent of the underlay network optimizations. The overlay network design is rather simplistic; just three roles for the devices are involved in delivering the overlay network:

• **Edge node:** A tunnel router (xTR) to which endpoints connect directly

• **Control plane node:** A Map-Server/Map-Resolver node

• **Border node:** An xTR or a proxy tunnel router (PxTR) connecting to an adjacent network

Figure 5-3 shows how these overlay roles are deployed on a generic IP network.

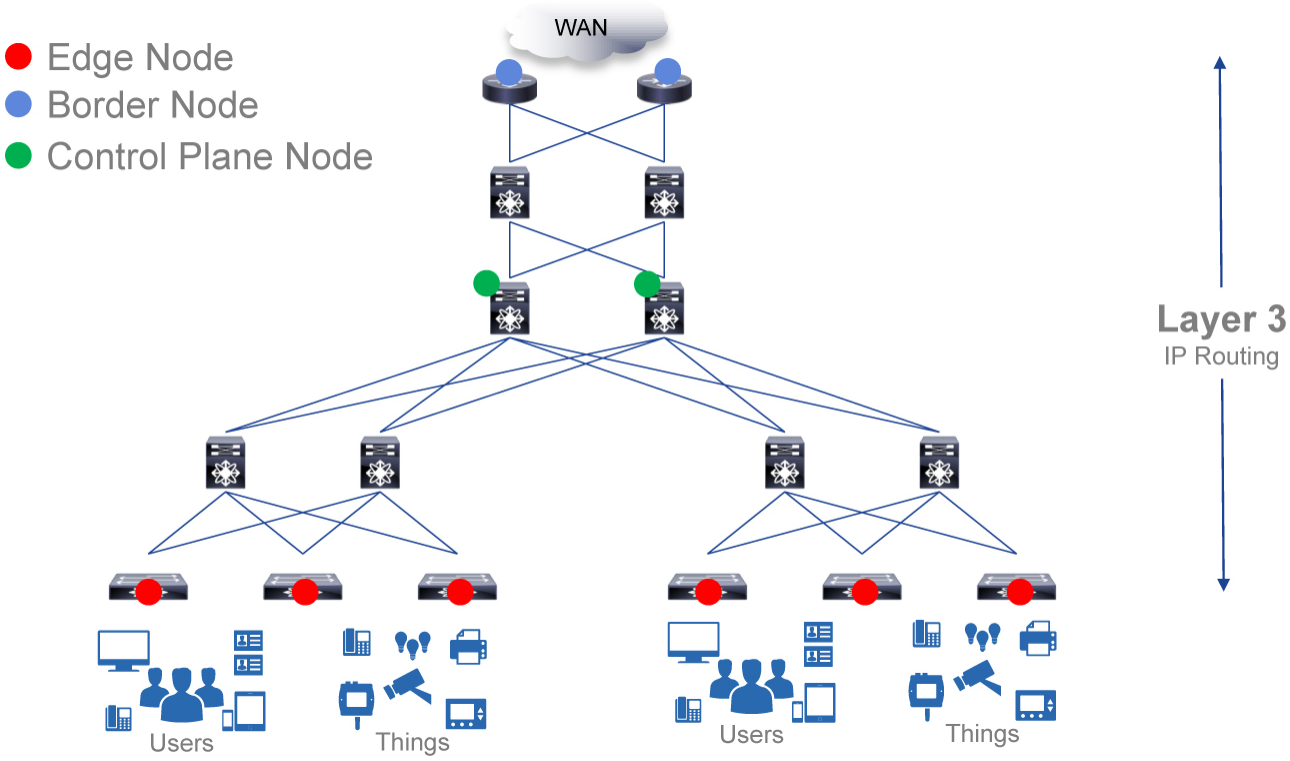


Figure 5-3 *LISP Overlay Access Network*

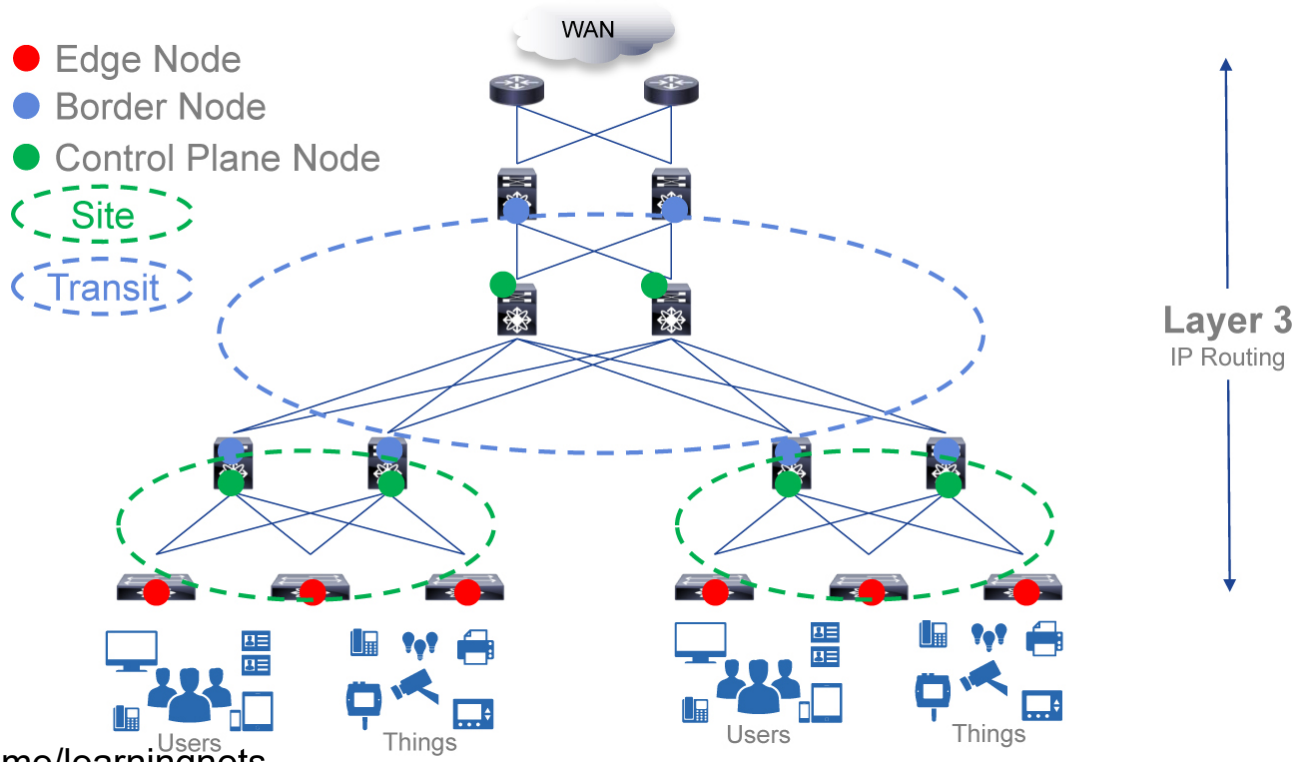
The design task is therefore reduced to choosing where to deploy the control plane and

https://t.me/learningnets

the points of connectivity to external networks. As for the connection of the endpoints, it is assumed that they connect directly to the edge nodes and that all access switches are therefore xTRs.

Control plane nodes should be deployed with resiliency and survivability in mind. One simple guideline is to deploy multiple Map-Server/Resolver nodes to provide resiliency in a LISP domain. Chapter 2, "LISP Architecture," discussed the mechanisms by which Map-Server/Resolver resiliency is achieved. Based on these resiliency mechanisms, multiple Map-Server/Resolver nodes are deployed within a domain to provide control plane resiliency. In cases where the required scalability of the Map-Server/Resolver nodes may exceed the capacity supported by the implementation, horizontal scaling is achieved by organizing multiple Map-Servers/Resolvers in a Delegated Database Tree (DDT). Chapter 2 and Chapter 4, "The Wide-Area Network: Bringing Traffic from Access to the Data Center," discussed how a DDT-structured Mapping Database System enables horizontal scaling of the control plane.

In many cases the risk profile mandates that the network be divided into multiple survivability domains to mitigate the possibility of a single failure affecting the entire network. Control plane resiliency within a domain is not sufficient in these cases, so multiple domains are created and connected together to provide the desired failure isolation. The domains are often referred to as *sites*. This model of federated sites was discussed in Chapter 4 and is illustrated in Figure 5-4. Each domain has its own set of Map-Servers/Resolvers and own set of borders, some of which are shared with the transit domain to provide the cross-domain connectivity required. The idea is that the domain would survive any failures that may occur in adjacent domains.



https://t.me/learningnets

Figure 5-4 *Multidomain Access Network Design*

All access switches connecting endpoints are configured as xTRs that register and service their directly connected hosts.

Once a domain hierarchy is decided, the network architect only needs to decide how to connect to external networks such as the Internet and the WAN.

Connecting to External Networks

An access network domain connects to one or more external networks. One obvious external network is the Internet, and normally, you do not want to import all prefixes from the Internet into the access network domain. For this reason, the Internet is usually reached when no explicit mappings are registered with the Mapping Database System for the destination requested. One way to look at this is that the Internet is full of unknown destinations, or destinations that are not registered with LISP, or non-LISP destinations. You connect to unknown destinations using a PxTR because the unknown destinations are non-LISP. You can deploy PxTRs at different sites, and you can configure each ingress tunnel router (ITR) to use a particular set of proxy egress tunnel routers (PETRs) to access the Internet and thus engineer the preferred path to the Internet in the network.

To attract traffic from the Internet toward the LISP network, you can use a couple of interworking options that are thoroughly documented in RFC6832. One is to advertise the endpoint identifier (EID) prefixes from the proxy ingress tunnel router (PITR) toward the Internet with a routing protocol. The other option is to use Network Address Translation (NAT) at the border router facing the Internet and translate the EID space to non-EID addresses that can already be reached in the Internet's non-EID routing. The latter model allows you to avoid having to advertise EIDs into routing protocols and keeps the LISP directory service independent from any interactions with routing. This model also keeps EIDs from being included in underlay routing.

Another type of external network is the WAN. Usually, the number of prefixes connecting to the WAN is relatively small, and it is therefore possible to learn prefixes from the WAN and register them with the LISP Mapping Database System at the access network site. These are seen as destinations that are known to the access network. Because these prefixes are registered with the LISP Mapping Database System, they are considered EIDs, and you connect to them using xTRs.

Unknown networks may be reached via two or more routers for redundancy. The multiple routers may even be connected to different provider networks, but, from the

https://t.me/learningnets

Technet24

perspective of the LISP overlay, consider the redundant array of routers and provider networks as a single set of locators for the reachability of non-LISP prefixes. In other words, the multiple paths to the non-LISP prefixes are consolidated onto a single mapping. The consolidation of all possible paths into a single mapping is equivalent to advertising a default route, in a traditional routing protocol, from multiple routers that are connected at the same place in the topology. Because their advertisements for the default route come from the same distance topologically, these routes have similar metrics and are considered equal cost routes and are all installed in parallel. In the case of LISP, the routers that connect to the non-LISP (or unknown) prefixes do not have to be in the same place in the topology, and the network operator can assign priorities and weights to the different routers to balance the load in a more prescriptive and topologically independent manner. Furthermore, the operator can configure policy at each ITR and specify which subset of the locators that connect to the unknown networks should be used by specific ITRs. Thus, the reachability of networks that the LISP Mapping Database System does not explicitly know is independent of the topology and can be subject to flexible policies.

Multiple different external “known” networks can have multiple points of connectivity. The multiple connections could provide parallel paths to common destinations. When the multiple egress points provide alternative paths to get to a particular destination, a priority order may be defined for the different points of egress to be used to reach a particular destination. This priority order is achieved using the priority attribute of the LISP mappings so that the routing locators (RLOCs) for the preferred point of egress have the best priority, and other points of egress are assigned priorities based on a policy or preference. Figure 5-5 illustrates the external connectivity considerations discussed. In this figure, the communication between A and B is preferred over the metro network with the WAN on SP1 being the backup path for the A-B communications, but the WAN on SP1 is the primary path for communications to prefix C on site 3 with a backup path over the Internet.

https://t.me/learningnets

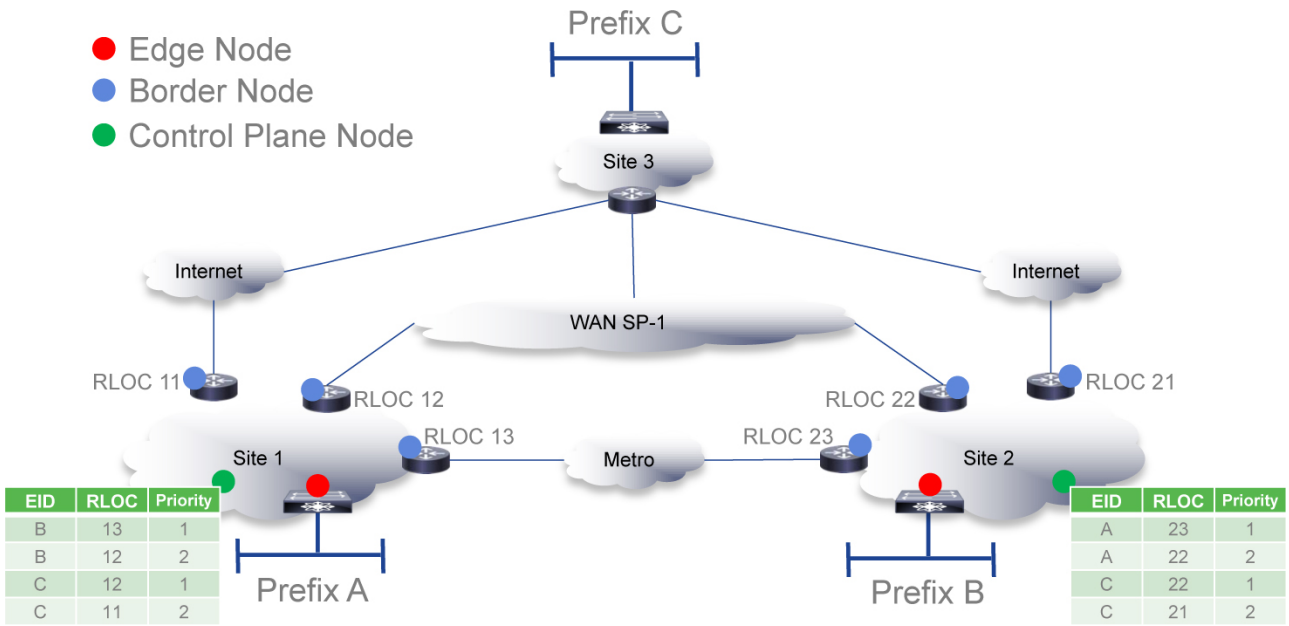


Figure 5-5 *External Connectivity for the LISP Access Network*

When connecting to external known networks, the xTR at the border may dynamically exchange routes with the adjacent network using a standard routing protocol such as Border Gateway Protocol (BGP). The destination prefixes from the routes learned from the external network must be registered with the LISP Mapping Database System as EIDs. Conversely, the xTR must advertise any EIDs registered in the Mapping Database System to the external network. This is a form of redistribution between LISP and routing protocols on the xTRs at the border, as illustrated in Figure 5-6. EID prefixes from the LISP overlay are advertised into the RLOC space, and prefixes in the RLOC space are registered into the EID space in LISP. When you are putting this type of redistribution in place, keep the RLOC space routing of the external network separate from the RLOC space routing of the LISP underlay network. In other words, advertising the EIDs to the external network shouldn't result in the EIDs being present in the underlay network that the LISP overlay is using. The main reason you want to avoid EIDs showing up in the underlay routing tables is that many implementations of LISP use the underlay routes at an ITR to forward traffic and bypass the LISP lookups altogether when a valid route is present in the underlay routing table. Separating the underlay routes from LISP routes is in place as a mechanism to support phased migration to LISP by enforcing a simple rule. If a prefix is in the underlay routing table, the ITR uses the existing routing, and if a prefix is not in the underlay routing table, LISP resolution and forwarding are in effect.

https://t.me/learningnets

Technet24

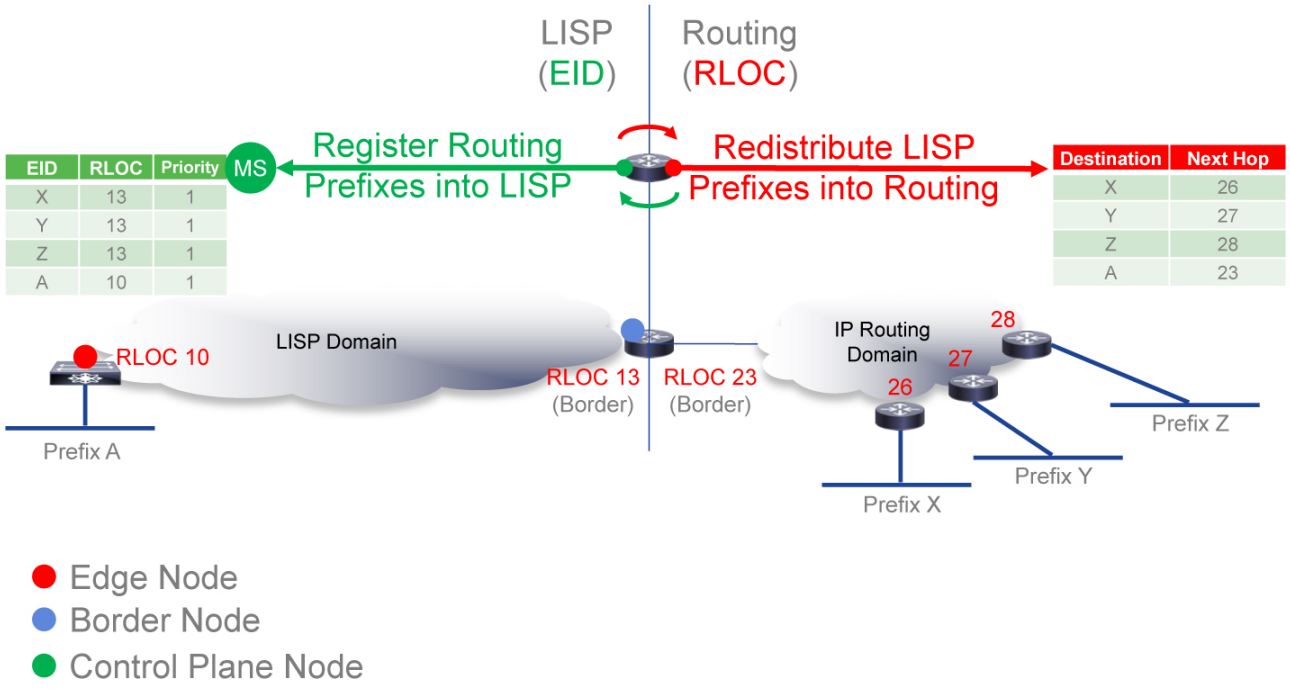


Figure 5-6 *Route to EID Redistribution at the LISP Domain Border*

One simple way of keeping the EIDs redistributed into the RLOC space from showing up in the underlay routing tables of ITRs is to keep the RLOC space that the xTRs use separate from the RLOC space beyond the borders of the LISP network. You achieve this by simply blocking routing advertisements at the topological boundary between the LISP domain and the IP routing domain illustrated in Figure 5-6. Well-understood routing techniques block the routing advertisements. For instance, using different instances of routing protocols on either side of the boundary and not implementing redistribution of routes between these instances of routing keep the two RLOC spaces separate from each other.

MOBILITY AND WIRELESS INTEGRATION

Wireless network access is instrumental in allowing the footprint of connected devices to grow rapidly. The simple notion of not requiring an expensive investment in cabling to bring a device onto the network is key to this growth. Additionally, wireless networks allow the connected device to be mobile and deliver an experience in which devices are connected to the network simply as part of being in the world.

Traditional IP networks are not particularly optimized for mobility because the entire scalability model in a traditional IP network is built around IP address summarization. To successfully group endpoints connected to the network into IP prefixes, IP addressing is structured in alignment with the network topology. For instance, all endpoints in location A use addresses within prefix A. Similarly, all endpoints in location B use addresses from IP prefix B. The network then can handle prefixes A and B without tracking the hundreds of devices that are represented by these two addresses.

https://t.me/learningnets

When endpoints are mobile, their location and IP addressing don't usually align with desirable summarization boundaries, so the network is challenged with maintaining much more granular reachability information. Historically, this task was handled by creating separate mobility networks that are, in effect, large Layer 2 networks and orthogonal to the IP routed infrastructure. The extension of Layer 2 over many locations is equivalent to allowing members of an IP subnet/prefix to exist at any location, with the caveat that these endpoints can access other networks through only one location. Funneling all traffic to a handful of handoff points between the mobile network and the Internet may not be a viable restriction moving forward, given the volume of connected devices and automation applications that keep proliferating. Furthermore, the use of L2 overlays such as those defined by Control and Provisioning of Wireless Access Points (CAPWAP) led to the loss of visibility of the network traffic and created an inefficient forwarding paradigm that steered all wireless traffic through relatively low throughput controllers. By using LISP, you can delegate the forwarding of traffic to the high-capacity switching infrastructure where visibility can also be regained.

Mobile networks traditionally were engineered differently than the IP networks that are used to build the Internet, which led to the networks existing separately and connecting at specific handoff points. Mobile networks are, for the most part, wireless networks. The enterprise access traditionally treated the wired and wireless networks separately for all the reasons discussed. However, with the introduction of LISP, scale and mobility are achieved seamlessly in the access network, effectively enabling the integration of wired and wireless networks into a single access network.

The LISP control plane supports very high scale given the fact that it is demand-based and only the states for active connections are cached at the ITRs. Given this connection-driven demand-based model, summarization of addresses becomes less relevant in scaling the system, and any IP address can move anywhere without posing an additional burden on the scale of the system. Nonmobile destinations may still be summarized, giving the LISP system the best scaling profile possible: cache only what you need and, if possible, summarize it. Because the LISP control plane is totally decoupled from the topology, summarization can still be achieved in the control plane of the LISP system while allowing members of a subnet to move freely to any location in the network topology. Members of a subnet can move to different locations in the topology yet continue to be registered to the same portion of the hierarchy in the control plane. Thus, the parsing of the control plane contents continues to benefit from swift lookups that leverage summarization and the structuring of the EID space; meanwhile, endpoints preserve their IP address and move anywhere. Figure 5-7 illustrates this behavior by showing members of the same subnet disseminated over the

https://t.me/learningnets
Technet24

topology, yet all are registered with the same node in the Mapping Database System. Other nodes in the Mapping Database System know to delegate lookups for anything in the mobile prefix to the authoritative node. The delegation information is summarized in the control plane structure without regard for how the addresses are mapped to the network topology.

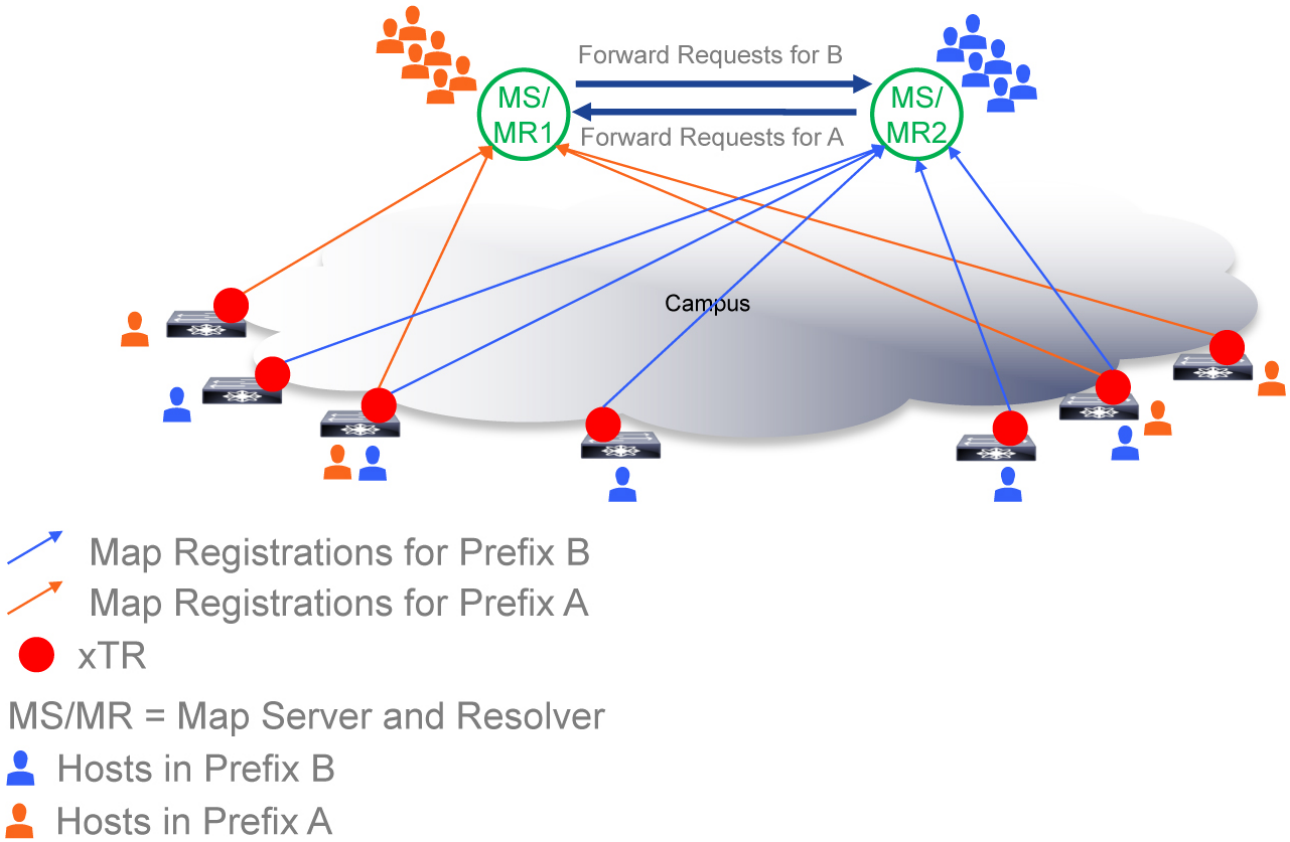


Figure 5-7 *Topology-Independent Addressing with Control Plane Summarization*

With this behavior in place, the network can support a large scale of hosts and host mobility without having to create a separate Layer 2 network or a nonintegrated overlay (for example, CAPWAP). The implication is that the wireless mobile network can now use the same control plane as the wired network and deliver all the mobility it needs to deliver. The benefits are many. One clear benefit is the normalization of wired and wireless operations as well as the available feature set for policy enforcement because now wireless access simply becomes a type of media to access the network, just like the different speeds of Ethernet are treated simply as variations of the access media. So, all traffic now hits an access port and is subject to an access and forwarding policy at these access ports regardless of whether the host is wired or wireless.

The integration of wired and wireless means that the control plane for wireless is replaced with a LISP control plane, and it also implies that the wireless data plane is handed off to the network at the access ports. Figure 5-8 illustrates what this network looks like. Note that the wireless controller's role is still to handle the aspects of radio

https://t.me/learningnets

frequency management relevant to the wireless access media.

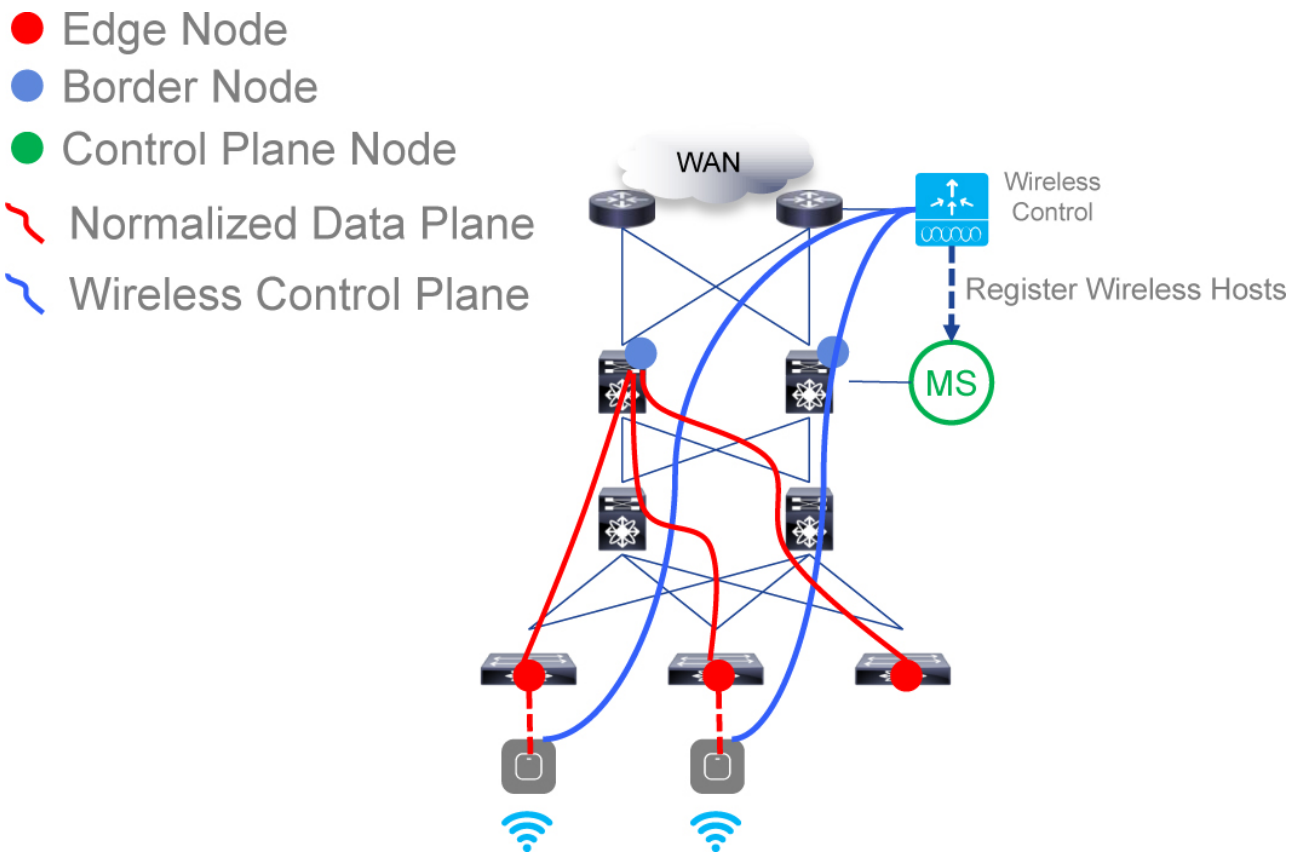


Figure 5-8 *Integrated Wired and Wireless*

One simple way of integrating the wired and wireless networks is illustrated in Figure 5-8. Mobility of wireless hosts is still signaled through the wireless controller; this configuration is ideal because the wireless controller can predict the direction in which a host is moving based on radio frequency telemetry. When the wireless controller is certain of a move event, it relays the location information to the LISP control plane. At this point, a wireless registration is indistinguishable from a wired registration, and traffic generated by both wired and wireless endpoints is treated exactly the same way by the network. This registration is done cleanly because the LISP Mapping Database System provides a logically centralized interface for the registration and updates of the location state of any endpoint. Thus, the wireless controller registers and updates wireless mobile host location directly with the LISP control plane, just like the access switches register and update wired host location to the LISP Mapping Database System. Figure 5-8 also illustrates how the data plane is modified to hand over all traffic from the wireless access points to the access switches and normalize the forwarding of wired and wireless traffic to use the same overlay in the access network. Traditionally, traffic from a wireless access point was tunneled to a wireless controller for forwarding. In the integrated model, traffic from the wireless access point is sent to the upstream access switch that it connects to, where it is then forwarded in the overlay network built with LISP among the access switches and the border routers. The native switching that the

https://t.me/learningnets

Technet24

LISP fabric provides is a much more efficient forwarding model than forwarding the traffic through a wireless controller.

Segmentation

Chapter 3, "Data Center Trends," discussed how to achieve segmentation in a LISP network by using extended EIDs (XEIDs), Instance-IDs, and virtual routing and forwarding instances (VRFs). As more and more applications rely on the network and as a larger number of devices become connected, the risk profile for the users relying on these applications and devices changes significantly. Having a larger number of connected devices means that the attack surface is much larger, and the exposure of the operation to malicious exploits is much higher. The use of segmentation in access networks is instrumental in reducing this attack surface and is also key to enabling the confluence of multiple networked applications.

Zero Configuration Networking: Service Discovery

Zero configuration networking is a set of technologies that create a functioning network without any manual intervention. To this effect, the following three functions are necessary:

• Automatic assignment of network addresses for connected devices

• Automatic distribution and resolution of hostnames

• Automatic location of network services

The automatic assignment of network addresses is built into the IPv4 and IPv6 stacks where link-local addresses are autoconfigured on hosts. You achieve the resolution of hostnames and service location functions by using Microsoft NetBIOS or Domain Name System (DNS) Service Discovery (DNS-SD) based on multicast DNS (mDNS). One common characteristic across these implementations is that the messaging is exchanged over a link-local channel over which discovery and service advertisement are multicast without requiring any configuration on the hosts. The practical implication of this approach is that the scope of these zero configuration services is the Layer 2 network on which they are active. In other words, zero configuration networking was traditionally achievable on an L2 network, and its scope is limited to the reach of such a Layer 2 network.

In a LISP-enabled access network, one simple way of supporting zero configuration networking is to enable Layer 2 virtual networks. These Layer 2 services were discussed

https://t.me/learningnets

in Chapter 3; they are simply achieved by using Media Access Control (MAC) addresses as EIDs and extending the EID space to use an Instance-ID to create extended EIDs (XEIDs) in the MAC address family. Doing so allows the creation of multiple Layer 2 virtual networks. The segmentation into different L2 VPNs is preserved at the xTRs by caching the XEIDs in bridge domains (aka VLANs) on the xTRs. The Layer 2 virtual networks also support link-local multicast and broadcast services by replicating traffic destined to the broadcast and link-local multicast addresses to all xTRs where the Layer 2 virtual network is instantiated. This type of service allows the mDNS and NetBIOS mechanisms to work the way they work today in a traditional network. But this also forces the network design of the overlay to stretch Layer 2 domains and align their scope with the desired scope of the zero configuration network.

The LISP infrastructure also provides support for the zero configuration networking environment without instantiating Layer 2 virtual networks. This is key in preserving the mobility and scale benefits discussed so far. A LISP implementation allows the creation of broadcast and link-local multicast distribution trees independent from the creation of Layer 2 virtual networks that would hold all unicast reachability state for Layer 2. This is simply a stateless flood service to enable discovery and name resolution mechanisms. The advantage of this approach is that the scope of discovery services is configured independent of the design of the overlay virtual networks and matched to criteria beyond the network design. For example, the policy for service discovery may be governed by the floorplan of the building.

Situational Policy (Beyond Just Location)

The information about EID to RLOC mappings stored in the LISP Mapping Database System can be easily extended to include context information related to the RLOCs as well as the EIDs. LISP enables you to encode more information for EIDs and RLOCs as well as support different types of EIDs and RLOCs through use of the LISP Canonical Address Format (LCAF). Because LISP operates as a directory service, it can be consulted based on different combinations of the extended semantics of the RLOCs and EIDs. The queries and responses to the Mapping Database System are subject to policies anchored on the extended semantics. For example, EIDs may be registered with semantics that indicate their geographic coordinates and also the type of device that hosts these EIDs; for example, devices may be wired, wireless, trusted/known by IT, or unknown/not trusted. You may define a policy to provide different mappings for different types of devices. Furthermore, the policy may also differentiate between devices that connect within a specific geographic region. In this way, the location and type of the device determines whether the device is allowed to connect to certain destinations and whether the traffic should be steered in any particular way, such as via

https://t.me/learningnets
Technet24

a specific path and potentially through inspection services such as firewalls or intrusion prevention systems.

Applications

Let's look at some common applications in which LISP-enabled access networks are being leveraged to solve relevant problems.

Optimized Campus and Branch Access

One leading application of a LISP-enabled access network is corporate access networks. Corporate networks are deployed in a variety of environments and combine a diverse set of wired and wireless endpoints. The current practice in corporate IT is to create separate/dedicated networks for different applications such as wired, wireless, building automation, surveillance, and so on. Furthermore, the design of the network and its deployment may vary from one location to another. A LISP-enabled access network consolidates all these disparate networks into one network optimized for mobility, segmentation, and scale. You can segment the consolidated network to continue to offer logically independent services for the different types of devices being connected.

The abstraction provided by the LISP overlay is instrumental in normalizing the design and deployment procedures across a diverse set of locations. Regardless of the location characteristics, you can apply the same design and the same deployment considerations to the overlay at any location. The ability to streamline the deployment of networking at a location is key to accelerating service deployment in operations that require swift enablement of new locations. Industries such as banking, health care, and retail are clear examples of verticals that can quickly benefit from the normalization of the network design across locations.

In many cases a physical location is very small, so IT departments deploy a subset of services to these locations, which are generally referred to as *branches*. The current tendency is to customize the design of the branch to keep the costs of small branches in check. The LISP overlay offers the ability to avoid these exceptions and allow the branch to be designed just like any other location would be designed. Because LISP provides a wide variety of services (Layer 2, Layer 3, cryptography, policy, mobility, etc.) using a common set of architectural principles, the functionality delivered at a given site can be scaled up or scaled down as necessary without a change in the design of the network. This allows the design of small branches to align with the design of large sites while enabling only the relevant network services to maintain the low cost of the branch.

https://t.me/learningnets

Connected Home

Home networks evolved over time to host a large number of connected devices. Initially, personal computers were connected to access the Internet; then connected consumer electronics proliferated in the home. The home network providing connectivity to these devices is a private network with private address space that cannot be reached from outside the home. When the devices connect to services on the Internet, the connections are initiated from the home to the outside and must go through a Network Address Translation boundary. The NAT service provides a public address for the device connecting to the Internet so that it can be reached over the public Internet. This was the operating model for home networks for many years, and it worked relatively well until an increased offering of connected home devices and the low entry point into cloud-based data centers started enabling home automation applications that required the private home network to open up.

Consequently, home networking continued to evolve to connect lightbulbs, surveillance cameras, window blinds, thermostats, water heaters, refrigerators, and anything that could be controlled or communicated with. Today the home looks very much like a small branch of a corporate network from the perspective of the mix of applications and devices running on it. A modern home has lightbulbs that connect to the network, surveillance cameras and smart doorbells that are connected to the network, and solar cells that connect to the network to be controlled. However, the modern home does not have a server room to host the control systems for any of these services. The modern home relies on the cloud to host the applications that provide the surveillance services necessary for its connected cameras to be useful. The homeowner also relies on the cloud to host the applications that allow lights, thermostats, and solar cells to be manipulated to optimize energy consumption in the home.

Chapter 3 discussed how the access network should ideally stretch to the edge of the data center to provide secure connectivity between applications. The same considerations apply to the modern home network where connectivity should ideally be extended to the cloud locations where the applications bringing value to the connected devices in the home reside. In this environment, connections are initiated from the home and also from the cloud-hosted applications to the home. Many applications address this connectivity by using a hub at home and establishing an application-level secure connection between the home hub and the data center. One of the premises is to offload from the endpoints and applications any network-related tasks; thus, a network service capable of providing secure connectivity between the home and the cloud-hosted applications is of interest. The LISP overlay provides all the necessary functionality to allow the connected home to connect to its cloud-hosted applications

https://t.me/learningnets

Technet24

securely. Virtual private networks (VPNs) are easily delivered in the LISP overlay to extend the footprint of the home network to include locations in cloud data centers (CDCs). This logical extension of the home private network allows the home to include the necessary data center resources without compromising on security or simplicity.

Campus Dormitory Rooms: A Virtual Home

Accessibility to technology and connectivity has changed significantly since the turn of the 21st century. Today, young adults have tens of connected devices. In addition to their personal mobile devices, they own video streaming devices, game consoles, printers, and so on. They attach these devices to their home network simply by specifying the name of their network. This network name is the SSID of their home Wi-Fi network.

In many cases these young adults are of college age. When school starts, students flock into the dormitories and bring with them their collection of gadgets in need of connectivity. They expect the experience to be the same as at home, where the Service Set Identifier (SSID) for their network was sufficient to get their gadgets connected. But it isn't practical to provide each student with a dedicated SSID. The wireless system simply cannot scale to support the large number of SSIDs. This scale limitation is due to the spectrum allocation and frequency management overhead that is associated with each SSID.

Rather than trying to solve this problem in the radio space, we can easily address this problem with routing policy in the LISP control plane. A student may be assigned an IPv6 prefix and be required to register devices with the university's IT department. This is one way to ensure that every device the student brings in is assigned an IPv6 address within the student's assigned prefix. Based on the IP prefix, the devices may be registered in a dedicated forwarding context, such as a VRF, that re-creates for the student the experience of the dedicated home network environment on the shared university network. Interestingly, because the LISP network supports any prefix in any location, at no additional cost, the student can move from one dormitory facility to another without any additional intervention from the university's IT department. In such an environment, the student could also grab a video streaming device and bring it to a classroom or auditorium to support a presentation. The footprint of the virtual home network is not limited to the student's room, but it can include the entire campus if the university's IT policy permits it.

LISP-Based Air-to-Ground Network

One interesting application of the large-scale mobility capabilities of LISP is that in

https://t.me/learningnets

which aircraft are assigned a fixed IP address so that any organization that may need to communicate or track the aircraft can uniquely identify it. This capability is instrumental in enabling communication with air traffic control, aircraft manufacturers, and others.

Planes connect to the ground network via a variety of radio networks that operate heterogeneously. The different radio networks aggregate at a ground backbone network that is designed to steer traffic to and from the radio networks based on policy signaled by the aircraft. Thus, aircraft are connected to one or more of these radio networks at any point in time, and from the perspective of the ground backbone network, the aircraft are moving from one attachment point to another with varying priorities and weights. Chapter 4 discussed how LISP delivers simple traffic engineering policies based on a simple hierarchy of priorities and weights. This mechanism is leveraged by the ground backbone network to guarantee the flow of traffic from ground to air over the radio links preferred by the aircraft policy. Figure 5-9 illustrates the LISP-based ground network. Note that the LISP network starts at the air-ground routers, which are LISP xTRs. Planes attach to different radio networks as they travel, and this eventually is reflected as the plane changing the air-ground router that registers it. In other words, a LISP mobility event takes place.

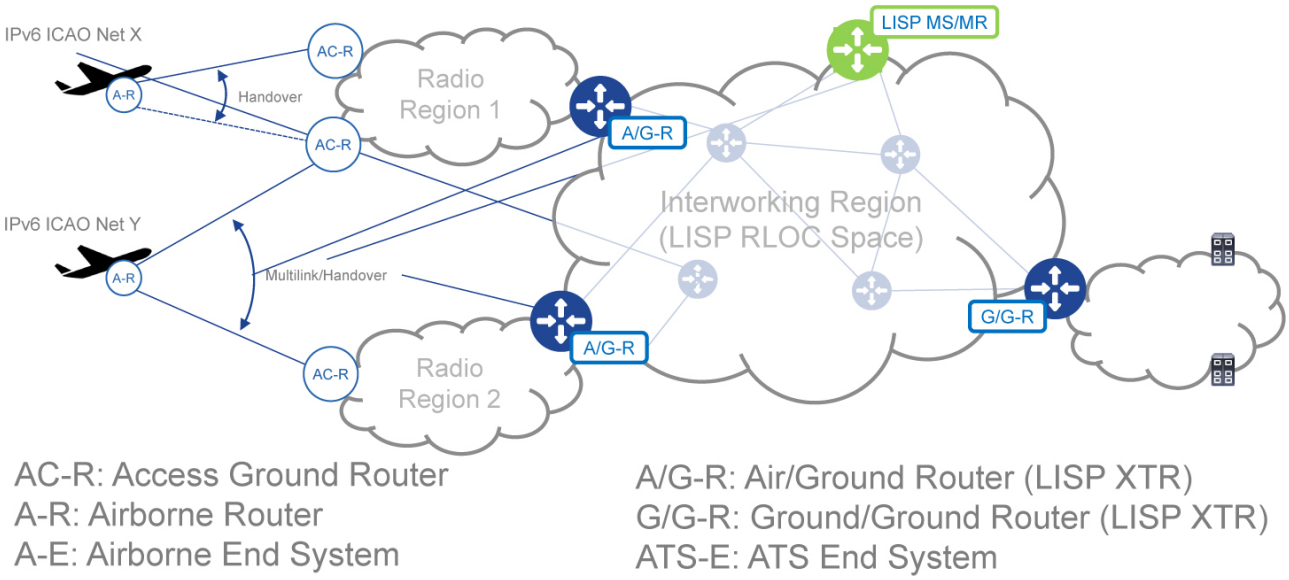


Figure 5-9 *LISP-Based Ground Network*

This network is meant to be a global backbone, and it is also understood to have different traffic profiles within a region and across regions. Just like the world is organized in geographical regions, the LISP ground backbone is organized in a similar way. You can obtain interesting optimizations from this structure. A vast majority of aircraft may not travel outside their home region. For example, a region may be a country, and many aircraft may never be used internationally, whereas a few aircraft are dedicated to international routes. Because the goal is to assign a prefix to each

https://t.me/learningnets
Technet24

aircraft so it can be uniquely identified, some prefixes will roam exclusively within a region and some prefixes will roam both within a region and across regions. Chapter 4 discussed the structuring of a large-scale LISP network into a federation of sites. This federation of sites is used to structure the air-to-ground LISP network into the corresponding regions. We already discussed the scale benefits of such a structure. In the air-to-ground network application, the cross-regional state is minimal, and the scale benefits of the LISP demand-based approach are realized to their full potential.

When each aircraft is assigned a large enough prefix, such as those provided by IPv6, the more specific addresses within the prefixes are used to identify types of traffic or even to signal policy from the aircraft down to the ground network. Thus, certain bits in the addressing can allow the LISP air/ground routers to determine the priorities and weights with which a certain prefix should be registered. Different types of data are serviced; some is air traffic control data, whereas other data allows aircraft manufacturers to access the plane. Air traffic control data connections typically occur within a region (aka site); actually, they occur between the aircraft and the closest air traffic control tower, so it is localized traffic. Meanwhile, aircraft manufacturer connections may be constantly crossing regions because U.S. manufacturers sell aircraft for use in Europe just as much as European manufacturers are successful at positioning their aircraft in the U.S. market. Neither type of connection poses an additional burden on the state required in the network. The state created in LISP is summarized across boundaries while still allowing the desired mobility of the aircraft.

Endpoint Tracking Applications: Geo-location

Tracking items is a costly endeavor in many industries. In refineries, mines, and manufacturing plants, tracking the location of tools is desirable to reduce cost and improve safety and productivity. In large retail operations, tracking the location of merchandise in transit or in warehouses is an ongoing pursuit, while tracking the location of potential consumers of these goods is an emerging trend. For the transportation industry, the financial implications of improving the accuracy of tracking the location of parcels and luggage are significant.

As items are connected to the network, the network is in a unique position to assist in the determination of the physical location of objects. In LISP, the topological location of an EID is expressed as a set of RLOCs, and as RLOCs are correlated to their place in the physical network topology, they can provide some sense of the physical location of the devices.

The LISP system is highly extensible in the semantics that can be associated to both EIDs and RLOCs. The standard mechanism to extend the semantics of ID and locator

https://t.me/learningnets

in the LISP control plane is provided by the LISP Canonical Address Format defined in RFC8060. This document specifies the address format to include geo-coordinates as part of the semantics of an RLOC. When the definition of an RLOC is expanded to include geo-coordinates, the set of EID-to-RLOC mappings registered with the Mapping Database System effectively becomes a directory of physical location information for the registered EIDs. Mobile EIDs attach to different xTRs as they move, and the geo-coordinates with which they are registered are updated in the LISP Mapping Database System as the EIDs move, thus providing an up-to-date directory of physical location. This directory can be queried to obtain the geo-coordinates of the xTR that an EID is registered with. For instance, a shipping company may leverage this functionality and assign an IPv6 EID to each parcel it handles. The network has LISP nodes deployed in many locations, and the nodes are either equipped with a GPS or have their geo-coordinates configured. As the parcels connect to the different LISP nodes, they are registered with different RLOCs in the Mapping Database System. The RLOCs include the geo-coordinates of the LISP node where the parcel is connected. The network now provides additional information for the parcels without increasing the cost of the electronics that have to be attached to the parcel. The applications are many. One simple application is to be able to track parcels with relatively good accuracy without requiring the parcel to be scanned by different operators as it travels across the courier's network, but simply by querying the LISP Mapping Database System. In the mindset of the Internet of Things, you may think of a parcel that is equipped with a GPS and communication equipment to report its location. The cost of packaging such a parcel would likely be prohibitively high. Lower cost technologies such as radio frequency identification (RFID) or Bluetooth Low Energy (BLE) beacons can be leveraged to implement a cost-effective interface between the parcels and the network. The key is that the parcels have little intelligence, and the network is supplementing these low-intelligence devices. Physical location is a clear example of the role of the network in augmenting the contextual information of things that connect to it.

Beyond the obvious application of determining location for an EID by extending the semantics of the RLOCs to include geo-coordinates, the extensibility available in LISP allows you to also extend the semantics of an EID. You can extend an EID to represent a geographical area or perimeter. This area is described in draft-farinacci-lisp-geo as a *geo-prefix*. A geo-prefix is simply a tuple of a geographical point (geo-point) and a radius around that point. The LISP LCAF provides a variety of possible formats to encode such a tuple. When EIDs are structured as geo-prefixes, requestors can query the LISP Mapping Database System to find further information about a particular geo-point. A geo-point falls within a geo-prefix, just like an IP host address falls within an IP prefix. In the LISP Mapping Database System, a locator set inclusive of information

https://t.me/learningnets

extensible beyond the IP address of the RLOCs is associated to a geo-prefix and is provided in response to any queries for geo-points that fall within the geo-prefix.

You are probably thinking that this doesn't have much to do with routing or reachability, and that is precisely the point. LISP should be seen as a directory service or an addressing architecture, not a routing protocol. As such, the LISP directory can be extended to allow the network to provide valuable and necessary information related to how and where endpoints attach to it.

THE INTERNET OF THINGS

For centuries, humanity pursued the classical Greek ideals of leisure. The Greek citizen was afforded the privilege of leisure by offloading busy and time-consuming work to slaves. The leisure of the Greek citizen enabled contemplation, education, and deep thinking. All were good things, with the exception of slavery and the challenge of choosing fairly who should do the work and who should be allowed to contemplate.

If "things" can be given a small degree of intelligence and they can be automated, maybe this network of things can take the role of the Greek slaves and effectively deliver a degree of citizenship to all of us. Instead of having a slave permanently adjusting the room temperature at the thermostat or making sure lights are turned off when no one is in the room, networked things and sensors can complete the chores of the slave. For a shipping operation, instead of having an employee mechanically scan packages as they are transported across the distribution network to determine their location and ensure their correct routing, sensors and robots can automate this task and allow the employee to shift focus to more interesting and beneficial tasks. Examples exist in all industries; for instance, manufacturing tasks are being offloaded to robots more and more. Traffic control tasks can also be automated by networking a series of sensors and automating traffic lights. The key is that these mechanical slaves must be instructed what to do and when so that the desired outcome is achieved. The orchestration of these things requires that they be connected to each other and to their control centers. This is the Internet of Things, and for it to be successful, the network must evolve and adapt to a world where connectivity is pervasive.

Security and Integrity

One important aspect of the Internet of Things is confidentiality. In many applications, it is necessary to encrypt the connections among things and between things and applications. Encryption exacerbates the scale requirements on the network significantly because it requires the maintenance of security associations in addition to

https://t.me/learningnets

the connectivity between a large number of locations where things are connecting.

LISP demand mechanisms come into play again in this area. LISP enables the exchange of public key information at the time of mapping resolution. Because the mapping resolution involves both ends of the connection (the ingress tunnel router and the egress tunnel router), public key information is exchanged between ITRs and ETRs to enable the ITRs and ETRs to calculate private keys to encrypt their communications. Rather than precalculating and pre-establishing all possible security associations required, only the required cryptographic connections are established when they are requested.

The ability to integrate the exchange of information for cryptographic key calculation into the standard LISP protocol messaging has fundamental security and scale implications. The pairwise keys calculated with this method are much more secure than shared or group keys. Also, from a scalability perspective, only required keys are calculated, and only required security association state is instantiated in the forwarding path.

The LISP control plane provides the necessary mechanisms to guarantee integrity of the EID information exchanged. Signature heuristics to handle the authentication and authorization of mapping information are built into the LISP control plane.

Chapter 6, "Security," discusses in detail the mechanisms by which LISP enables cryptography as well as the complete set of security functions integrated in LISP to provide integrity and confidentiality of data traffic and control plane information.

Sensors: Mega-Scale Aggregation of Very Little Data

Sensors are the canonical example of a thing that benefits from being connected. Sensors are deployed to cover different scopes: a home, a manufacturing plant, a country, or an entire continent. Thus, sensors are widely distributed to cover a wide footprint with granularity and precision. These applications include

- Building automation
- Meteorology
- Seismology
- Energy generation
- Health care

https://t.me/learningnets

• Traffic management

• Manufacturing process optimization

• Performance monitoring

• Tuning of vehicles, such as aircraft or race cars

In many of the applications, a large number of sensors are distributed, and the data generated by the sensors is centrally aggregated and correlated. Each sensor transmits a small amount of information, and sometimes the information is transmitted with a relatively low frequency or even in random bursts. The net result is a low volume of data from each sensor. Yet, the number of sensors in these networks is extremely large, in the order of hundreds of thousands in some applications. The sensors can be distributed enough that each sensor connects to its own xTR. This means hundreds of thousands of xTRs in a single network. With a traffic pattern that is aggregated to a central location where the data from all the sensors is collected, the aggregation xTRs service a large distribution of connections. Figure 5-10 illustrates this high density of sensor locations and how their connections aggregate at a central location.

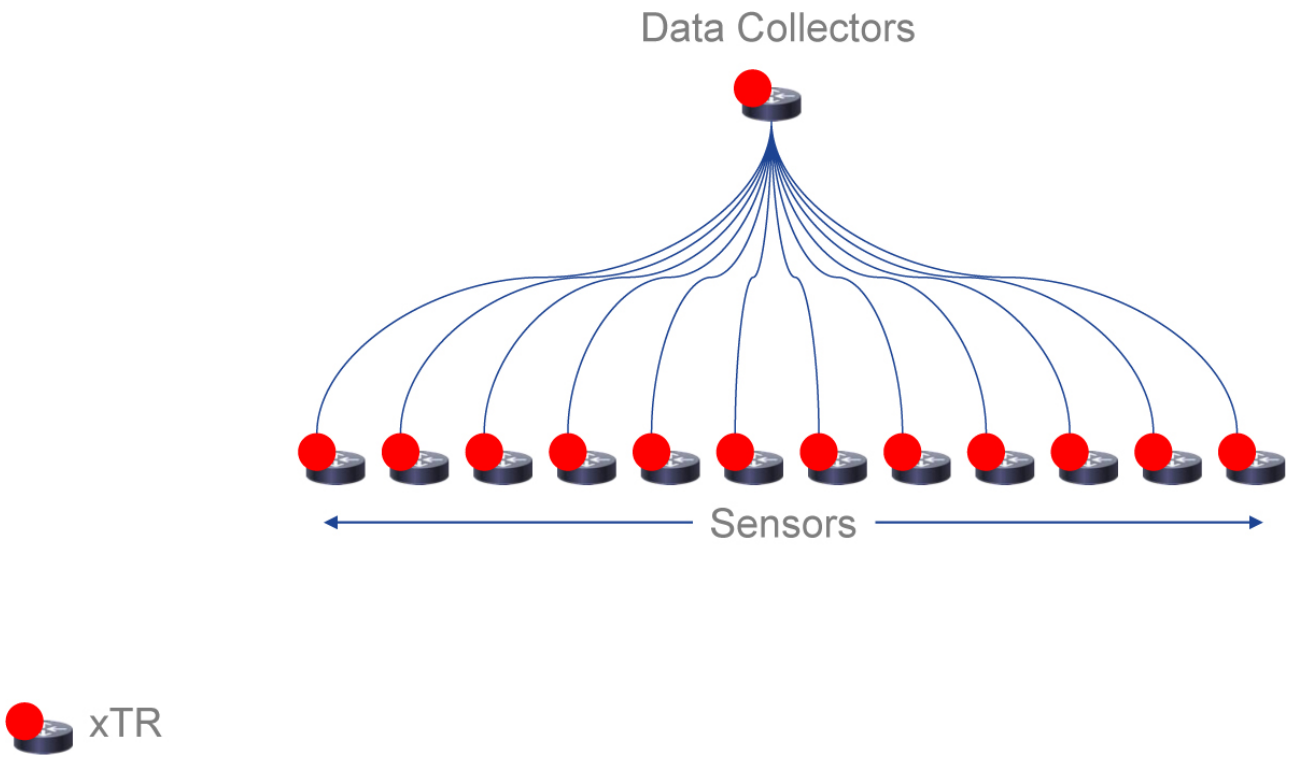


Figure 5-10 *Aggregation of Sensor Data*

LISP horizontally scales the aggregation points by simply using multiple xTRs to service the aggregation site, as illustrated in Figure 5-11. The traffic engineering parameters of priority and weight come in handy to ensure that connections are distributed evenly across the multiple xTRs at the aggregation site. For instance, the xTRs can register the

https://t.me/learningnets

EIDs for the aggregation servers using a common priority and balanced weight for all RLOCs available on the xTRs at the aggregation site. The result is different connections using different xTRs to bring traffic from the sensors to the data aggregation site.

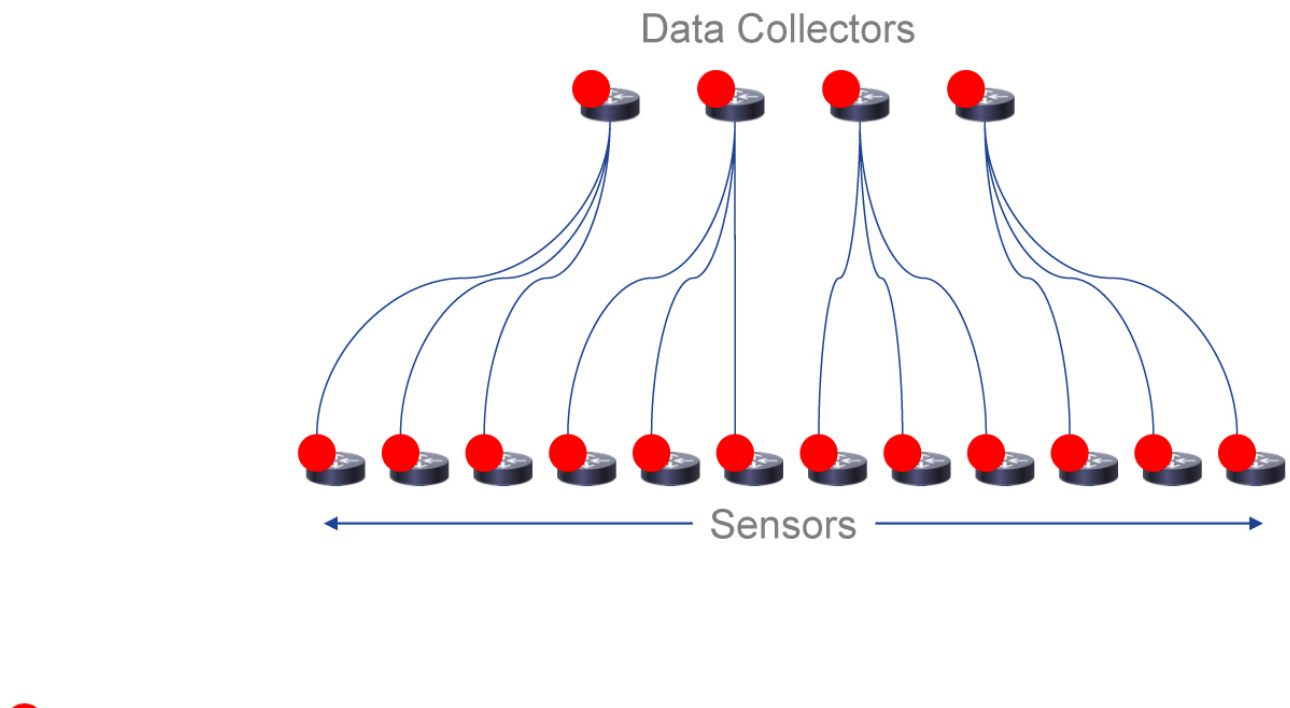


Figure 5-11 *Horizontal Scaling of xTRs at Data Collection Site*

In many cases, the flow of information is unidirectional and limited to the sensor-to-collector direction. In this case, the ingress traffic engineering settings at the aggregation site are sufficient to balance the load at the aggregation site. You may have noted that if traffic is expected to flow only in the sensor-to-collector direction, the number of cached mappings is small. Basically, a handful of mappings are cached for the collector EIDs at the sensor site ITRs, and because traffic isn't returning to the sensors, there shouldn't be any mappings on the xTRs at the collector site. So why bother balancing the load? We already stated that the volume of traffic produced by the sensors is low, so even when you look at the total aggregate data from hundreds of thousands of sensors, simple mathematics indicates that one or two routers could handle the load. So why scale horizontally? The main reason is that the connections for this large distribution of sensors must be encrypted. A second driver is that the communication will be bidirectional in many cases, burdening the collector-side ITRs with a large number of Map-Cache entries. Both of these conditions result in an increase in the amount of state that must be held at the xTRs at the aggregation site. Let's analyze how LISP addresses the scale requirements in these sensor networks.

When cryptography is used in LISP, security associations are created as connections are established. Based on a tuple that includes the connection header source and

https://t.me/learningnets
Technet24

destination parameters as well as the LISP priorities and weights for the destination EID, the Map-Resolution involves a particular ETR at the aggregation site and creates some security associations with some ETRs and other security associations with other ETRs, as illustrated in Figure 5-12. This process of establishment of encrypted connections is discussed in more detail in Chapter 6. Without going into the details here, Figure 5-12 illustrates how the security association state is horizontally distributed across multiple xTRs.

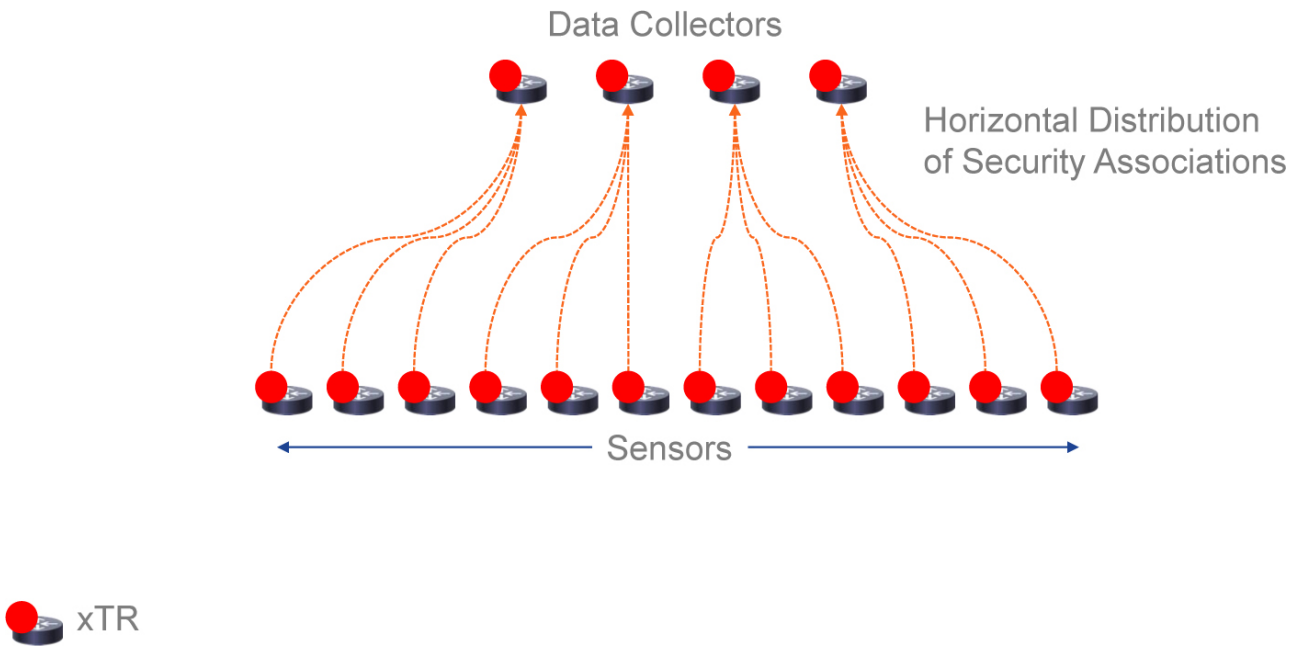


Figure 5-12 *Horizontal Distribution of Security Associations*

When traffic is bidirectional, security associations are established in the return direction from collector to sensor. LISP does not have a way to prevent connections from taking a return path that is different from the ingress path. In other words, a sensor-to-collector connection can come into one xTR, yet the return traffic from the collector to the sensor can hit a different xTR at the collector site, as shown in Figure 5-13. As illustrated, cryptography state is created randomly across the different xTRs in different directions. This asymmetry is outside the control of the LISP portion of the network. However, as long as flows are balanced evenly in the routed network between the collector and the collector-site xTRs, horizontal scaling of the security associations is achieved in both directions. The balancing of flows in the routed network can be engineered or simply the product of equal-cost load balancing in a symmetric network.

https://t.me/learningnets

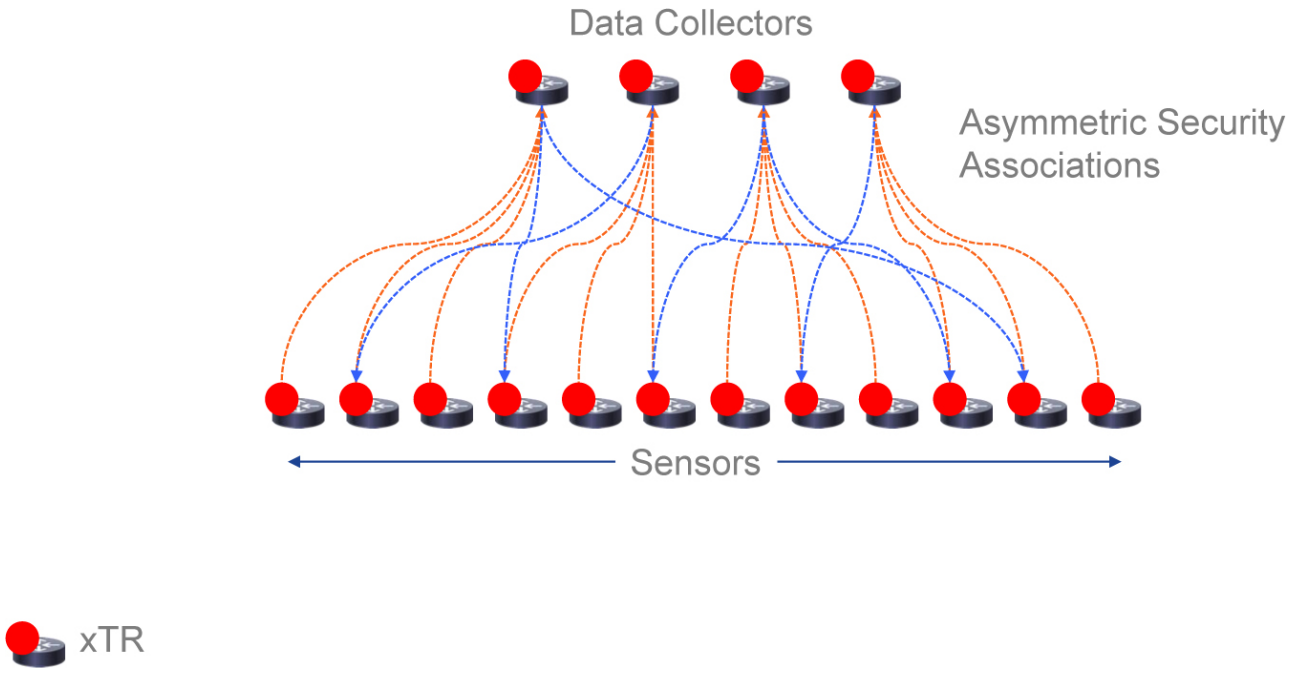


Figure 5-13 *Collector-to-Sensor Horizontal Scaling of Security Associations*

As discussed in Chapter 6, the security associations established with LISP cryptography are unidirectional, so the asymmetry observed does not actually have a fundamental impact on the effectiveness of the horizontal scaling design discussed.

An alternative design to aid in the scaling of the aggregation points can be based on the multisite hierarchy discussed in Chapter 4 in the context of interconnecting the access network and the WAN as adjacent LISP networks. In the multisite hierarchy, the sensor network can be organized into a multitude of smaller access domains that connect to an aggregation domain, as depicted in Figure 5-14. The number of connections and security associations to be handled within each domain is reduced significantly from what is required in the single domain model. In the hierarchical model, the encapsulation, as well as the encryption, is terminated and reinitiated at the re-encapsulating tunnel routers (RTRs) at the boundary between the domains. Most state will be concentrated in the RTRs, as shown in Figure 5-14, where you can appreciate how the amount of state required from the different LISP nodes is significantly reduced.

https://t.me/learningnets

Technet24

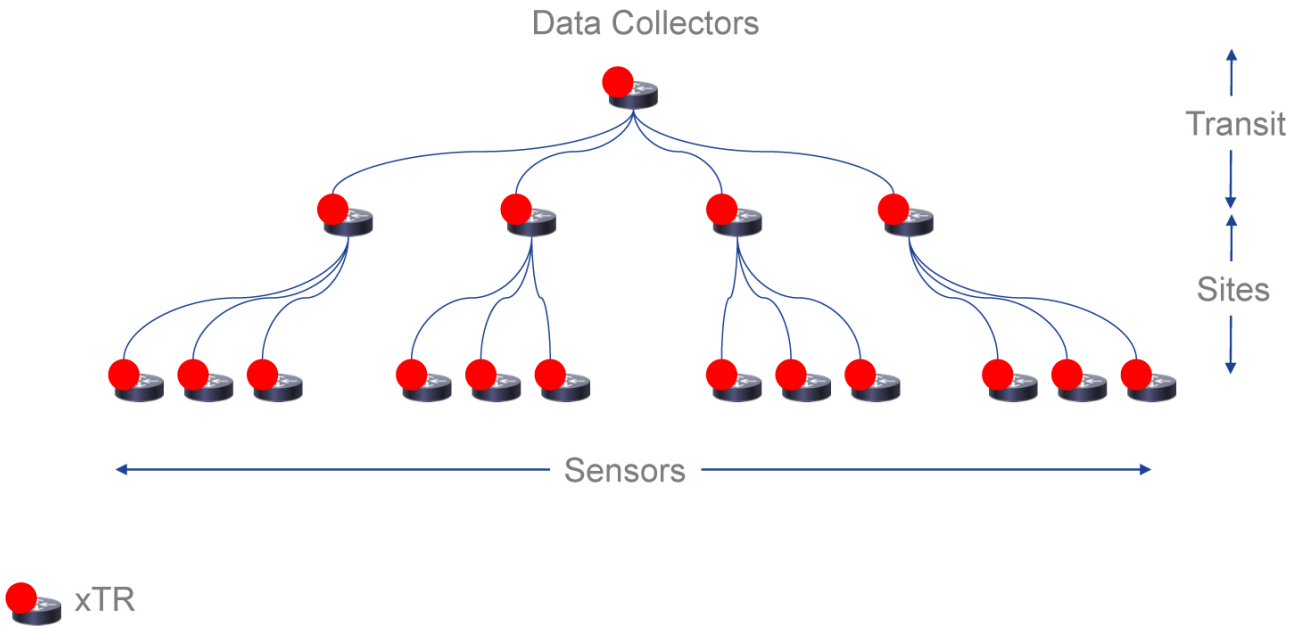


Figure 5-14 *LISP Hierarchical Sensor Network Design*

A Protocol Fitted for Low-Power, Light-Footprint Applications

This chapter has focused on the applications where the things connecting to the network do not have a lot of processing power and network processing is delegated to the network device the things connect to. However, for many applications, you might want to have the LISP xTR functionality on the endpoint device. A LISP xTR has a small CPU and memory requirement and is particularly well suited to be deployed on the endpoint device.

Benchmark analysis on delivering an L3 VPN service found that the LISP central processing unit and memory requirements are roughly between 5 and 10 percent of what BGP requires to provide an equivalent service at the same scale. This is in addition to the fact that LISP keeps small Map-Caches versus the fully populated routing tables that BGP produces. This delta in processing requirements is to be expected because LISP is focused on a different problem than BGP and therefore runs different types of processes for which it holds state in a different manner. Again, each protocol has its place and mission. For the purposes of the connectivity and policies required by a low-cost mobile device, LISP provides the optimal combination of scale, mobility, segmentation, and context extensibility.

It is plausible to envision applications in which a LISP stack is included on small processors to deliver customized connectivity for devices in a way that is totally autonomous from the network they connect to. Devices connected to a cellular data network, for example, benefit from having autonomy in their ability to utilize a LISP directory to join ad hoc VPNs or participate in geo-location-centered applications. The medium could also be a private set of radio links, Wi-Fi, Bluetooth, or even Near Field

https://t.me/learningnets

Communications (NFC) connections among the devices, or anything enabling the formation of a low-cost ad hoc network among devices. In the field, geo-location, policy, and privacy are key attributes of networking that are delivered easily with a combination of low-cost processing capacity and the LISP directory with the xTR functionality built into the devices. For instance, a series of drones can form an ad hoc virtual network in which geo-coordinates are registered with the LISP Mapping Database System to enable a series of applications ranging from simple geo-fencing to geographically precise augmented reality.

A Lightbulb for Utopia

When Sir Francis Bacon wrote *The New Atlantis* in the 17th century, he drew a picture of a technological utopia that was rather comprehensive. He wrote about a series of houses where different aspects of science were explored and where innovation took place. In particular, he wrote about "perspective houses":

We have also perspective-houses, where we make demonstrations of all lights and radiations; and of all colours: and out of things uncoloured and transparent, we can represent unto you all several colours; not in rain-bows, (as it is in gems, and prisms,) but of themselves single. We represent also all multiplications of light, which we carry to great distance, and make so sharp as to discern small points and lines. Also all colourations of light; all delusions and deceits of the sight, in figures, magnitudes, motions, colours all demonstrations of shadows. We find also divers means, yet unknown to you, of producing of light originally from divers bodies.

The accuracy with which Bacon predicted the current scientific landscape some 400 years ago was uncanny. Our current houses aren't precisely the light houses from *The New Atlantis*, but they do have a multitude of lightbulbs and outlets that are automated to help contribute to an ecological and financial utopia by saving energy and therefore reducing the cost of running homes.

So, what is it that makes a lightbulb fit for these purposes? Aside from the ability to produce light of arbitrary purity, it is the ability to be controlled and of the lowest possible cost that truly makes these automated things relevant. If these lightbulbs and electrical outlets are to be connected to an IP network, the network could offload a series of functions related to the network connectivity from the devices and handle them differently to reduce their cost.

We already discussed how the network provides mobility to hosts without requiring Layer 2 extensions and effectively allow any IP address to be used anywhere in the

https://t.me/learningnets

network. The implication of the way in which IP mobility is handled in a LISP network is that a lightbulb (or electrical outlet) may be given a fixed IP address at the manufacturing plant and yet connect to any network in any location without having to burden the lightbulb with running addressing protocols such as DHCP or even having to self-generate a link-local address in the absence of a DHCP server. Furthermore, because the mobility mechanisms that the LISP overlay uses override the behavior of Address Resolution Protocol and Neighbor Discovery with a normalized response for all queries (the MAC address of the first-hop xTR), the ARP and ND processes could also be eliminated altogether from the network stack of the lightbulb by reserving a well-known MAC address of link-local significance that can be used to identify a router in a point-to-point link. So, in this utopian network, lightbulbs would be connected and network protocols would be reduced to a minimum—thus maximizing functionality, reducing cost, improving reliability, minimizing exposure to attacks by reducing the OS footprint, and benefiting both users and administrators as they optimize their experiences.

SUMMARY

We live in a connected world in which the requirements for connectivity continue to increase. The proliferation of personal devices is one accelerator of the connectivity demand; the other is the Internet of Things. The access network is challenged with the task of providing connectivity for an unprecedented number of devices, most of which will be mobile. At the same time, the network is challenged with being easier to manage and has a unique opportunity to provide added value to the large number of connected devices and the applications they support. The level of scalability and the degree of flexibility required calls for a revision of the fundamental principles regarding how the network is architected and how the networking information is handled. LISP provides an architecture that fundamentally supports the scale and flexibility desired. As discussed in this chapter, a rich set of mechanisms in LISP plays a pivotal role in how networking is provided to the growing connected world.

https://t.me/learningnets

Chapter 6. Security

As more and more critical information is trusted to the digital infrastructure, the risk of information being compromised increases. Furthermore, as more devices are connected to the network, the paths by which criminals may compromise information are more numerous than they used to be. It is necessary to deliver a comprehensive and hardened set of security measures that allow the network to be the first line of defense in the IT security strategy.

Traditionally, security was added onto the network as a separate stack of solutions and functionality. LISP can offer a comprehensive set of security functions that are integrated into the networking control and data plane to deliver segmentation, access control policy enforcement, connection integrity, confidentiality for data in-flight, and endpoint anonymity. This chapter discusses how LISP provides integrated security services to improve the scale, flexibility, and manageability of the necessary security functions that the network must deliver. It also discusses how the LISP infrastructure itself is secured and protected from attacks that may attempt to compromise the network or use the network as a platform to launch an attack.

ATTACK SURFACES, LATERAL MOVES, AND BOT-NETS

Malware has become one of the main risk factors to an IT operation, compromising the availability of IT services as well as the integrity and confidentiality of the data handled by the IT infrastructure. Malware propagates through a simple two-step process in which it first infects the most vulnerable/available host found. Then it leverages any trust given to that host to move laterally and infect other hosts that may have richer information and broader access privileges that, in turn, may enable access to critical data and computational assets. As more and more businesses rely on the IT infrastructure, the number and types of connected devices increase. This increase in the number of connected devices broadens the attack surface and represents an improved opportunity for attackers to infect that initial host from which the infection may propagate laterally until it secures enough information to move into the data center. It can compromise data and applications or even cause a denial of service (DoS) attack without even moving into the data center by leveraging what could potentially be a

https://t.me/learningnets

large network of malware agents or what is commonly referred to as a *bot-net*. One common DoS attack is in the form of hijacking for ransom, where the infected computer is rendered inoperable and instructions on how to pay the ransom using nontraceable crypto-currencies are delivered to the infected computer as part of the attack.

Traditional security practices are centered around creating a hardened perimeter to keep untrusted endpoints outside the trusted perimeter from accessing applications and other IT assets. However, an authorized user is able to connect through the trusted perimeter and access applications without much guarantee that they aren't infected. The integrity of authorized hosts is meant to be preserved by maintaining the latest security patches on those trusted hosts. However, in many cases the security patches cannot protect the hosts in a timely manner; in other cases, the hosts may be necessary legacy devices unable to run the levels of code required to address the security vulnerabilities. The next line of defense is to ensure that the radius of connectivity of any host is truly limited to what it absolutely needs to reach. This is where network segmentation plays a key role.

POLICY, SEGMENTATION, AND THE VIRTUAL PERIMETER

The network perimeter is always a key area of concern and investment for IT operations. One of the main lines of defense that security teams implement is a hardened perimeter that separates the internal network from the external network. You can think of this perimeter as a wall separating two territories (in this case, the internal and external networks). An impenetrable wall certainly gives your internal network maximum security by not allowing any communication between the inside and the outside. However, the main reason to build the network is to enable communication. Thus, communication between the inside and outside networks must be allowed and carefully controlled so that only absolutely necessary communication takes place. So, a simple wall won't do. Maybe the wall should have a gate, and at the gate guards should enforce access controls and restrict the communication to the absolutely necessary. A more sophisticated (and secure) model involves the use of two walls and a buffer zone in between them where communication takes place securely. In geo-politics, this buffer zone is referred to as a *demilitarized zone* (DMZ). In networking, we use this term to refer to the network segment separating the internal and external networks. The DMZ hosts internal services that must be available to the outside network and also connects to the internal network. To provide connectivity to these services while keeping the internal network secure from the external network, the DMZ uses a front-end or perimeter firewall to inspect and restrict all connections between the external network and the DMZ. Similarly, a back-end or internal firewall is deployed to inspect and

https://t.me/learningnets

restrict all connections between the DMZ and the internal network. A DMZ model is shown in Figure 6-1.

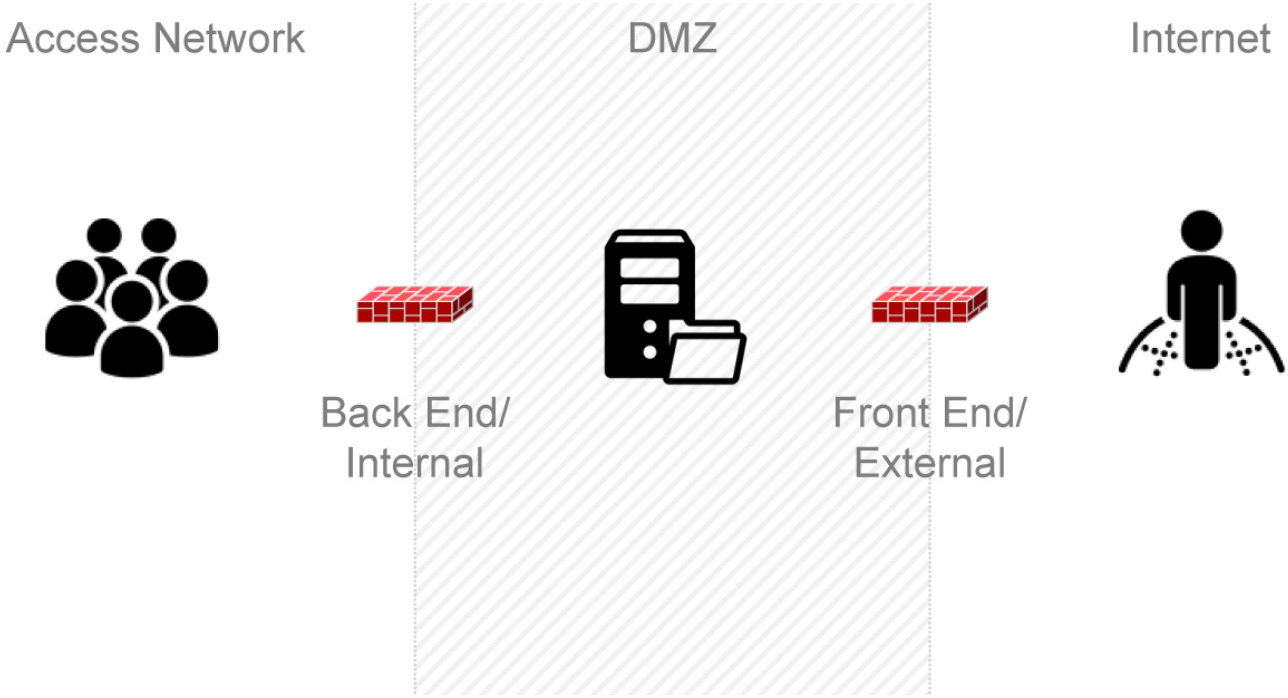


Figure 6-1 *The Demilitarized Zone at the Network Perimeter*

Real-life DMZs have strict security controls at each border. Highly trained (usually armed) guards enforce access restrictions into the militarized zones adjacent to the DMZ. In the IT DMZ, application layer firewalls are required both at the front end and back end to inspect the shape and contents of the traffic traversing the DMZ. Stateful Layer 4 inspection does not suffice because attacks could impersonate valid traffic profiles up to Layer 4 and traverse the DMZ undetected, similar to how a spy uses a false identity to cross a DMZ and penetrate enemy lines unnoticed. Deeper inspection of the traffic in-flight could prevent malicious traffic from crossing into an unauthorized network through the DMZ, but this does imply that the signatures on the application layer firewalls are capable of detecting the latest threats.

One notable challenge with the DMZ is that it is as secure as its firewalls are, and because the threat landscape is constantly changing, firewalls must be constantly updated. The signatures on firewalls could become outdated, making a compromised perimeter a distinct possibility. Another challenge with the DMZ model is that it was traditionally constrained to a specific physical topology and is therefore static and expensive to operate. LISP helps address these issues by hardening the internal network with segmentation and simplifying operations by enabling the creation of a virtual DMZ through a combination of virtual networks and dynamic service insertion.

The perimeter may be breached when infected traffic travels undetected across the firewalls and proxy devices in the DMZ. This breach gives the attacker the opportunity

https://t.me/learningnets
Technet24

to install its malware at the application servers inside the firewalled perimeter. From that point onward, the infection may spread to any host that can be reached from the infected device. If the internal network is not segmented, the scope of the attack may be unbound because there isn't a firewall or other policy enforcement point to prevent the malware from propagating within the internal network after the malware is past the firewalls. If the network is segmented, the infected host may propagate the malware only to the subset of hosts within its segment. The more granular the segmentation, the smaller the potential radius of the attack. Segmenting the network inside the security perimeter adds an additional line of defense by constraining the impact of any malware that gets past the perimeter. This is a way to virtualize the perimeter by creating multiple internal virtual networks; the perimeter indeed appears as multiple perimeters, which raises the bar for attackers who now need to be much more precise in their attack. If attackers succeed, their potential attack scope is also constrained, reducing their scope of influence significantly.

This notion of a virtual perimeter can, and should, be extended to the access network. Restricting the spread of malware after it gets past the firewalled perimeter is one important security measure to contain the impact of a perimeter breach. Restricting the available attack scope among clients in the access network may even prevent the malware from getting to a client with enough access rights to serve as a cover for the malware to breach the perimeter. Segmenting the access network is a way of hardening the perimeter and extending the perimeter virtually to the network access ports so that it becomes effective even before traffic is sent to the perimeter firewalls. Figure 6-2 shows the different policy enforcement points at play when a perimeter is virtualized by the use of segmentation in the internal and external networks.

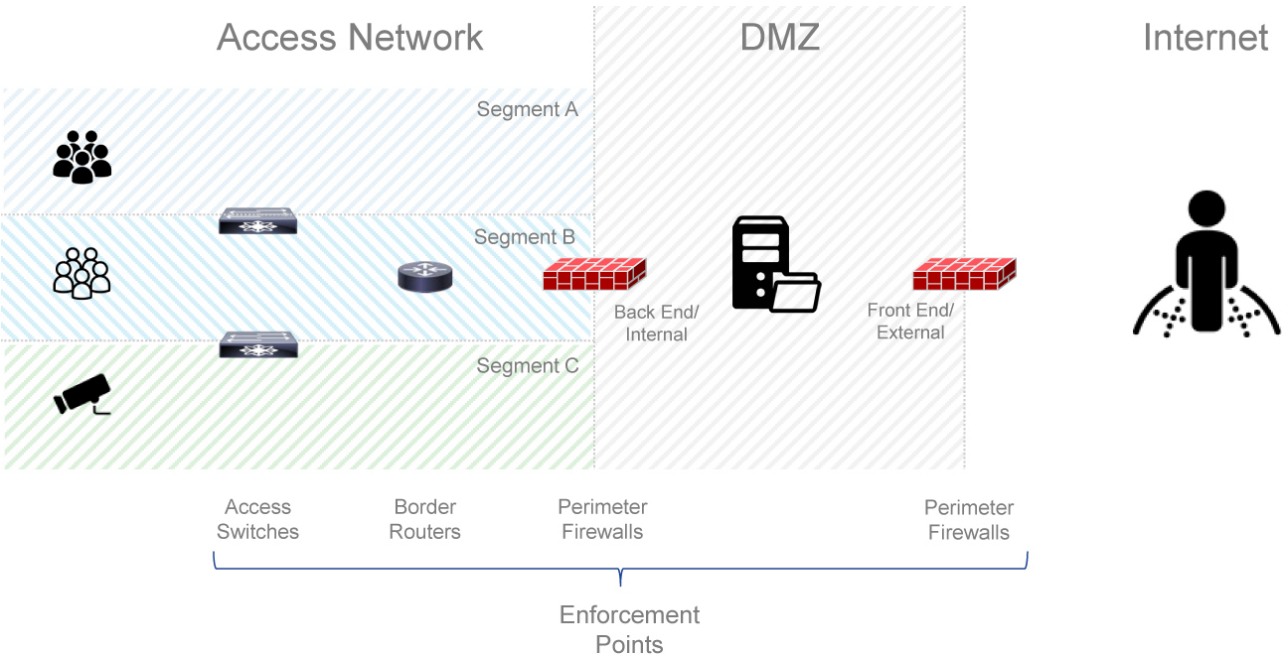


Figure 6-2 *Security Enforcement Points in the Virtualized Network Perimeter*

https://t.me/learningnets

One simple observation derived from the virtual perimeter depicted in Figure 6-2 is the rich matrix of policy relationships that are enforced by the virtual perimeter. Previous to the segmentation of the network, firewalls controlled the connections between users and applications. With the segmentation of the internal network, the communication between applications and their components is constrained to the absolutely necessary. The segmentation of the access network controls user-to-user communication. The virtual perimeter addresses all dimensions of possible communication: user to application, application to application, and user to user. Figure 6-3 illustrates the different dimensions of policy enforcement in the virtualized perimeter.

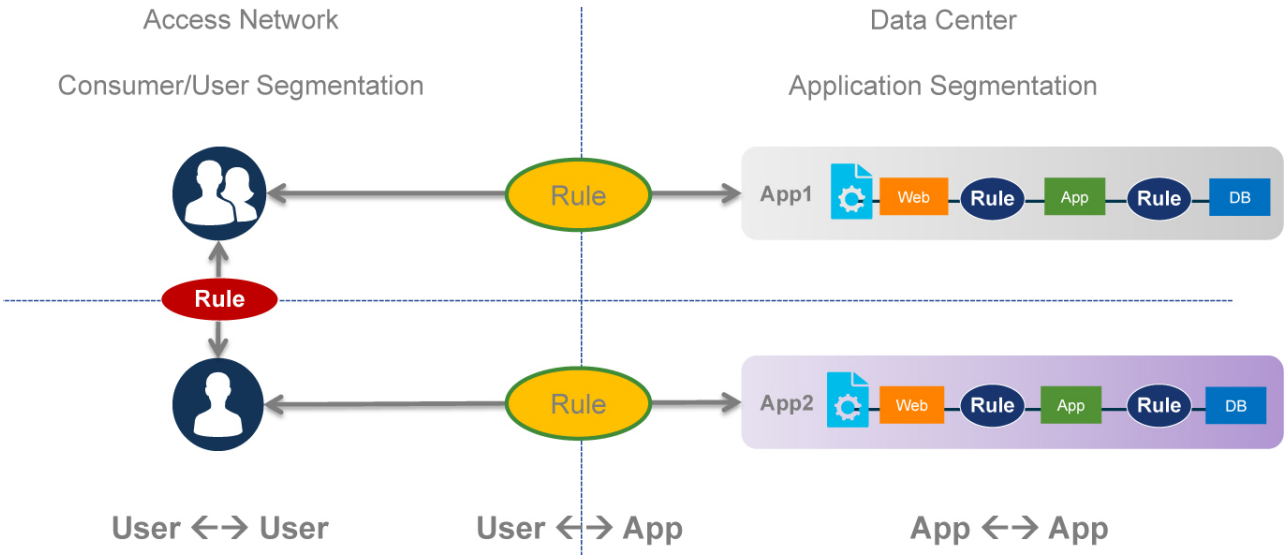


Figure 6-3 *Dimensions of the Security Enforcement Vector*

Based on the observations discussed so far, we hope it is evident why network segmentation took on such a prevalent role in securing the IT infrastructure. Network segmentation is procured at different levels of granularity:

• Virtual network level, which is referred to as *macro-segmentation*. This is what you would expect from the creation of multiple virtual private networks.

• Endpoint level, which is referred to as *micro-segmentation*. The kind of communication segregation achieved here is equivalent to what you would expect to achieve by using access control lists (ACLs).

• Software-process level, which is referred to as *process-level segmentation*. The enforcement of the segmentation is done through policy that governs a particular process's capability to initiate a connection to another process.

LISP enables the mechanisms required to provide macro-segmentation and micro-segmentation in the network. LISP may also provide the necessary machinery to enforce process-level segmentation in a container networking stack implemented using

https://t.me/learningnets
Technet24

LISP.

Macro-segmentation

LISP inherently delivers macro-segmentation by enabling simple mechanisms to implement Virtual Private Networks (VPNs). Chapter 3, "Data Center Trends," discussed in detail how LISP implements VPNs by using extended endpoint identifiers (EIDs) and Instance-IDs to partition the EID space, segment the Map-Caches, and color the control and data plane messages to create virtual networks. The functionality required to build a virtual network is broken into three main components:

- Control plane segmentation
- Data plane segmentation
- Device segmentation

LISP tags the control plane with a 32-bit Instance-ID and the data plane with a 24-bit Instance-ID to identify control plane messages and data traffic as belonging to a particular segment. To tag the control plane, the 32-bit Instance-ID label is added to the EIDs to create an extended EID (XEID) so that the LISP nodes can multiplex and demultiplex control plane messages into different segments without altering the messages in the protocol. The Mapping Database System leverages the Instance-IDs to populate different partitions of the registration table (one per Instance-ID). Figure 6-4 illustrates the tagging of the control plane with the instance-ID.

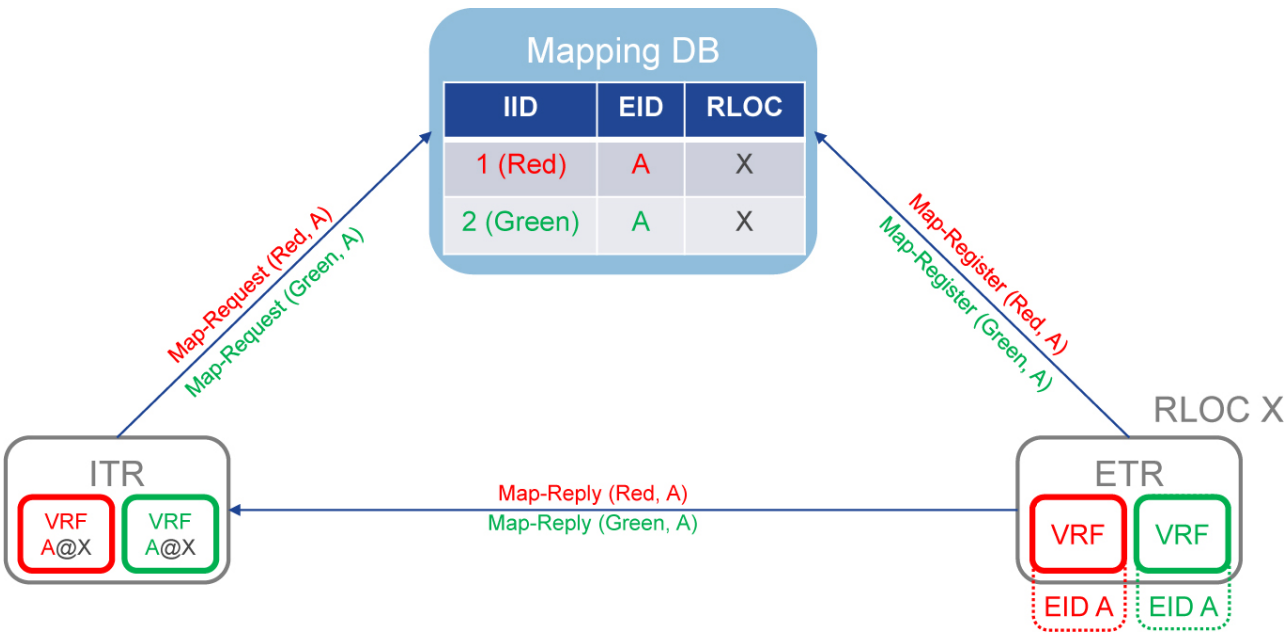


Figure 6-4 *Virtual Network Control Plane Segmentation*

To tag the data plane, you include the Instance-ID in the shim header of the tunnel

https://t.me/learningnets

encapsulation used to forward traffic between xTRs. Only 24 bits are used for the data plane to accommodate hardware constraints. How they map to the 32 bits in the control plane is implementation specific. Including the Instance-ID in the shim header allows the multiplexing and demultiplexing of LISP traffic between the forwarding partitions created on the xTRs and the tunnels used to transport traffic between the xTRs. Figure 6-5 illustrates the segmentation of the data plane and how the Instance-ID is used to keep traffic segmented as it travels between xTRs.

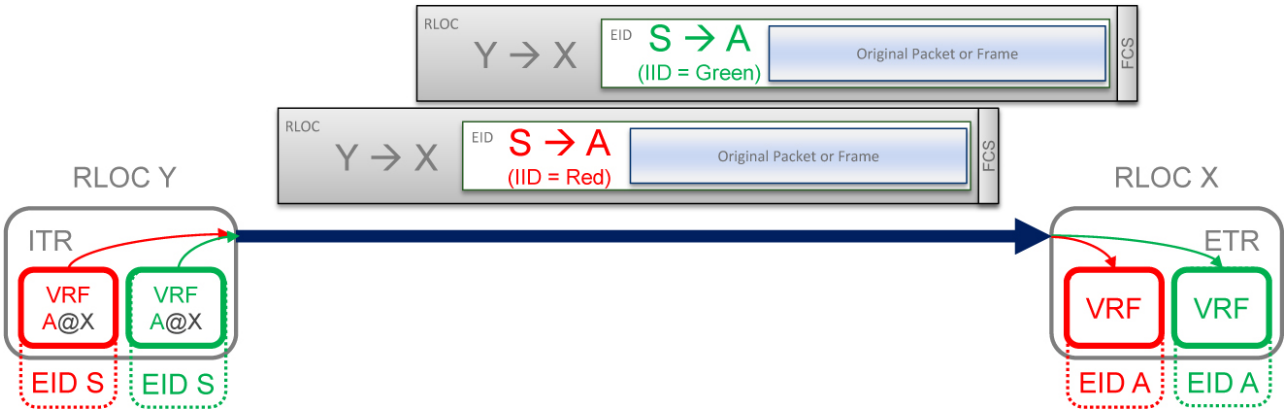


Figure 6-5 *Virtual Network Data Plane Segmentation*

The forwarding devices (tunnel routers, or xTRs) must also be segmented. To this effect, the routing/forwarding table is partitioned to create a separate forwarding context per Instance-ID. The coloring of the control plane is used to populate routing and forwarding partitions in the xTRs, as illustrated in Figure 6-4. Each virtual network instance (or segment) has its own routing and Map-Cache partition on the xTRs. Figure 6-5 shows a virtual routing and forwarding (VRF) instance as the forwarding segment on the ingress tunnel routers (ITRs) and egress tunnel routers (ETRs). However, this segment can be implemented as a VRF, a virtual LAN (VLAN), or a bridge-domain depending on the address family being handled and the capabilities of the xTR platform. Traffic is encapsulated and de-encapsulated in the context of the different forwarding partitions. The coloring of the data plane is leveraged to maintain data traffic in the context of its partition as it is encapsulated to travel between xTRs. Figure 6-5 illustrates how the Map-Cache partitions are mapped to the Instance-IDs in the tunnels. Chapter 3 discussed how to leverage the mapping of forwarding tables to Instance-IDs to hop across virtual networks and support controlled cross-VN communication for the delivery of shared services across network segments. Traditionally, providing such a service required an exchange of all routing information between virtual networks (cross-VRF route leaking), which led to an inefficient multiplication of routing state across VRFs. In LISP, this cross-VN reachability is implemented simply as a lookup policy without unnecessarily replicating state across VNs.

https://t.me/learningnets

Technet24

These virtual networks are a seamless extension of the way EIDs are normally handled in LISP and therefore enjoy all the benefits of scale, mobility, and address family independence that LISP provides. In other words, the benefits of LISP and the operational environment for LISP are equal in the presence or absence of virtual networks. For instance, one salient benefit of the LISP architecture is that unicast traffic and multicast traffic are handled similarly, which allows any operation to deliver multicast without additional operational cost. This benefit continues to be realized without change in the presence of virtual networks, including the ability to provide extranet services for multicast traffic. Operationally and conceptually, the addition of the virtual network does not impose any changes to the LISP control plane. It is seen simply as an extension of the EID definition. In a way, virtual networking in LISP is seen as the extension of the EIDs to be 24 bits longer, where the upper 24 bits simply identify a forwarding context.

Previous chapters discussed how the demand-based nature of LISP allows it to scale to a large distribution. The scale and flexibility with which virtual networks can be created and managed in LISP are unparalleled. Thus, LISP provides an optimal control plane for the creation of virtual networks to deliver macro-segmentation.

Micro-segmentation

LISP enables micro-segmentation in a few different ways. It enforces the policies directly in the control plane; it assists data plane enforcement mechanisms such as the Cisco TrustSec; and it provides dynamic firewall insertion services to support those cases in which stateful enforcement of the policy is required. Let's look at these three aspects of micro-segmentation in more detail.

The mission of the network is to provide connectivity between endpoints. Different endpoints have different roles and different permissions. Some endpoints host applications; other endpoints host clients to those applications. In a given IT operation, you can find a finite number of applications and client types. To define an effective access control policy, you should group the different types of endpoints according to their role and rights. With the endpoints classified into groups, you define the policy as the relationships allowed or disallowed between these groups of endpoints. Traditionally, IP addresses were used to identify different groups of endpoints. IP based micro-segmentation was a large driver behind restrictive IP address management practices that seek to align the IP prefix space with the actual groups of endpoints. However, aligning IP addresses with endpoint roles becomes harder and harder as devices proliferate, endpoints become more mobile, and roles overlap across devices. Rather than leveraging IP addresses to identify the group to which an endpoint belongs,

https://t.me/learningnets

you can assign to endpoints the tags or labels that reflect their membership in one or more groups. Group based micro-segmentation allows endpoints to be identified as members of a group without relying on their IP address. The group-based method of identification results in group-centered policies that are defined and enforced independent of the IP addresses assigned to the endpoint. If the address assignment changes, the policies are not affected. Using groups rather than IP addresses to define and enforce policy also has implications on the required capacity of the policy enforcement devices that have to handle a relatively small handful of groups rather than a large number of IP addresses.

LISP enforces policies between endpoint groups in the control plane before a connection is even established. Rather than inspecting traffic in flight and enforcing access policies in the data plane as done in traditional access control lists, the LISP control plane constrains reachability at the time it is being consulted to establish a connection. Remember that LISP is a demand-based protocol in which policies can be defined in the Mapping Database System and enforced when a Map-Request is issued for a particular destination. Micro-segmentation policies are defined among endpoint groups. The groups may be assigned a group tag that identifies an endpoint (and its traffic) as belonging to a particular group. To enforce the micro-segmentation policy, the network must be able to determine which group the source and destination endpoints belong to. To determine which group an endpoint belongs to, LISP inspects the IP prefix of the endpoints or the group tags assigned to the endpoints. The LISP control plane enforces policy based on these groups either by looking at IP addresses or tags. When using group tags, you must register the tags along with the EIDs.

When policies are enforced in the LISP control plane, a malicious user may intentionally try to communicate with endpoints that are restricted in an attempt to launch a DoS attack on the Mapping Database System. LISP prevents this type of situation by having the ITRs cache the results of a lookup. This keeps the rate of events sent to the Mapping Database System to a minimum because events that have a match in the cache will not trigger a lookup in the Mapping Database System. When an ITR caches a mapping for a destination, it will do so with a directive to drop, redirect, or forward the traffic. The directive provided will match the policy. This directive prevents further queries for the destination from being sent to the Mapping Database System. The Mapping Database System also replies with a mapping for the largest covering prefix that includes the destination, which further reduces the number of events that would result in a query to the Mapping Database System.

Micro-segmentation policies are enforced in the data plane based on IP addresses or based on data plane metadata that reflects the grouping of endpoints independent of

https://t.me/learningnets

Technet24

their IP addresses. Chapter 3 discussed at length how LISP complements the enforcement of micro-segmentation policies based on endpoint groups by enabling the derivation of remote endpoint group membership at the ingress tunnel routers to enforce policies based on source and destination group information. The enforcement of a policy requires that both the source and destination group information are known at the enforcement point. By registering group membership along with the EIDs in the Mapping Database System, the LISP control plane provides the necessary source and destination group information necessary for an xTR to enforce policy that involves endpoints that may not be directly connected to the xTR and for which the group information wouldn't normally be known.

Micro-segmentation policies may also be enforced on stateful devices such as firewalls. LISP provides firewalls with the necessary grouping information for the firewalls to be able to enforce policy on EIDs that are not directly connected to it. Firewalls may not always be on the path of the traffic they must enforce policy for, but LISP can alter the traffic path and steer the traffic through the relevant firewall. Chapter 4, "The Wide-Area Network: Bringing Traffic from Access to the Data Center," introduced the concept of logical topologies that are dynamically created in LISP to steer traffic through service nodes and enable service insertion.

Process-Level Segmentation

Macro-segmentation refers to the separation of endpoints into virtual networks within which any communication is permitted, whereas micro-segmentation refers to the enforcement of rules that restrict communication among groups of IP hosts (endpoints) to specific types of traffic. Process-level segmentation is a form of micro-segmentation that is applied at a more granular level than the host, the workload level. In general, process-level segmentation manifests itself as the segmentation of containerized workloads.

You can think of a container or process as an endpoint. The definitions for micro-segmentation and process-level segmentation are basically the same: the enforcement of rules that restrict communication among groups of endpoints to specific types of traffic, where endpoints can be defined as IP hosts (micro-segmentation) or as workloads within an IP host (process level segmentation).

The application developer defines process-level policies, and therefore, you could think of them as being implemented within the application and having no relation to the network. However, these policies, once authored, are handed over to the network stack of the application environment for enforcement. For instance, in a container

https://t.me/learningnets

environment, these policies manifest themselves as a combination of a logical network topology and an IP list. The logical network topology can be in the form of a basic bridge or an overlay. IP lists are basically ACLs that are enforced at select virtual network interfaces in the container virtual network.

The role of the logical network topology in process-level segmentation is of interest because it may determine the placement of the policy enforcement points. The two main logical topology types that are commonly found in containerized environments are bridged logical networks and overlay logical networks. These networks may be deployed in the default mode provided by the container infrastructure in use or may be customized.

Running a LISP xTR in a container and attaching the LISP container to a bridge logical network could, in theory, allow the LISP container to provide routing and LISP services for the logical bridge network. The LISP container then would perform both control plane and data plane duties. Figure 6-6 is an example.

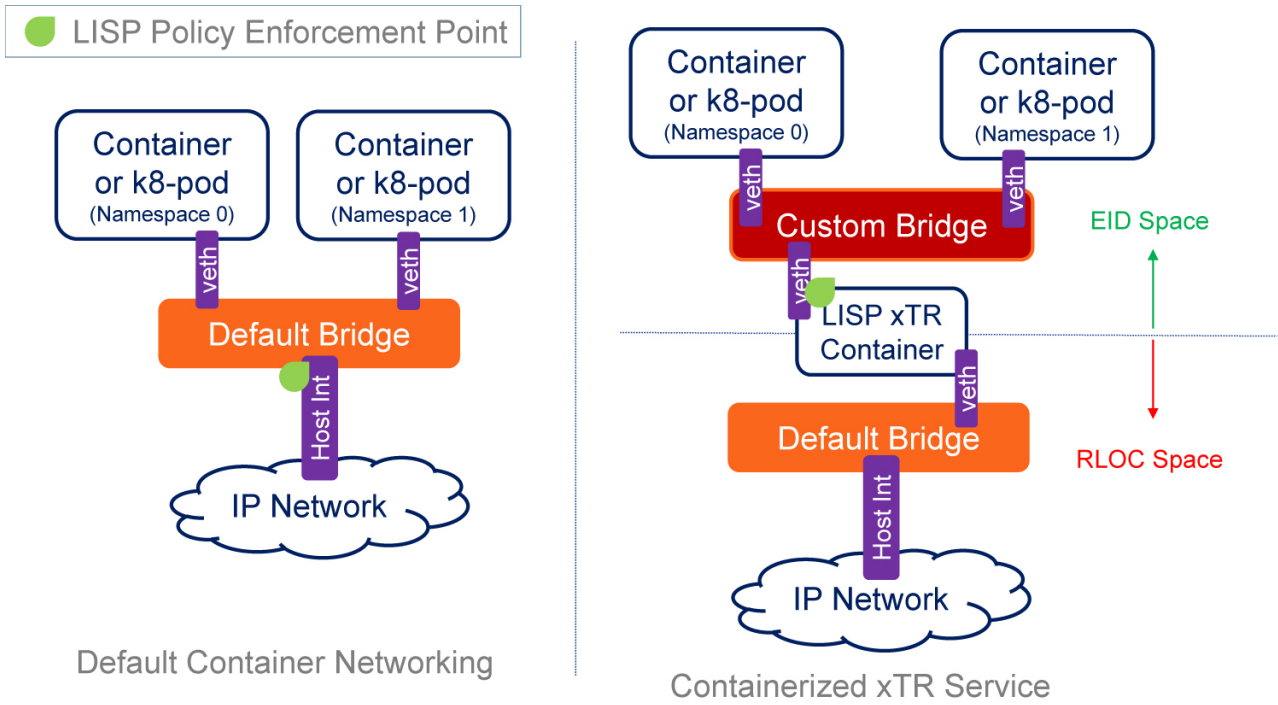


Figure 6-6 *Containerized LISP xTR as Default Gateway on Logical Bridge*

Overlay logical networks are provided natively in most container environments. They are instrumental in providing secured communication between containers hosted in different compute nodes. These overlays can be customized, in which case the overlay may be a LISP overlay. The overlay network starts at each container bridge. In the LISP case, this means each container bridge connects directly via an xTR service to the network. In this scenario, LISP has the necessary footprint to enforce the policy between containers running on different nodes in the control plane if desired or to

https://t.me/learningnets
Technet24

assist the data plane in using more scalable mechanisms for group-based policy enforcement that is independent of IP addressing.

One straightforward way to deliver a LISP-based overlay network in the container stack is to create a custom plug-in for the stack. This may or may not be an option, depending on how much flexibility is given by the cloud provider where you intend to deploy the workloads. A custom plug-in allows the LISP control plane to not only control the overlay across hosts but also control the forwarding of the local bridges. As shown in Figure 6-7, this configuration gives LISP a footprint to enforce policy for all communication among containerized workloads. The benefit of this approach is that policy is defined and enforced in a consistent manner across containerized and noncontainerized workloads, enabling security operations personnel to normalize their policy definitions and audit practices.

https://t.me/learningnets

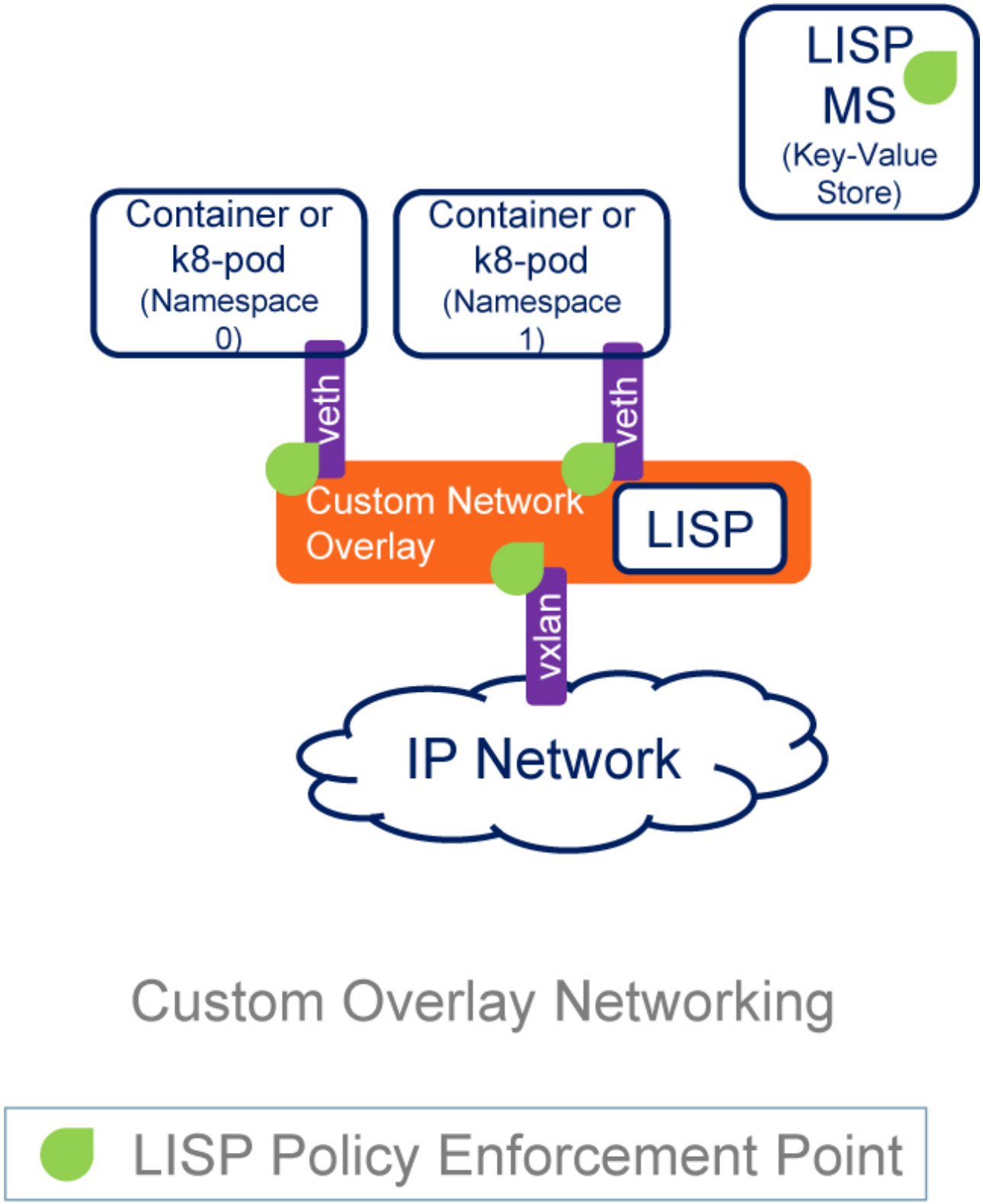


Figure 6-7 *LISP-Based Container Overlay Custom Network*

Overlay networks in container environments traditionally required an external key-value store. Some examples of key-value stores frequently utilized include Consul and Zookeeper. As seen in Figure 6-7, when LISP is the overlay, the key-value store is effectively the LISP Mapping Database System. This example is comparable to the type of service the Calico networking services plug-in provides to Kubernetes containers, where Calico provides a complete suite of networking services including IPAM, external routing, policy management, and of course, its own key-value store. LISP provides a network overlay that Calico could leverage or could enable a replacement to something

https://t.me/learningnets
Technet24

like Calico.

Having this kind of consistency provides significant benefit. One obvious benefit is operational such that networking is consistent across domains and the interface between a container environment and the rest of the network is simplified by using a common approach. From a security and segmentation perspective, the LISP control plane can be extended to include group semantics that are instrumental in delivering micro-segmentation at the host level and, with the introduction of LISP in the container stack, at the workload level. As discussed, LISP enforces group policies in the control plane, allowing the segmentation policy and functionality to evolve independently and rapidly in the user space without being constrained by any dependencies on data plane or kernel capabilities.

In the context of container overlay networking, cryptography is a key requirement. This chapter studies how the LISP Mapping Database System serves as a key exchange facility to support large-scale pairwise private key computation without creating a central store. Most solutions in the market leverage the central key-value stores used for overlay resolution to also be central stores for cryptographic key material. From a security standpoint, central stores are vulnerable and a well-understood attack vector that should be avoided whenever possible. A LISP implementation of the container networking stack delivers large-scale cryptography without the risks of key information centralization.

The LISP control plane can be extended to accommodate different semantics. We already discussed how to extend a mapping to include a policy group tag. In the same vein, you can characterize endpoints in LISP as a tuple of IP and port number to accommodate the granularity introduced by containers and extend the micro-segmentation behavior to the container level.

One popular service in container networking is load balancing. This service is embedded in the network stack. When LISP is the enabling technology for the container stack, the load is balanced, leveraging the LISP system of priorities and weights across a set of multiple RLOCs. In the case of the granular segmentation delivered by containers, the definition of the RLOC is extended to include port numbers in a tuple that consists of ports and RLOCs to identify the different workloads and assign the workloads weights for load balancing.

How to Integrate the Control Plane into the Assurance Loop

The security policy enforced in the different levels of segmentation attainable with LISP

https://t.me/learningnets

is only as effective as the mechanisms available to verify that the policy is indeed being properly enforced. Modern networks must have mechanisms to continuously provide assurances that the policies and desired network behavior are being properly enforced.

Because of its transactional nature, the LISP control plane centralizes transactional events that are normally distributed and also centralizes the enforcement of many of the policies that must be subject to assurance controls. By centralizing transactional events and also the enforcement of policy, the LISP control plane provides a logically centralized point of telemetry collection and audit for an assurance system to verify a large portion of the behavior that must be verified. The logical centralization of the control plane significantly simplifies the task of providing assurance controls by reducing the scope of places from which information is gathered. Not all required information is centrally available in the control plane, but the fundamental information to cover a large percentage of the assurance data set is centralized. The assurance system can then devote valuable compute capacity to tasks beyond the fundamental collection of telemetry, verification of policy, and network behavior.

The same principle of logical centralization of transactions and policy enforcement is leveraged to make the network-enforced policy adaptive. Part of a modern network includes the ability to constantly monitor traffic patterns and connectivity behavior. Some of this information may be derived from LISP, but some systems are designed to monitor application traffic directly and generate a baseline of acceptable behavior. These systems are capable of reacting to deviations from the baseline that may be considered security threats or even performance-impacting situations. When such deviations are detected, the monitoring system may notify (directly or via a security policy orchestrator or controller) the LISP control plane and centrally enforce a change in the policy for any hosts that were identified as potentially compromised by the change in behavior. This notification could be in the form of inserting a traffic inspection engine in the path or quarantining the impacted host into a different macro- or micro-segment.

Traffic Steering and Service Chains

The ability to steer traffic conditionally is key to a segmented network where the enforcement of segmentation may require stateful controls and therefore the steering of traffic through devices such as firewalls. This capability, often referred to as *service chaining*, is integrated in LISP as policy. Chapter 4 discussed how logical topologies are implemented in LISP and how LISP dynamically steers traffic through services based on the same policy definition that drives the network micro-segmentation.

https://t.me/learningnets

Technet24

CRYPTOGRAPHY IN LISP

Data privacy, integrity, and authentication are critical aspects of a security solution and must therefore be integrated into a LISP-enabled network. Cryptography is instrumental in providing data privacy. The mechanisms by which cryptography is implemented are also key in assuring the integrity of the data. Additionally, digital signatures to authenticate information sources are based on public-key cryptographic methods.

One obvious way of providing cryptography in a LISP network is to combine LISP with a separate crypto solution. You can find ample literature and discussion around this topic in the public domain. For that reason, we focus instead on the scenario in which cryptography is built into the LISP control and data planes.

One key aspect of providing cryptography is the ability to secure reliable key material for the parties involved in the communication to use for encryption and decryption. There is always a privacy component to the key material being handled. The key material may be exchanged between the interested parties leveraging public-key algorithms, a secure channel, or key-exchange protocols. Public-key algorithms tend to be used with asymmetric cryptography, whereas symmetric cryptography leverages secure channels and key-exchange protocols.

Public-Key Cryptography

Public-key cryptography is based on the pairing of public and private keys to enable asymmetric encryption. Public keys are openly advertised and mathematically paired with a private key that is not advertised at all. Asymmetric cryptography uses one key to encrypt traffic and the other key in the private-public pair to decrypt the traffic.

Whether the public key is used for decryption or encryption depends on whether the keys are being used to encrypt data or to digitally sign a transmission. For instance, in the Rivest-Shamir-Adleman (RSA) algorithm for data encryption, any sender may use the public key to encrypt traffic to the recipient, but only the private key may successfully decrypt the traffic. Because the receiver keeps the private key a secret, only the receiver may decrypt the traffic. In the Digital Signature Algorithm (DSA), the public key is used for decryption of digital signatures that are produced by a sender using the private key in the pair. Because the sender is the only one in possession of the private key, only the sender can sign messages with this key pair; anyone else can retrieve the public key and verify the signature. Figure 6-8 illustrates the use and location of public and private keys for the application of asymmetric cryptography to digital signatures and data encryption.

https://t.me/learningnets

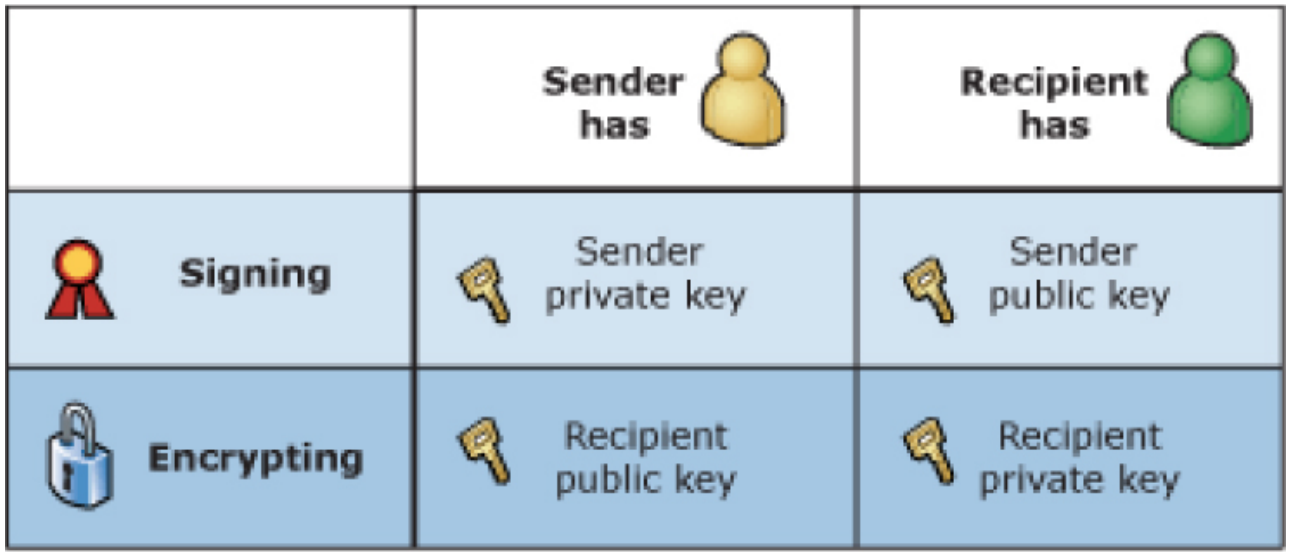


Figure 6-8 *Applications of Public-Private Key Pairs*

The public and private keys are mathematically paired in a cypher or hash so that this combination of encrypting with one key and decrypting with another is possible. The mathematical functions that correlate public and private keys are one-way functions. One-way functions are, in theory, nonreversible, so private keys cannot be derived from their corresponding public keys. The existence of true one-way functions remains unproven. So, in practice, one-way functions are functions that are computationally impractical to reverse. The amount of time and processing power required to reverse a one-way function is such that a private key cannot practically be derived from its public key during the time the key pair is valid. Cryptography systems are designed to change keys often and make the task of obtaining a private key from public-key material even more challenging.

The mathematical operations required to calculate the key pairs used in asymmetric cryptography are also computationally expensive. Some of the cryptographic algorithms commonly used are based on mathematical problems for which it is widely acknowledged that no efficient solution is known. One example is the use of elliptic curve relationships. Elliptic curve cryptography is capable of producing smaller keys that deliver an equivalent level of security to that achieved by larger keys in other algorithms. The Elliptic Curve Digital Signature Algorithm (ECDSA), as its name suggests, is based on the Elliptic Curve algorithm and is leveraged by LISP to sign Map-Registrations and Map-Requests.

The public-key algorithms used for asymmetric cryptography do not require a secure communication channel for the exchange of initial private key material between the parties. However, because the level of computation is high, asymmetric cryptography is usually used to protect the transmission of small amounts of data, but not the entire data stream. Thus, in practice, asymmetric cryptography is used to provide a secure

https://t.me/learningnets
Technet24

channel for the initial exchange of symmetric shared keys. The shared keys are then used to protect the actual data transmission at the much lower computational cost of symmetric encryption.

LISP leverages asymmetric cryptography mainly to provide digital signatures for Map-Registers and Map-Requests. Map-Replies are signed with a hash that leverages shared keys over a secured channel. These mechanisms are studied in more detail in the section "How the LISP Control Plane Is Secured."

For the data encapsulated in LISP, you leverage symmetric cryptography coupled with a key-exchange mechanism embedded in the LISP control plane. Next, let's see what symmetric-cryptography is about and how it works in LISP.

Symmetric Cryptography

Symmetric cryptography algorithms use a single shared key to encrypt and decrypt traffic. The mathematics involved with the ciphers in symmetric cryptography are much simpler than those involved in asymmetric cryptography. Symmetric ciphers are therefore much faster and better suited to provide privacy for large amounts of data. LISP includes a symmetric cryptography service for the encryption of data between xTRs. The mechanisms by which LISP provides symmetric cryptography and how the control plane enables shared key exchange are defined in RFC8061. These mechanisms are often referred to as *LISP cryptography*.

The symmetric cryptography mechanisms defined in RFC8061 integrate a Diffie-Hellman–based algorithm in the LISP mapping resolution message flow to achieve the private exchange of shared secrets between the ITR and ETR. The shared secrets derived from this Diffie-Hellman exchange are used to generate encryption keys as well as the message authentication codes (MAC) that the different cypher suites use. The choice of cypher suite to use for a particular ITR-to-ETR connection is also negotiated during the map-resolution exchange. All cipher suites leveraged in LISP cryptography provide authenticated encryption with associated data (AEAD). AEAD ciphers not only encrypt the information being transmitted but also generate MAC codes and message hashes to sign the messages and guarantee the authenticity and integrity of the code. By using AEAD ciphers, the LISP cryptography solution delivers confidentiality, integrity, and authenticity assurances on the data being sent.

RFC8061 specifies a key identity field that is included in the data plane header to enable the system to cycle through up to three precalculated keys. This capability optimizes the rekeying process by allowing the key computation operations for the

https://t.me/learningnets

generation of new keys to happen in the background and have up to three precalculated keys ready to be used immediately when rekeying is required. Each key is associated with a key identity between 1 and 3, and rekeying is as simple as changing the key-ID in the packet header from one value to the next. New keys can be calculated to replace old keys for a particular key-ID value while other keys are in use. Figure 6-9 illustrates the LISP cryptography data plane and the optimization of the rekeying process.

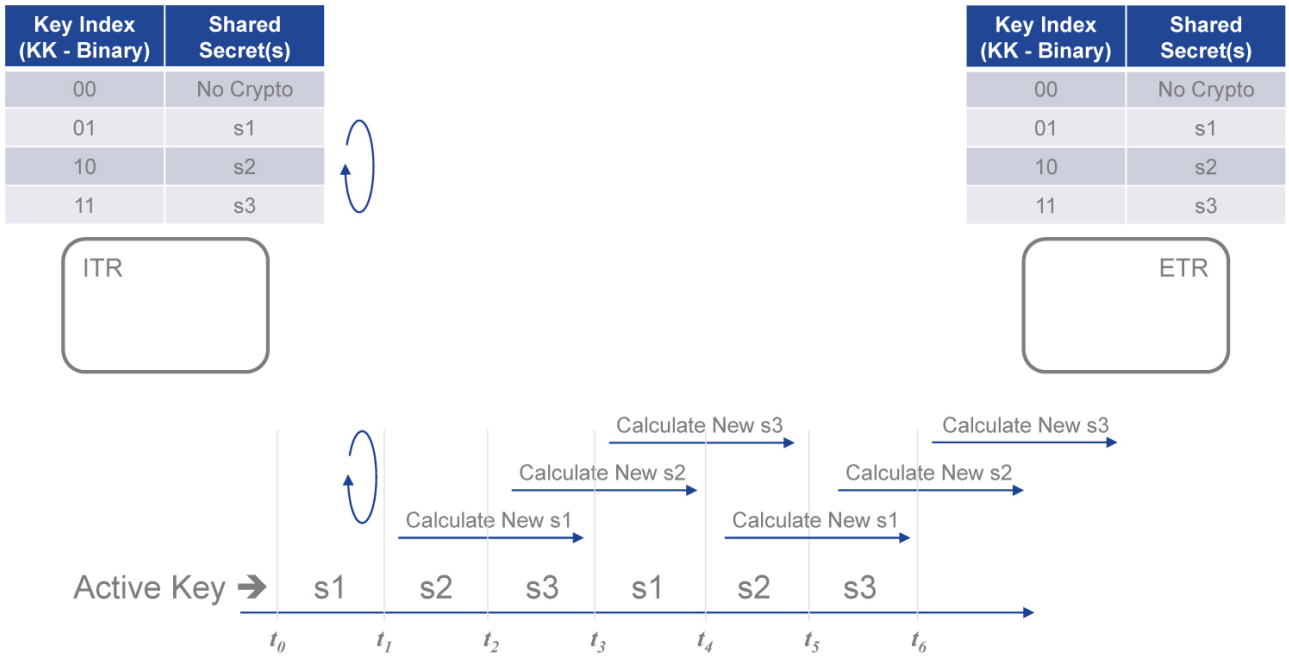


Figure 6-9 *LISP Cryptography Data Plane and Rekeying*

The main challenge behind symmetric cryptography is the exchange of the shared key. This private key cannot be exchanged openly. Historically, a secure physical medium was used to share the private keys among the parties. In many implementations, an asymmetric encrypted channel is used to exchange the shared keys to enable symmetric cryptography. A widely employed method is to use a key-exchange protocol such as the Diffie-Hellman protocol that allows the secure exchange of private keys over unsecured channels. This is the mechanism used in LISP cryptography. Next, we discuss key exchange and how it is implemented in LISP.

Integrated Key Exchange

The field of computer science is rich with stories that illustrate some of the most relevant problems that occupy computer scientists. Dijkstra's Dining Philosophers problem illustrated the challenge of contention for shared resources by concurrent processes. The Problem of the Traveling Salesman and their calculation of an optimal travel path illustrate a challenge in combinatorial optimization that may be only partially solved at moderate to large scale without absolute certainty. The Problem of the Two Generals is mathematically proven to be unsolvable and illustrates the

https://t.me/learningnets
Technet24

challenges of reliable communication across an untrusted medium. In 1982, Leslie Lamport from Microsoft Research famously introduced the Problem of the Byzantine Generals, who are expected to reach an agreement on an action to benefit their army, even in the presence of traitors in their ranks. The framing of these problems as stories has captured the imagination of many bright individuals over time and fueled major advances in mathematics and computer science. As for this humble book on the inner workings, applicability, and benefits of LISP, and how it may apply to the area of cryptography, we use a simple analogy of colors and color mixing.

Two people exchanging messages must find a way to mark their messages and verify their authenticity and integrity because these messages may travel in the hands of untrusted individuals. Their messages are handwritten, and all they have to work with are a few different colors of ink.

Because all they have are colors, these two people decide to play a clever trick of color mixing so that they can agree on a secret color that they can both identify as their shared secret. To understand what they did, let's give this couple random fictional names; say the two people are Bob and Alice. Each of them picks a private secret color that they won't share, and they decide on a public color that is known to both of them (and potentially others). So, three colors are at play: blue and red are Bob and Alice's respective private secret colors; yellow is their public color.

Let's see how they arrive at yet one more color that they can both share but that won't be known to others. Bob mixes his private secret color (blue) with the public yellow and sends Alice a card with his mix. When Alice receives this card, she adds her private secret color (red) to the mix on the card. The resulting color is a new secret color. At this point, only Alice knows the new secret, but this secret color is eventually known to both of them.

In return to Bob's card, Alice mixes her private secret color (red) with the public yellow on a different card and sends Bob the resulting mix. When Bob receives the card, he adds his private secret color (blue) to the mix on the card to produce the same secret color that Alice produced when she received his card. At this point, they have both arrived at the same shared secret color. Notice that they never shared their private secret colors (at least not without mixing), nor was the resulting shared secret color ever explicitly exchanged between them. However, they were able to arrive at the same color independently. So, the lucky couple found a way to create shared secrets that others cannot see or re-create. One key property of their method is that color mixing is a one-way function, meaning that a color mix cannot be reversed to re-create the original colors that were used in the mix. So even if their foes know their trick, they are

https://t.me/learningnets

not able to find out the couple's secret colors and produce the shared secret. The color exchange is illustrated in Figure 6-10. As shown in the figure, the method allows both parties to obtain all components of the shared secret color mix without transmitting any of the secret colors in the clear.

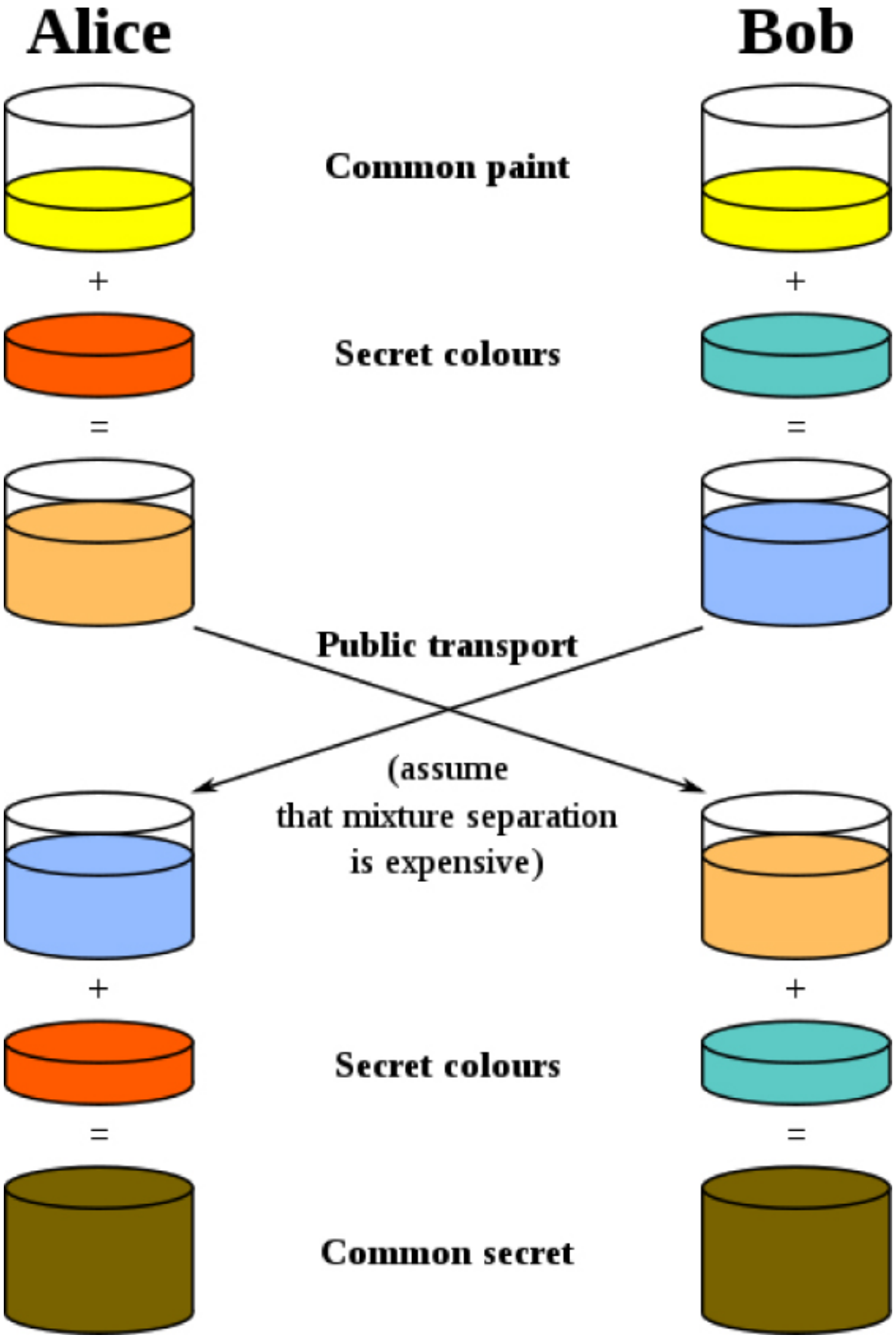


Figure 6-10 *Creating Shared Secrets with Colors*

https://t.me/learningnets

Technet24

This shared secret can be used in many ways. For example, when the couple exchange messages, they could write the messages using their private-public color mix. They could then validate the authenticity of their messages by adding their private color to the mix and comparing the resulting color with the shared secret color produced earlier. If the color matches, the message is authentic; if the color doesn't match, the message is fake.

The color mixing method illustrates (hopefully in simple terms) some of the basic principles used in exchanging key material over an unsecured channel. It also shows how this exchange is used to provide authenticity assurances for the messages being sent. If, instead of colors, the couple used numbers and mathematical processes to do a similar thing, the resulting key material could be used to authenticate, hash, encrypt, and decrypt traffic and provide privacy, integrity, and authenticity assurances. Let's look at how to model the color exchange as an algorithm with specific mathematical properties rather than a color-mixing heuristic. Let's also look at how LISP facilitates the exchange of parameters between the parties involved.

The method for the secure exchange of shared secrets over a public channel is widely known as the Diffie-Hellman (DH) key exchange and is named after Whitfield Diffie and Martin Hellman, who first published and patented the scheme in 1976. As it turns out, the British signals intelligence agency has known how to achieve public-key cryptography since 1969. The work of James H. Ellis, Clifford Cocks, and Malcolm J. Williamson at the British Government Communications Headquarters (GCHQ) remained classified and therefore unknown outside "The Doughnut" until 1997, when it finally became public. (The GCHQ is headquartered in a circular-shaped building that resembles a doughnut. The building is located in Gloucestershire in the southwest of England.) Despite history, the algorithm is referred to as Diffie-Hellman.

Think of the communicating parties now as the ITR and the ETR for a given connection. This example subtly reveals an interesting property of using LISP cryptography, which is the directionality of the keys. If communication is between xTR1 and xTR2, the key material calculated when xTR1 is the ITR and xTR2 is the ETR will be different than the key material calculated for the reverse direction. Thus, keys are unidirectional in LISP cryptography, and the more granular, elusive, and varied cryptographic key material can be, the more secure the solution is considered.

Now let's look at Diffie-Hellman in the context of LISP cryptography. The ITR and ETR each have a secret, i and e, respectively. The ITR sends two publicly known parameters, p and g, to the ETR so that they are known by both. Now, remember, when you are combining these parameters, the operation must be mathematically unidirectional, so p

https://t.me/learningnets

and g have certain restrictions. The parameter p must be a prime number, and g is a primitive root modulo p. If you are not familiar with the specifics of what this means for p and g, don't worry. Know that they can't be just any number and keep in mind that the value of p determines how wide the shared secret can be. With the private and public parameters in place, the rest is a simple process of applying the Diffie-Hellman formula and exchanging the results (this is the equivalent to mixing colors and exchanging cards). Figure 6-11 illustrates the parameter exchange and the mathematical operations used to generate public and private keys.

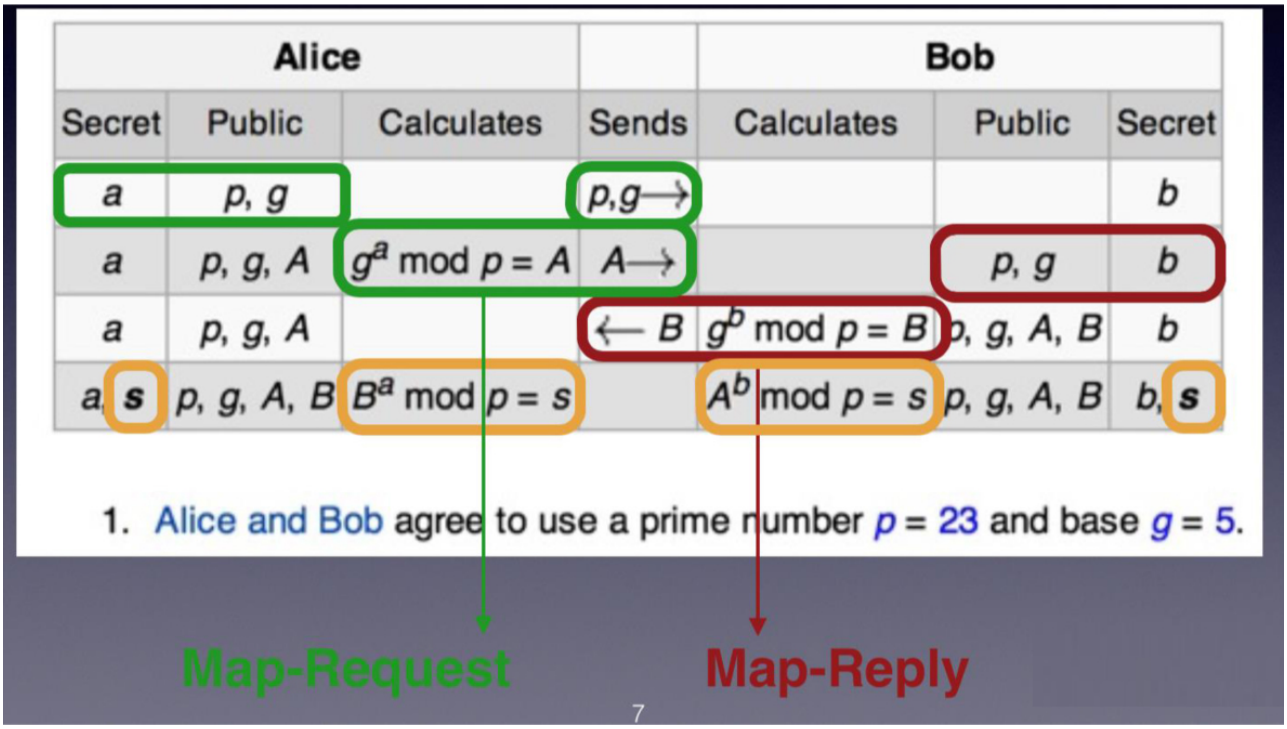


Figure 6-11 *Diffie-Hellman Key Exchange (DH) in LISP Cryptography*

As shown in Figure 6-11, the ITR and ETR use p and g to calculate a public key by "mixing" p and g with their corresponding secret (i or e). When resolving the mapping for a destination EID, the ITR calculates a public key I and sends the calculated public key I to the ETR in the Map-Request packet. The ETR also calculates a public key E, which is sent to the ITR in the Map-Reply packet as a response to its original Map-Request. The ITR and ETR take each other's public key and "mix" it with their own private key to locally generate the shared secret. This shared secret could be directly used as a crypto key. In LISP cryptography, the shared secret is used as input into a cypher that generates a key of a specific width and also a message authentication code. These keys are used to guarantee privacy, integrity, and authenticity for the data traffic that LISP will handle.

Coupling the Diffie-Hellman exchange with LISP provides a series of benefits. Not only are the calculated keys unidirectional, but different keys are calculated for different ITR-ETR pairs. We can refer to this property as pairwise keys; its granularity is in

https://t.me/learningnets
Technet24

contrast with other prevailing shared key mechanisms and is critical to providing successful data privacy. Because LISP operates on demand, similar to how Map-Caches scale, these pairwise keys are calculated and instantiated only when active conversations are taking place between the ITR-ETR pair. The result is a scalable pairwise cryptography system.

In the current definition of LISP cryptography, a given ITR-ETR pair calculates a key to be used to encrypt all traffic between the xTRs. So, the same key is used across different Instance-IDs when multiple virtual networks are instantiated in the xTRs. In certain environments, you might want to provide different keys for different Instance-IDs. This extension is easily achievable in LISP while still maintaining the pairwise behavior described. There is an ongoing debate as to whether such level of granularity adds any security benefit. It is believed that it does not bring added security, but for compliance reasons some networks may require that different key material be generated for different tenants.

HOW THE LISP CONTROL PLANE IS SECURED

The LISP specification, documented in RFC6830bis and RFC6833bis, includes basic security mechanisms for the control plane. The base mechanisms are designed to prevent rogue unauthorized ETRs from registering mappings into the Mapping Database System and to protect ITRs from receiving unsolicited mapping information.

To authenticate EID-to-RLOC mapping registrations and ensure that they are from an authorized ETR, LISP uses shared secret keys between ETRs and the Mapping Database System. Only ETRs that have the shared secret key are able to register EID-to-RLOC mappings to the Mapping Database System. Without the correct key, the authenticity of the Map-Register message cannot be verified, and the Mapping Database System must reject the Map-Register. The shared keys used to authenticate Map-Registers are distributed across ETRs and MS/MRs by configuration; no automated key exchange mechanism is leveraged.

In addition to authenticating EID registrations, it is recommended that the Mapping Database System restricts EID registrations to configured EID prefix ranges. Thus, an authorized ETR is allowed to register EID prefixes only within the EID prefix range configured in the Map-Server. An implementation could even restrict the registration of specific EID prefixes by specific ETRs to prevent malicious ETRs from claiming authority for EID prefixes they shouldn't. Although this is possible and could be seen as more secure, it requires the Mapping Database System to be statically configured with specific EID prefixes for specific ETRs. Any additions or changes in EID location would

https://t.me/learningnets

require manual intervention to reconfigure the Mapping Database System, making the system extremely static.

An alternative approach to prevent spoofing of EIDs from ETRs is to harden the authentication mechanisms so that the ETRs can indeed be trusted to dynamically register EIDs. Not only registrations, but all mechanisms of the control plane benefit from more sophisticated authorization mechanisms. The LISP specification includes, as part of its control plane messages, variable size fields for authentication data to enhance the authentication of the control plane. The architecture is modular so that future improvements in the field of authentication and cryptography can be integrated into LISP. This allows the mechanisms for authentication to evolve independently from the LISP protocol. As of this writing, a suite of mechanisms to enhance the security of the control plane is documented in draft-ietf-lisp-sec as well as in draft-ietf-lisp-ecdsa-auth. Before we get into the details of enhanced control plane authentication, the last basic security mechanism included in the base LISP specification is the use of nonce keys both in the control plane and in the data plane.

ITRs should accept only Map-Replies that it solicited. When an ITR sends a Map-Request, it includes a nonce key with this Map-Request. The Map-Reply must include a matching nonce key for the ITR to verify that this is a response to one of its queries. If the ITR cannot verify that this is a conversation it initiated, it ignores and drops the Map-Reply. This mechanism is geared toward guaranteeing that a malicious operator does not try to insert unsolicited mappings to steer traffic its way or try to run a DoS attack by exhausting the Map-Caches of the ITRs. To prevent man-in-the-middle (MITM) attacks, the nonce mechanism is supplemented with the mechanisms described later as part of LISP-SECurity (LISP-SEC).

The nonce field in the LISP control plane is a 64-bit field and is present in every LISP control plane message. Nonce values are used for different purposes during different phases of the LISP message exchange. ITRs use this field to identify unsolicited Map-Replies, and ETRs use the control plane nonce to track acknowledgments for different EID registrations issued. In the LISP data plane, 24-bit nonce values are present to enable verification of ETR reachability by a process known as *echo nonce*. The role of the data plane nonce is not security related; the echo nonce process was discussed in previous chapters and is defined in detail in RFC6830bis.

Enhanced Control Plane Security

The LISP protocol is designed to be enhanced with additional security and authentication mechanisms. In the following sections, we discuss some of the currently

https://t.me/learningnets

Technet24

defined security mechanisms that harden the integrity of the LISP control plane.

LISP-SEC

LISP-SEC defines a set of security mechanisms to provide origin authentication, integrity, and antireplay protection to the EID-to-RLOC mapping data conveyed in a Map-Reply. LISP-SEC also enables verification of authorization on EID-prefix claims in Map-Reply messages. It does this by calculating a series of hashes along the path of the map-resolution process.

The principle behind LISP-SEC is simple: a one-time-key (OTK) is originated at the ITR, is sent with the requests for mapping resolution, and is the seed material for the generation of a series of keyed hash message authentication codes (HMAC) on the data at different stages of the map-resolution process. The hash material comes back to the ITR as the mapping resolution completes and is verified by the ITR.

The OTK from the ITR and its derivate keys are used along the different steps of the map-resolution process to authenticate the contents of the control plane messages and assure their authenticity and integrity. The mappings conveyed in the Map-Reply are hashed at the Map-Server, and then the Map-Reply is hashed at the ETR. The hash uses the OTK originated at the ITR and OTKs derived from the original OTK to perform the hashes. Because the ITR originates the OTK, it verifies the hashes after it receives the corresponding Map-Reply.

By calculating multiple hashes along the map-resolution path, LISP-SEC includes the logic necessary to assure that the EID scope of the replies issued by an ETR are in line with what the system administrator configured in the Mapping Database System. This mechanism prevents what is known as a *prefix overclaim attack*, in which a Map-Reply includes a mapping to a broader EID prefix than the one configured in the Mapping Database System. This type of attack attracts unauthorized traffic to the ETR issuing the offending Map-Reply.

The OTK mechanism is used with the nonce mechanisms in LISP-SEC to harden the authentication and integrity of the map-resolution process by signing the Map-Replies and their content with keyed-hash message authentication codes. The OTK and the nonce are originated at the ITR. The nonce is transported in all the map-resolution messages until it comes back to the ITR. The OTK is used to seed the generation of HMACs along the path of the Map-Request to eventually sign the Map-Reply and its contents so that a signed Map-Reply can make its way back to the ITR, where authenticity and integrity can be verified. A simplified view of the LISP-SEC process is depicted in Figure 6-12.

https://t.me/learningnets

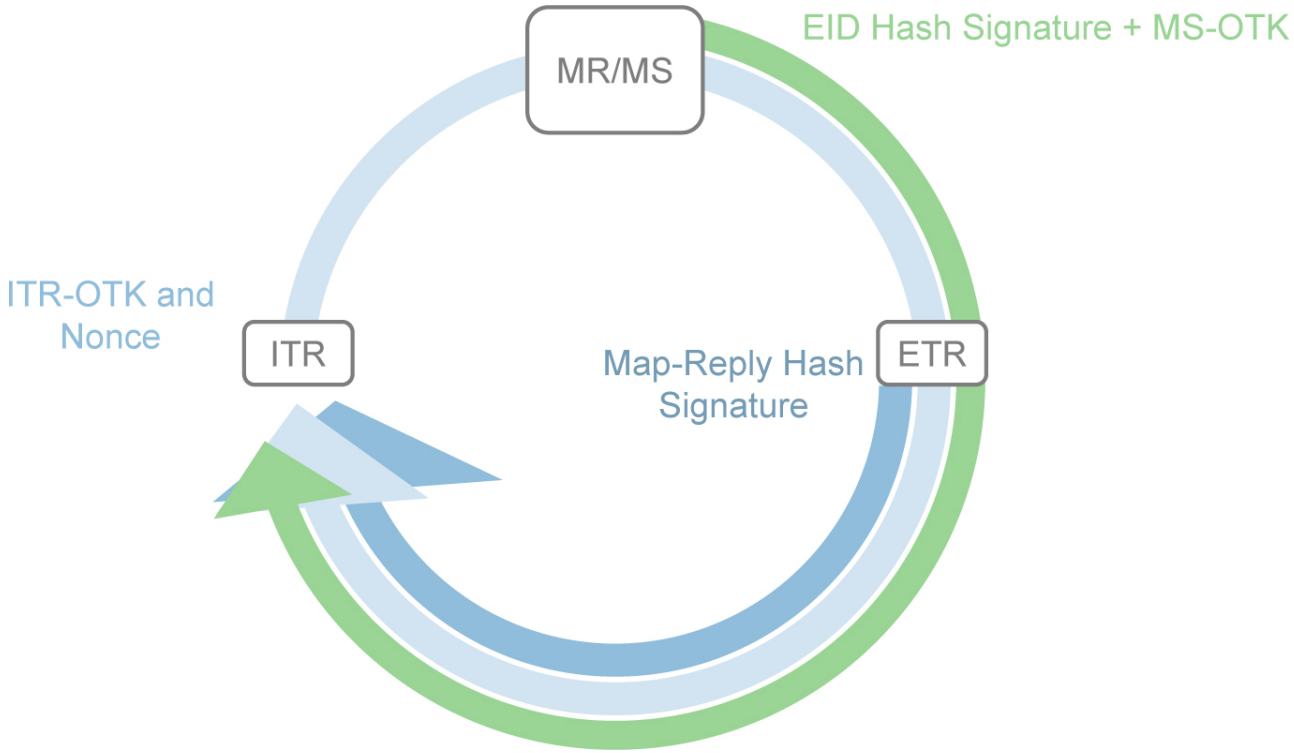


Figure 6-12 *A High-Level View of LISP-SEC*

The ITR generates the OTK, referred to as the ITR-OTK, and sends it to the Mapping Database System (MS). The ITR also generates and sends a nonce.

The Mapping Database System does a lookup among its configured EIDs and signs the longest prefix EID match with the ITR-OTK. The signature is forwarded to the ITR via the ETR piggybacking on existing Map-Request/Map-Reply messages. The EID-prefix signature is verified only by the ITR or the Mapping Database System, which are the only devices that have the ITR-OTK. This signature is meant to prevent the ETRs from claiming authority over broader EID prefixes than the longest prefix match in the Mapping Database System.

The Mapping Database System generates an MS-OTK derived from the ITR-OTK using a key derivation function. The MS-OTK is given to the ETR so that the ETR can sign the Map-Reply to be sent to the ITR with an HMAC. Sending this signature from the ETR allows the ITR to verify the authenticity and origin of the Map-Reply.

The transmission of the OTKs is secured by encrypting the control plane messages using the shared keys configured on the xTRs and Map-Servers.

LISP-SEC is described exhaustively in the IETF Working Group document draft-ietf-lisp-sec. Figure 6-13 illustrates in more detail the heuristic proposed in draft-ietf-lisp-sec to secure the map-resolution message flow. The diagram is included as an aid to help you read through the specification if it is of interest. In this rather elegant solution, authentication data is hashed along the way and ultimately verified when it completes

https://t.me/learningnets
Technet24

its loop through the Mapping Database System at the ITR where the request was originated.

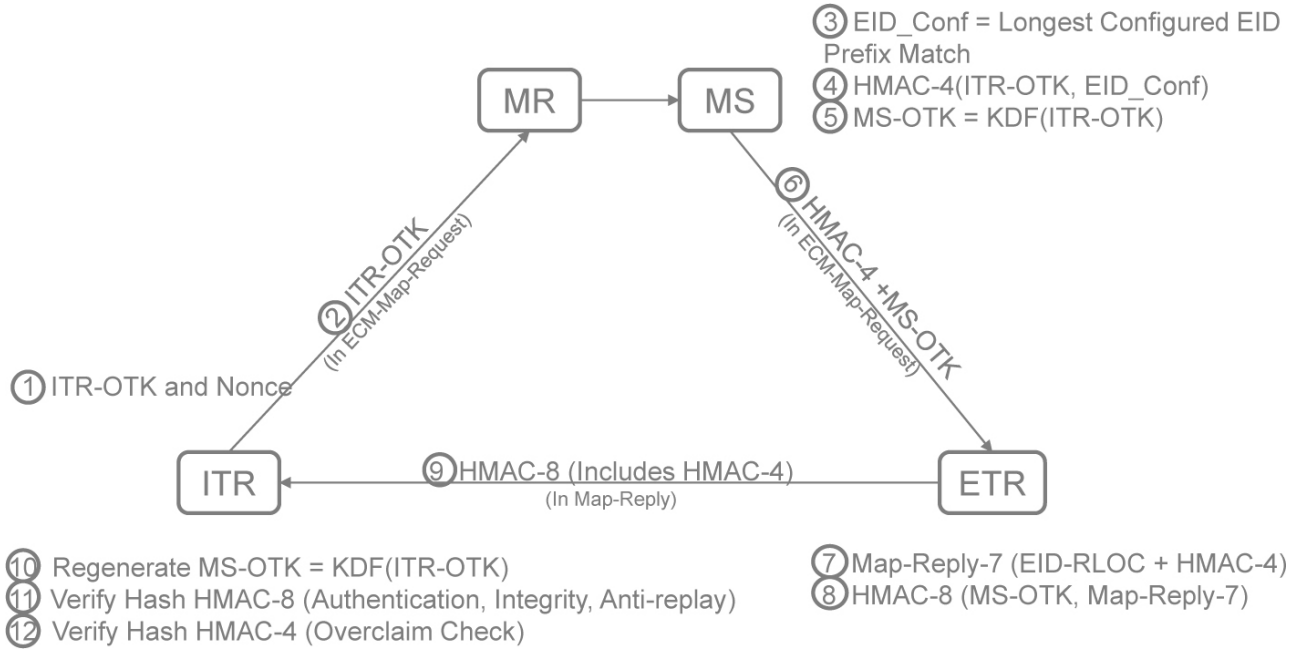


Figure 6-13 *A Detailed View of LISP-SEC*

Threats Addressed by LISP-SEC

LISP-SEC thus provides the necessary mechanisms to secure the Map-Request and Map-Reply messages of the map-resolution process in the LISP control plane. You can find a complete analysis of the threats that may concern anyone implementing or deploying LISP in RFC7835. The following are just some of the security threats that LISP-SEC can mitigate:

• **Origin authentication and integrity:** The origin and integrity of authorized EID-prefixes are preserved by signing the longest prefix EID match with the ITR-OTK at the Map-Server. The origin of the Map-Reply messages is authenticated by signing the Map-Replies with the MS-OTK at the ETRs. The signatures used are hashed message authentication codes in both cases, which ensure the integrity of the messages. One common attack type that is prevented by the HMAC-based authentication of the origin and the messages is a man-in-the-middle attack.

• **Antireplay protection:** If attackers capture a valid Map-Request or Map-Reply, they may replay it with malicious intent. The ITR-OTK and nonce pair for a particular Map-Request are removed from the ITR after the original Map-Reply is received, rendering the replay of a Map-Reply ineffective. There is no benefit to attackers in replaying a Map-Request because doing so simply triggers a new LISP-SEC computation.

https://t.me/learningnets

• **Denial of service (DoS) and distributed denial of service (DDoS) attacks:** LISP-SEC mitigates DoS and DDoS attacks by preventing attackers from forging Map-Request/Map-Reply messages or overclaiming EID-prefixes in an attempt to redirect traffic to a potentially large number of hosts.

• **EID-prefix overclaim protection:** When an attacker takes control of an authorized ETR, the attacker could try to issue Map-Replies that cover an EID prefix that is much larger than the EID prefix authorized by configuration on the Map-Server. This *overclaim attack* gives the attacker control over prefixes that it is not authorized for. As discussed, LISP-SEC imposes a digital signature using an HMAC on the most specific EID configured in the Mapping Database System. When the attacker attempts to overclaim, the HMAC signature fails to be verified, effectively preventing such an attack.

LISP Elliptic Curve Digital Signature Algorithm (ECDSA) Authentication and Authorization

ECDSA may be incorporated into LISP to provide robust digital signatures for Map-Request and Map-Register messages. This functionality is complementary to the signatures provided for Map-Replies and EID-prefixes by LISP-SEC, and it is also complementary to the shared-secret signatures used to secure LISP Map-Registers.

ECDSA is a variant of the asymmetric cryptography DSA algorithm that uses elliptic curve (EC) cryptography. EC cryptography allows the use of much smaller public keys in DSA. For instance, the size of the public key in ECDSA to achieve an 80-bit security level is 160 bits, while standard DSA required a 1024-bit public key. There are no improvements in the size of the signature with ECDSA.

It is proposed that ECDSA be leveraged in LISP to sign Map-Registers and Map-Requests. The public and private key pairs required in ECDSA can be generated on the xTRs or a different external entity and then written to the xTRs. The private key is kept at the xTRs, and a hash is generated using the public key and the <IID, EID> tuple that identifies the xTR holding the private key. This associates the hash to both the public key and also to the signing xTR. The public keys are registered in the LISP Mapping Database System with their corresponding hash as the lookup key. This makes the public keys available through a lookup in the LISP Mapping Database System. When messages are sent to the Mapping Database System, the first check is to see if the hash received in the message matches any of the hashes registered. If there isn't a match, the control plane message is rejected. If there is a matching hash, the public key is determined and used to verify the signature in the message. If the signatures cannot be verified, the message is rejected.

https://t.me/learningnets

Technet24

ITRs and ETRs use the private keys corresponding to the public keys registered with the Mapping Database System to sign their Map-Requests and Map-Registers, respectively. The hash produced with the public key and the xTR <IID, EID> tuple is also included in the Map-Request and Map-Register messages. The LISP Map-Servers maintain the mappings of hash to public keys so that public keys can be obtained from the hash included in the Map-Requests and Map-Registers. When the public key is retrieved, the Map-Server verifies the signatures received in the control plane messages. If the signature cannot be validated, the messages are rejected.

Each xTR is identified by an IPv6 EID. The EID of the xTR is combined with the hash to form an IPv6 crypto-EID by repurposing some of the EID low-order bits to include the hash. You can think of this IPv6 crypto-EID as the signer-EID. The signer-EID is sent in the control plane messages so that the hash can be extracted from it at the Map-Server. Thus, both a signature and the xTR/public-key hash are sent along with the control plane messages. The hash is referred to as the hash-EID in the IETF specification and is an SHA256 hash of the <iid><prefix><public-key> tuple. The signature is the result of applying an elliptic curve mathematical process to a combination of parameters that are relevant to the origin and contents of the Map-Request or Map-Register messages. On reception of a signed Map-Register or Map-Request, the Map-Server extracts and reconstructs the hash-EID from the crypto-EID in the message. The hash-EID is then used to resolve the public key for the crypto-EID and verify the signature. Figure 6-14 illustrates this integration of asymmetric cryptography into the LISP control plane and Mapping Database System.

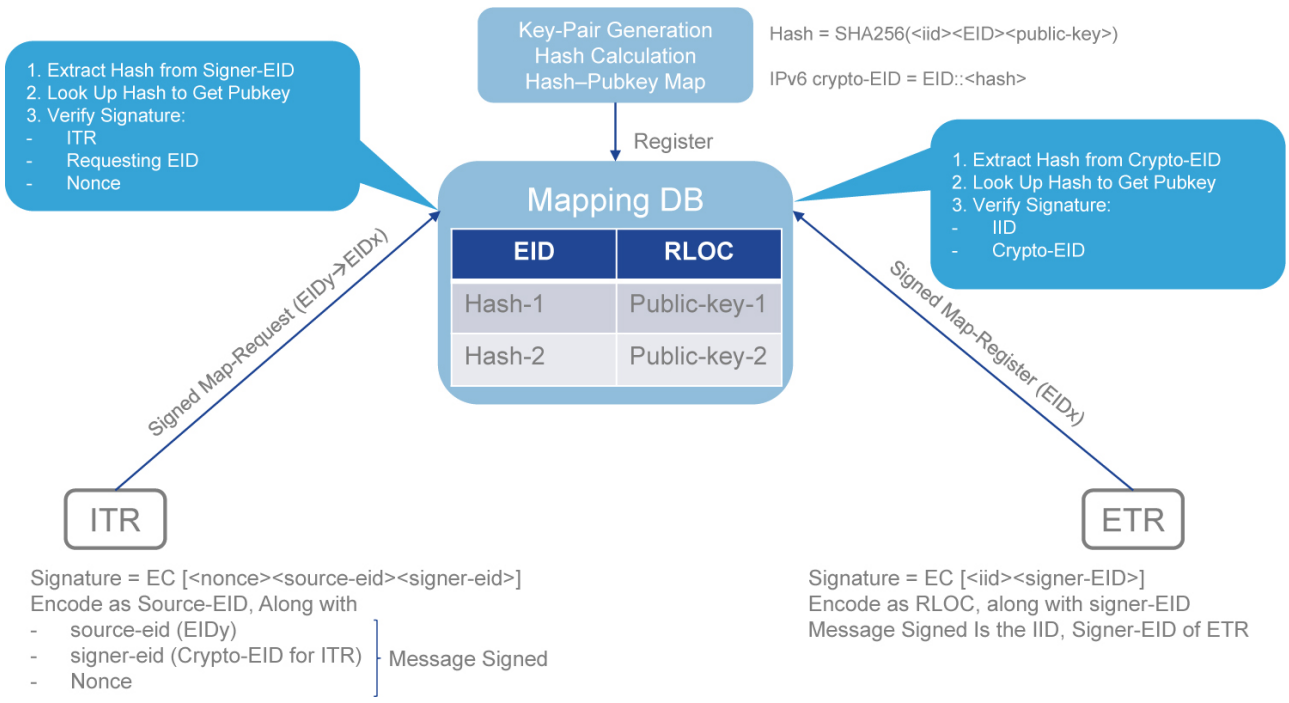


Figure 6-14 *LISP ECDSA Authentication and Authorization*

https://t.me/learningnets

Map-Register messages are signed at the ETR using the ETR's <iid> and <signer-EID> tuple as signature data. When the Map-Server receives a Map-Register, it extracts the hash-EID from the signer-EID included in the Map-Register. It then looks up the resulting hash-EID in the Mapping Database System to determine the value of the corresponding public key. If the hash-EID is not registered with the Mapping Database System, the Map-Register is not accepted. If the hash-EID is registered, the public key is retrieved. With the public key, the IID, and the signer-EID, the Map-Server uses elliptic curve methods to verify the signature received. If the signature cannot be verified, the Map-Register is not accepted.

Map-Request messages are signed at the ITR using the <nonce>, <source-eid>, and <signer-eid> tuple as signature data. The source-eid is the sending EID that triggered the Map-Request. The signer-EID is a crypto-EID that combines the EID of the ITR (the signer) with the hash-EID. When the Map-Server receives a Map-Request, the EID that triggered the Map-Request is the one that is signed. The signature for this signer-EID is verified in the Map-Server by extracting the EID-hash from the signer-EID, doing a lookup for its public key, and verifying the signature. If the EID-hash is not found in the mapping database or the signature cannot be validated, a Negative Map-Reply is sent indicating that the authentication failed.

ANONYMITY IN LISP

LISP allows endpoints to potentially keep their IP address fixed as they move between locations. This means that endpoints can be uniquely identified by their IP address and that any traffic originating or destined to a particular IP address can be unequivocally traced back to a specific endpoint, which in many cases leads to the identification of a particular user. Thus, the property of endpoint address preservation creates a privacy concern as IP addresses map directly to certain user or application identities. Attackers looking at compromising specific users can exploit this information.

One approach to providing anonymity is to support ephemeral EIDs. This approach is similar to a credit card provider giving you a temporary credit card number to complete online transactions and then deprecating the number. The hosts connecting to the LISP network instantiate ephemeral EIDs and register them in LISP in the same way they do regular EIDs. For LISP to be able to register these ephemeral EIDs, they should belong to known address blocks so that the Mapping Database System can be configured to accept their registration.

The use of ephemeral EIDs and the assignment of temporary credit card numbers illustrate the fundamental principle of separation of identifiers from the actual identity

https://t.me/learningnets

of the user or endpoint. The temporary addresses or credit card numbers are identifiers for the more permanent resources, which have an immutable identity. The identity is important to protect. Dynamically mapping an identity to mutable identifiers for reachability and other purposes allows the identity to operate somewhat anonymously. Now, this correlation between identity and identifier is basically a mapping operation. This type of correlation is what LISP was designed to do. At the time of writing, the specifics on how the mechanisms for identity and identifier separation are to be standardized are being actively discussed. LISP is front and center in such discussions.

SUMMARY

LISP enables network-based security mechanisms that are necessary in modern networks. It delivers these mechanisms in modern, elegant, and efficient ways. The efficiencies achieved in LISP are unprecedented and set LISP apart from more traditional routing and security protocols.

Network segmentation is instrumental in limiting the spread of malware in the network. Segmentation at the virtual network, endpoint group, and workload levels is delivered seamlessly and with unprecedented scalability in LISP, allowing common primitives and operations in the control plane to deliver segmentation at different levels of granularity. Traffic steering services in LISP complement the network segmentation services by allowing the dynamic insertion of stateful enforcement devices where the policy may require them.

Data privacy, integrity, and origin authentication are paramount security requirements for businesses. LISP provides an integrated cryptography solution that leverages the LISP demand-based model to deliver pairwise cryptography at unprecedented scale with peer-to-peer public-key exchanges and no dependency on central key infrastructure.

The demand-based model used in LISP authenticates and assures the integrity of data connections while securing the control plane. A thorough set of security mechanisms is incorporated in the LISP control plane so that all messages and resolution processes in the protocol are digitally signed and validated using a well-thought-out combination of shared keys, asymmetric keys, and one-time keys.

LISP offers all this security in a seamless manner that does not add operational cost or limit the scalability of the system.

https://t.me/learningnets

https://avxhm.se/blogs/hill0

Chapter 7. LISP and the Next-Generation Mobile Network

The endpoint densities, rates of mobility, network redundancy, and path requirements being driven by next-generation applications pose demanding requirements on the network. These requirements go beyond what can be addressed by simply optimizing the existing incumbent networking models. A shift toward overlays and overlay-optimized demand control planes is necessary to satisfy this next wave of requirements. This chapter discusses how LISP supports mobility and how these mechanisms adapt to different use cases that should illustrate the high bar that was set for the network to surmount.

LISP EID MOBILITY AND LISP MOBILE NODE

LISP supports two mobility models. One model is focused on supporting mobility of hosts around a fixed LISP infrastructure. The other model allows elements of the LISP architecture (tunnel routers, or xTRs) to be mobile.

LISP endpoint identifier (EID) mobility allows a host to be mobile and preserve its EID IP address as it moves around a LISP network. As EIDs move, they attach to different xTRs that are fixed across the network. The EIDs are mapped and remapped to the different routing locator (RLOC) sets of the xTRs they connect to as the EIDs move. The mechanisms by which LISP supports EID mobility are defined in draft-ietf-lisp-eid-mobility.

With the LISP mobile node, LISP also supports the capability for an entire LISP site to move. A LISP site consists of one or more xTRs and a series of EIDs connected to those xTRs. The entire LISP site could be implemented on a mobile device such as a smartphone or be the combination of endpoints and xTRs on a moving platform such as a ship, plane, or satellite. As the site moves, the xTR obtains new IP addresses for its RLOCs. LISP updates the mappings of the EIDs of the mobile site to the new RLOC addresses assigned to the mobile xTR as the site moves. This functionality, referred to as *LISP mobile node*, is defined in draft-ietf-lisp-mobility.

https://t.me/learningnets

Technet24

From the point of view of the xTRs in the LISP site, these two mobility models are seen as methods to handle changes in the EIDs or changes in the RLOCs. In the EID mobility case, the RLOCs for the xTR are fixed, and the EIDs that attach to the xTR change. In the mobile node case, the EIDs are fixed, and the RLOCs for the xTR change as the mobile node moves.

LISP EID Mobility

Moving hosts about the network while maintaining a constant IP address and preserving any connections they have is a common requirement in data centers where host and workload mobility is leveraged to fulfill a variety of operational tasks. Seamless endpoint mobility is also a growing requirement in the access network where wireless connectivity and portable endpoint form factors require that connectivity be preserved without interruption as the devices move. Uninterrupted communication and mobility became a foundational requirement for any network and are assumed to be part of the baseline of connectivity today.

The LISP EID mobility solution allows hosts to move about a network without requiring any LISP software to run on the host. LISP EID mobility provides a highly optimized solution to the endpoint mobility problem and provides consistent shortest path reachability to the mobile endpoints, fast convergence, and unprecedented scale.

Endpoints move around a LISP network and attach to different xTRs while keeping their IP address unchanged. Leveraging the simple principle of locator/identifier separation, the LISP network abstracts the entropy and variability introduced by the moves of the EIDs (identifiers) from the RLOC routing space. Because the RLOC routing space does not see any of the moves in the EID space, you can design it for stability, robustness, and resiliency. Endpoints simply attach to different xTRs at the periphery of the network, the xTRs detect the endpoints, and they map the detected EIDs to the RLOCs of the xTRs where they are detected. The role of LISP is simply to manage the constant updates of the EID-to-RLOC mappings and Map-Caches as the endpoints move between different RLOCs. With the mappings and Map-Caches updated, LISP ITRs encapsulate traffic directly to the optimal ETR, as they normally would do. Figure 7-1 illustrates a LISP-enabled network with endpoints moving between xTRs.

https://t.me/learningnets

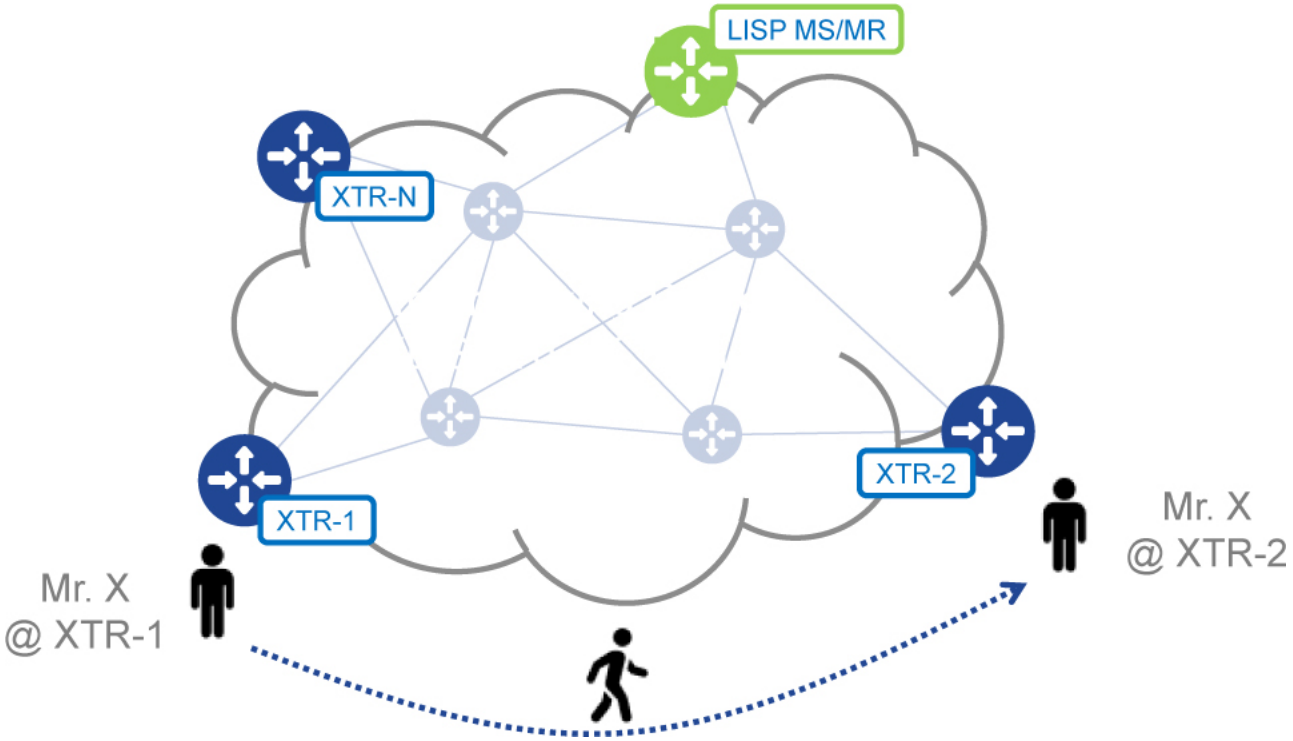


Figure 7-1 *LISP EID Mobility*

The endpoints do not need to run any LISP components or any client software to benefit from the mobility that a LISP network provides. Additionally, the LISP EID mobility solution delivers shortest path reachability to the roaming endpoints and does not rely on anchor points that would result in suboptimal paths to and from the roaming endpoints. This clientless and path-optimized mobility solution is designed to support high rates of mobility for a large density of endpoints. The key to the scale and fast convergence of the LISP EID mobility solution is the demand-based nature of LISP. Any protocol updates triggered by mobility events are scoped only to those xTRs that are involved in communications that include the roaming endpoint. The LISP EID-mobility solution limits significantly the amount of messaging required and its reach, which in turn reduces the processing required to complete a move. Mobility is supported in networks with endpoint densities that are orders of magnitude higher than what access networks have traditionally supported, and the endpoints may move at unprecedented rates and enjoy virtually uninterrupted traffic forwarding as they move. These characteristics of high-density and high-frequency mobility are desirable in IoT environments where robots, sensors, and other connected devices depend on being reliably and uninterruptedly connected to a central controller to operate properly. These requirements are also critical in the transportation industry and are a cornerstone to the 5G network design discussion. These use cases for LISP EID mobility are discussed in more detail later in this chapter.

LISP EID Mobility Mechanics

LISP EID mobility performs three fundamental tasks:

https://t.me/learningnets

Technet24

• Detection of the moving host

• Mapping updates

• Map-Cache updates

Because the solution is clientless, LISP xTRs that support EID mobility must implement mechanisms to detect host moves based on the standard behavior and traffic of the hosts. LISP xTRs configured to support EID mobility monitor the source of any L2, IP, or ARP traffic they receive from a host. This way, the xTRs know which EID addresses are connected to it and if there are any new EID addresses for which it must update the Mapping Database System and Map-Caches in the overlay. If the host was not previously known to be connected on the port where the traffic is received, the LISP mobility convergence process is started. Figure 7-2 illustrates this convergence process.

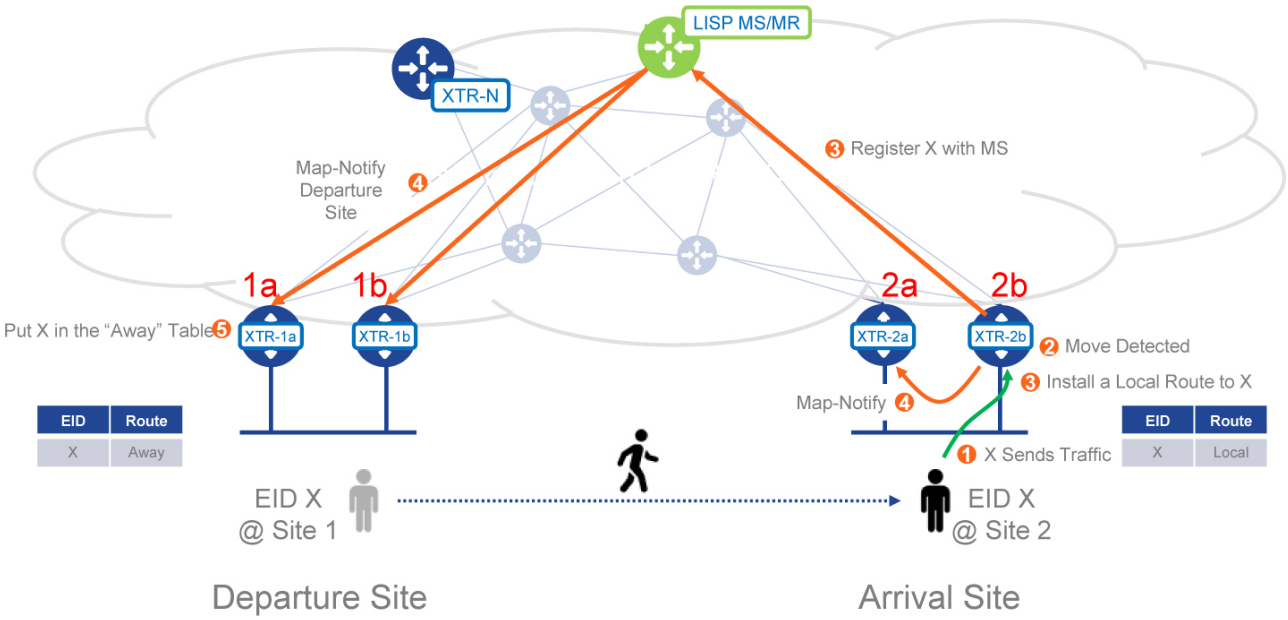


Figure 7-2 *LISP EID Mobility Convergence*

Figure 7-2 shows EID X moving from site 1 to site 2. For the purposes of this explanation, we refer to site 1 as the departure site and site 2 as the arrival site; the xTRs at each site are the departure xTRs and arrival xTRs, respectively. Here, we refer to EID X as the roaming EID. Figure 7-2 illustrates EID X roaming from site 1 to site 2. At the arrival site, one of the xTRs is the first to receive traffic from the roaming host and detect the roaming EID. The arrival xTR is configured as the default gateway for the mobility prefixes. After receiving traffic, the arrival xTR compares the source IP address of the traffic received with the IP prefix configured for mobility. If there is a match, the arrival xTR installs a local host route to the detected EID on its local interfaces and also registers the detected EID with the Mapping Database System using its own RLOCs for the mapping. By registering the EID, the arrival xTR updates the

https://t.me/learningnets

location of the roaming EID and sets it to itself. When the xTR receives and de-encapsulates traffic destined to the roaming EID, it uses the local host route it installed when the EID was detected to send the traffic onto the roaming EID.

In many cases, the system has redundant xTRs connecting the L2 segment where the EID lands to the LISP overlay. If more than one xTR is servicing the L2 segment where the EID lands, the detecting xTR sends a Map-Notify message in the L2 segment to notify other xTRs servicing the L2 segment of the presence of the detected EID so that they can install the appropriate host routes and initiate the necessary registrations.

> Note
>
> *The Map-Notify messages include the RLOCs that the EID is to be registered with or a site-id. When an xTR receives the Map-Notify message, it checks the RLOCs or site-id in the Map-Notify. If the RLOCs/site-id in the Map-Notify corresponds to the RLOCs/site-id local to the xTR, the xTR receiving the Map-Notify installs a local host route for the detected EID and also registers the EID with the Mapping Database System. If the RLOCs/site-id in the Map-Notify does not correspond to the RLOCs/site-id local to the xTR receiving the Map-Notify, the xTR includes the detected EID in its away tables. Map-Notify messages are sent as link-local multicast messages so they are replicated to every xTR in the L2 segment. If the L2 segment is stretched across multiple sites, the Map-Notify reaches xTRs in the different sites. Comparing the RLOCs/site-id in the Map-Notify with the RLOCs/site-id in the Map-Notify allows each xTR to treat the notification as one for a new locally connected EID or as a notification stating that the EID is now "away" at a remote site*.

When the EID is registered with the Mapping Database System, the Map-Server sends a Map-Notify to the departure xTRs (the xTRs that registered the EID before the move). This Map-Notify message tells the xTRs at the departure site that the EID moved "away." Upon receiving this message, the departure xTRs install the roaming EID in their away table and remove any specific local host routes to the roaming EID that were present.

Up to this point, the roaming EID is registered to the Mapping Database System with its new location information and the routing tables on the departure, and arrival xTRs are updated to reflect the location of the EID. However, we have not addressed how the Map-Caches of any ingress tunnel router (ITR) sending traffic to the roaming EID are

https://t.me/learningnets

to be updated.

The Map-Caches on ITRs sending traffic to the roaming EID may be updated through a couple of mechanisms: data-driven solicit map requests (SMRs) or subscription notifications. When using data-driven SMRs, the ITRs continue to send traffic to the departure ETR, and after receiving this traffic, the departure ETR sees that the destination EID address for the traffic sent matches an entry in its away table. Because the destination is "away," the departure ETR issues an SMR to the sending ITR, basically instructing the ITR to refresh its Map-Cache as it is sending traffic to an old location. The ITR then issues a new Map-Request for the roaming EID and receives an updated mapping so that it can start sending the traffic to the arrival xTRs. This scenario is illustrated in Figure 7-3. The mechanism is similar to what happens in the mail system when someone moves: letters are sent back to the sender with a note to update the destination address; after receiving the returned letter, the sender looks up the updated address and resends the letter.

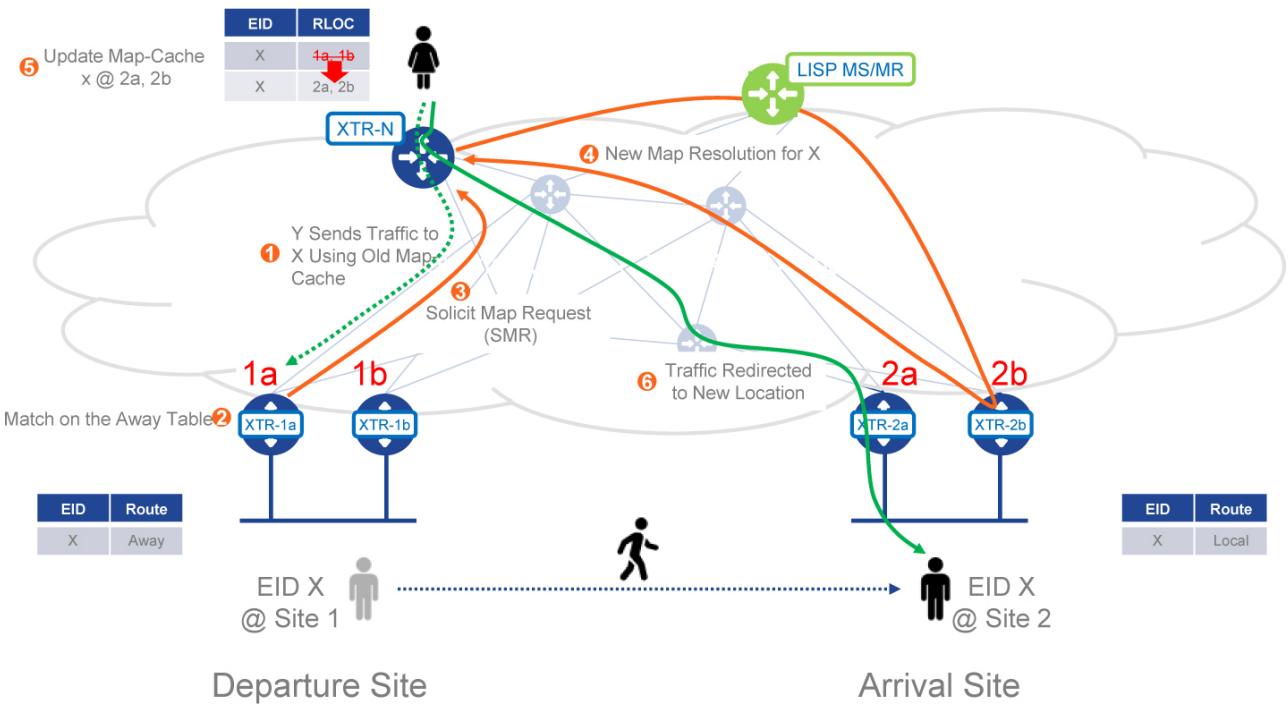


Figure 7-3 *LISP EID Mobility Data-Driven Map-Cache Refresh*

Data-driven SMRs provide the benefit of large scalability; only active Map-Caches are updated and only when required. The drawback may be in convergence time because the process involves waiting for packets to get to the departure xTR, signaling the sender, and finally doing a new lookup. All these steps take time, and during this time, traffic is lost. The process takes milliseconds, but in a world of low latency and high availability, it may be desirable to reduce the packet loss impact of the mobility event. The mechanisms by which convergence may be optimized to reduce packet loss are based on a combination of reduced signaling and traffic redirection. The optimization

https://t.me/learningnets

of the convergence process for mobility is described in detail in “Mobility Convergence Optimization” later in this chapter.

LISP Mobile Node

LISP mobile node gives the mechanisms for an xTR to move around the network. The xTR moves along with any EIDs that may be attached to that xTR, effectively creating a mobile LISP site. One clear benefit of this approach is that the mobile xTR can attach to any network, whether the network is LISP enabled or not, and provide mobility for the EIDs independent of the transport network. Transportation applications such as ships may choose this approach because it also allows the craft to move between transport networks or even multihome to the different networks.

One instantiation of a LISP mobile site could simply be a mobile xTR deployed on board of a moving craft that has a multitude of EIDs that are serviced by the mobile xTR. Another approach is found in mobile devices, such as smartphones, where both the xTR and EIDs are inside the mobile device. In either case, the logic that the xTR must follow to update its RLOCs and the corresponding EID mappings is the same. We refer to the functionality as LISP mobile node and focus on the mobile device scenario for the purposes of this discussion. The concept of a LISP mobile node is illustrated in Figure 7-4.

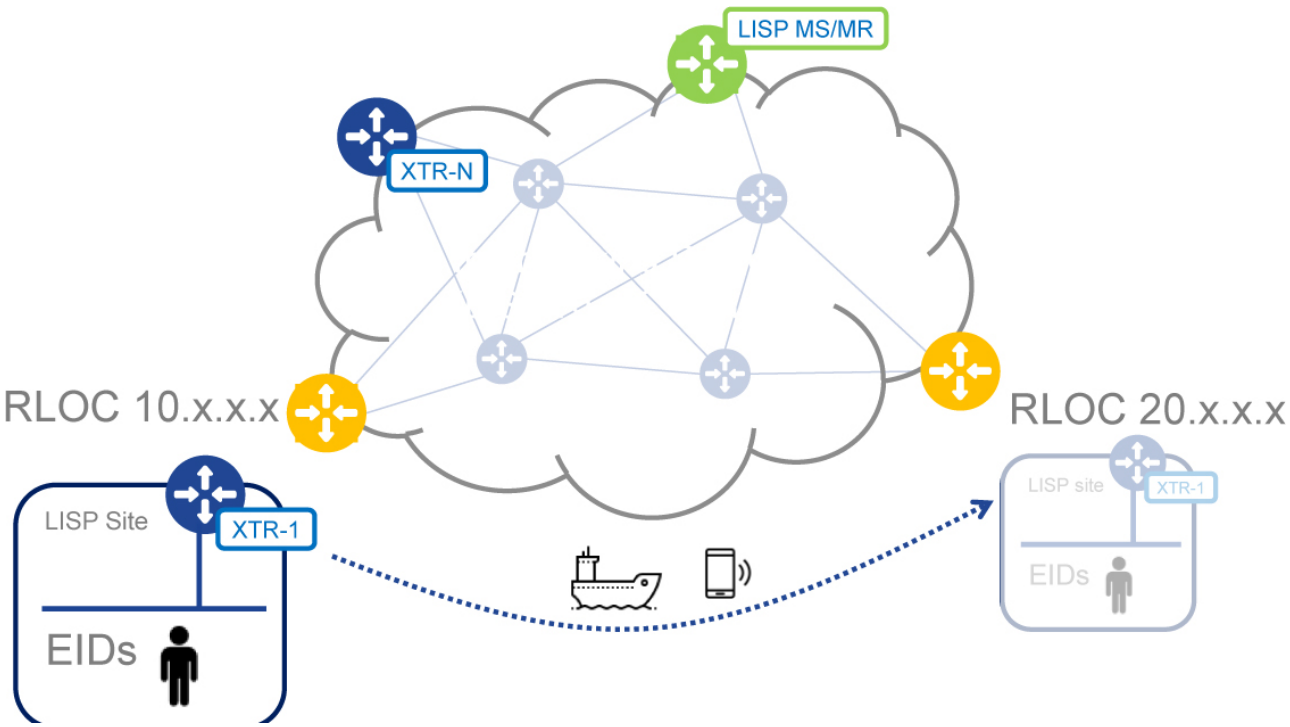


Figure 7-4 *LISP Mobile Node*

The LISP mobile node is an element of the LISP architecture that runs on a mobile device. A lightweight ITR/ETR is implemented on a mobile device to service EIDs that

https://t.me/learningnets

are assigned to the mobile site (in this case, the site is within the device). The objective is to enable application session persistence in the face of mobility by allowing applications running on the mobile device to preserve their EID IP addresses as the mobile device roams from one network location to another and the RLOCs for the mobile node xTR change. The mobile node may also roam between RLOC networks and multihome to several RLOC networks. The lightweight xTR in the LISP mobile node basically masks the mobility event so that the applications running on the mobile device do not perceive any change in the connectivity as their EIDs remain constant.

The mobile node is assigned one or more EID prefixes, and the lightweight xTR on the LISP mobile node registers those EIDs with a Mapping Database System in the mobile network address space. Thus, the LISP mobile node is basically a LISP site within a mobile device. As the mobile node roams, the mobile node gets new addresses on its interfaces; these addresses are the RLOCs to the lightweight xTR in the mobile device. In the mobile node model, the RLOCs of the xTR change and trigger an update of the mapping. In the EID mobility model, the EIDs attached to the xTR change and trigger an update of the mapping.

The mobile device may have multiple antennas or interfaces. This scenario of multiple interfaces is inherently handled by the lightweight xTR as a multihoming scenario like the ones discussed in previous chapters. One clear advantage is that the mobile device can now have multiple interfaces enabled simultaneously and balance the load of its traffic over the different interfaces based on the priority and weight policies in LISP.

Because battery life is critical to mobile devices, the lightweight xTR in the LISP mobile node relies on the Map-Server to reply to any Map-Requests on its behalf. The proxy reply mechanism used by LISP mobile-node reduces the amount of signaling sent to the mobile node, which in turn reduces battery drain. In the same vein, other resources like processor, memory, and bandwidth benefit from offloading the processing and signaling required to reply to Map-Requests.

LISP Mobile Node Mechanics

Just like in EID mobility, the updates in the mappings resulting from the move events must be communicated to the remote ITRs so that they can update their Map-Caches. In this scenario, there isn't a departure xTR and an arrival xTR, so the notion of an away table at the departure xTR does not apply. In the LISP mobile node scenario, an xTR moves from a departure network to an arrival network and gets new RLOC addresses when it moves. So, data-driven SMRs are not an option in this case. The different mechanisms used to update the Map-Cache of the remote ITRs include

https://t.me/learningnets

• **ITRs subscribing to the roaming EID:** This leverages the publish/subscribe functionality described in draft-ietf-lisp-pubsub to make sure a Map-Notify is sent to any ITRs encapsulating traffic to the roaming xTR when the xTR roams. ITRs encapsulating traffic to the roaming xTR are assumed to have subscribed to one or more of the EIDs serviced by the roaming xTR. The subscriber ITRs update their Map-Cache after receiving the Map-Notify message.

• **Setting a small TTL on Map-Replies:** This simply accelerates the time-out of the ITR Map-Caches so that they reissue Map-Requests more frequently and procure updated information for EIDs on a moving xTR within the time window of the time-to-live (TTL).

• **Piggybacking mapping data:** When communication is bidirectional between the mobile node and the remote ITR, the LISP mobile node has the address of the remote ITR as an RLOC in its Map-Cache. The LISP mobile node may include updated mapping information in Map-Requests sent to the RLOCs in its Map-Cache (which are the RLOCs of the remote ITRs) and trigger an update of the Map-Cache of the remote ITRs.

• **Sending SMRs to RLOCs in the Map-Cache:** The ETR on the mobile node may elect to send SMRs to any RLOCs in its cache. This mechanism assumes that communication is bidirectional between the mobile node and the remote ITRs so that the mobile node has the RLOCs of the remote ITRs in its Map-Cache. The SMR message suggests to the remote ITR to issue a Map-Request for all EIDs on the mobile node. Issuing a Map-Request may result in additional state in the remote ITRs because they may have initially requested mappings for only a subset of the EIDs on the mobile node.

• **Map versioning:** Assuming bidirectional communication, a remote xTR receives new map versioning information from a roaming xTR. The new version information indicates to the remote xTR that it must issue a new Map-Request for the EIDs whose version changed.

A few open-source projects implement LISP mobile node. One interesting project is the Open Overlay Router project at https://openoverlayrouter.org. This project includes implementations of LISP mobile node compatible with Android and Apple IOS.

MOBILITY CONVERGENCE OPTIMIZATION

Mobility must be seamless. In networking terms, this means minimal to no packet loss. LISP provides a few interesting mechanisms to minimize the possibility of packet loss

https://t.me/learningnets

due to mobility convergence. In the following sections, we explore some of them. All these optimizations apply to the EID-mobility clientless model. The LISP mobile node model benefits from the publish/subscriber and predictive RLOC methods but not from the redirection method.

Redirection

One thing that you can do in a LISP implementation of EID mobility is to install a redirect mapping at the departure xTR so that it re-encapsulates traffic for the roaming EID after the xTR gets a notification of the EID's move. In this model, the departure xTR installs a LISP mapping in its cache pointing to the RLOCs of the arrival xTRs and re-encapsulates traffic destined to the roaming EID to the arrival xTRs while the mobility convergence completes. The departure xTR builds this re-encapsulation mapping with the information received in the Map-Notify message, or it installs this mapping earlier with a preference that is lower than the local route to the EID so that it is pre-installed and takes effect only if the local route to the EID becomes unreachable. Thus, while the departure xTR issues SMRs and the ITR refreshes its Map-Cache, any traffic for the roaming EID received at the departure xTR can be re-encapsulated to the arrival xTRs and thus avoid packet loss during a portion of the reconvergence window. This proposed redirection is illustrated in Figure 7-5.

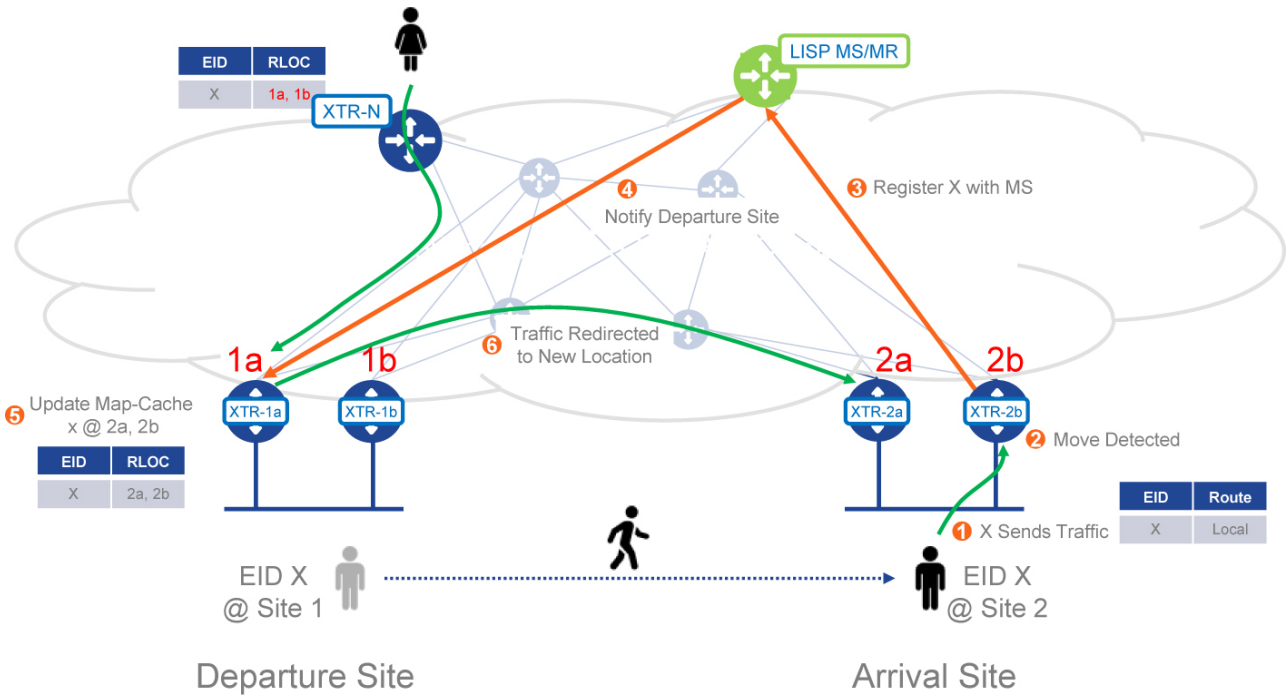


Figure 7-5 *LISP Traffic Redirection to Assist Convergence*

Pub-Sub

Following the same principle of installing a new Map-Cache entry in response to a Map-Notify message, the Map-Notify message could be sent directly to the encapsulating

https://t.me/learningnets

ITR and avoid the redirection as well as the SMR process altogether. This approach comes at the cost of scale because the Mapping Database System must know which ITRs to send the Map-Notify to. To do this, the Mapping Database System needs to maintain a list of ITRs that may have active mappings for the different roaming EIDs. You can think of this as the Mapping Database System maintaining a list of subscribers (ITRs) to the different EIDs. If any change to the mapping information related to the EIDs occurs, the subscriber ITRs must be notified of these changes. The mechanisms by which the Mapping Database System populates and manages this subscription list are known as the LISP publish/subscribe functionality and are defined in draft-ietf-lisp-pubsub. The principle is simple: whenever an ITR issues a Map-Request for a particular EID, it has the option of subscribing to updates on any changes to the mapping information for that EID. When information changes for an EID, the Mapping Database System publishes updates to its subscribers. One mechanism proposed to issue these updates is Map-Notify messages.

In the case of the mobility solution, the ITRs subscribe to the roaming EID when they first issue a Map-Request for the roaming EID. From that point on, the Mapping Database System notifies the subscriber ITR of any changes in the mapping information of the roaming EID. This notification is sent after the Map-Server receives a Map-Register with the new information for the roaming EID. The mechanism of subscription-based mobility updates is effective for both the LISP EID mobility and LISP mobile node models. The flow of signaling is illustrated in Figure 7-6.

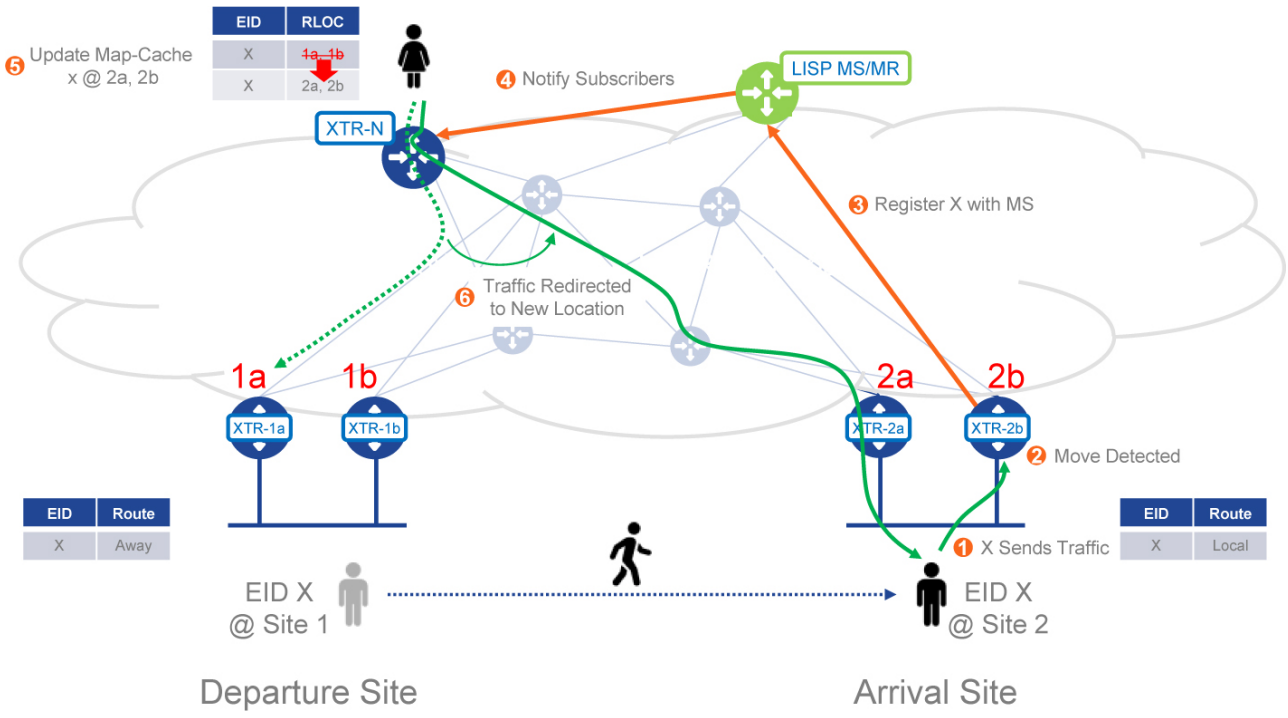


Figure 7-6 *Pub-Sub-Enabled LISP EID Mobility*

The benefit is clearly a reduction in the convergence time required. The cost is the

https://t.me/learningnets

maintenance of the subscription state in the Mapping Database System. Having too many subscribers and a large number of discrete EIDs to subscribe to may represent a challenge to the Mapping Database System. Depending on the traffic patterns, the system may have a relatively small number of subscribers to a large number of roaming EIDs. The subscribers could opt to subscribe to an aggregate prefix that would include updates for a large number of host EIDs in a single subscription. For the right application, the publish/subscriber model may provide a good option with manageable scalability. The obvious degenerate case of this functionality is the scenario in which every xTR subscribes to every EID, basically turning the system into a push system. This misuse of the mechanism is rare, but it is still worth pointing out so that you keep it in mind when designing networks and enabling subscriptions.

The publish/subscriber signaling model for Map-Cache updates in mobility can be combined with redirection from the departure xTR to the arrival xTR to prevent packet loss during convergence. This capability is relevant in the cases in which the EIDs may be connected to multiple sites and the reconvergence event is triggered by a failure rather than an actual move. In this scenario, the redirection mappings are cached at the xTRs ahead of the failure so that they are effective immediately when a failure takes place.

Predictive RLOCs

Predictive RLOCs refer to the mechanisms at play when an ITR predicts where an EID or an xTR would move next. In this scenario, the ITR creates multiple copies of its traffic and encapsulates them to the known current RLOC and also to any potential RLOCs to which its destination could move next. As a host moves from one RLOC to the next, an ITR sends any traffic destined to this host/EID to both the departure RLOC and the arrival RLOC. No matter what stage of the mobility process the system may be in, a copy of the traffic is always available to the roaming device at the xTR it is attached to. Theoretically, the roaming device should not lose any traffic when completing a roam operation. The principle of predictive RLOCs is illustrated in Figure 7-7.

https://t.me/learningnets

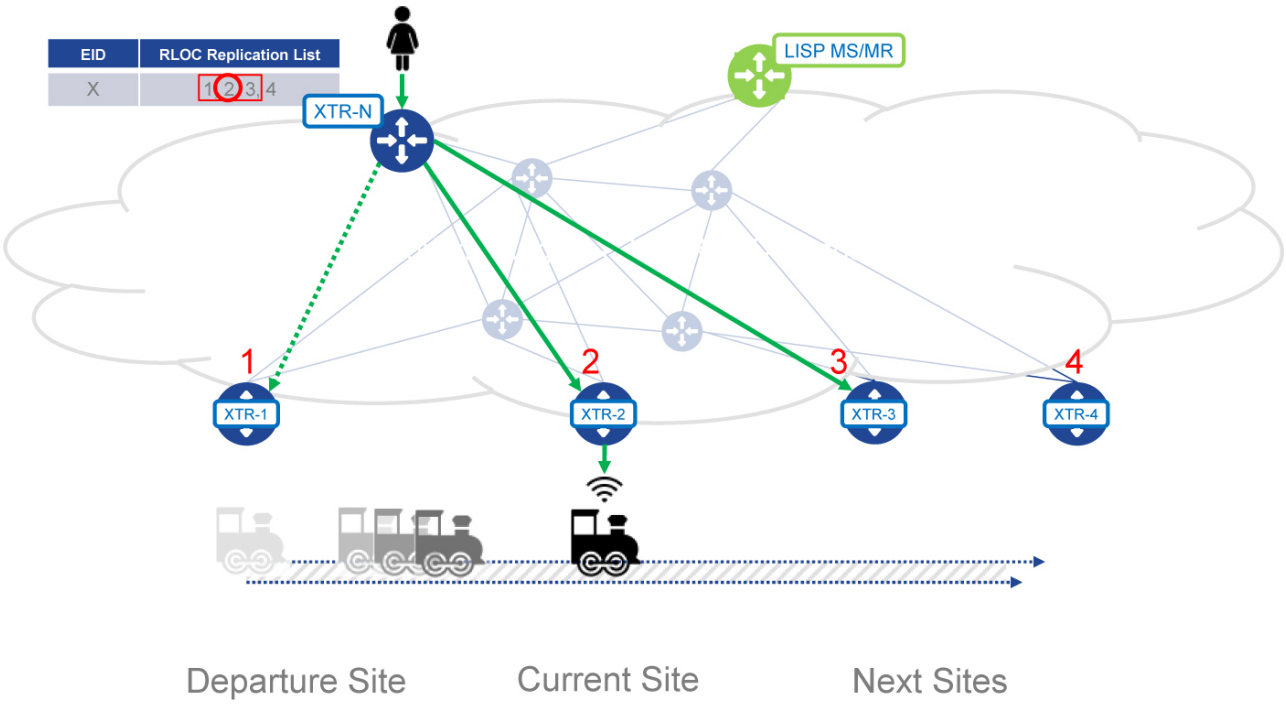


Figure 7-7 *Predictive RLOCs*

In geographically linear systems such as railroads, it is rather straightforward to come up with an ordered list of RLOCs that a moving train attaches to as it travels down a particular line. You can register this list of RLOCs in the LISP Mapping Database System, and the ITRs can leverage this list to determine which RLOCs to replicate the traffic to. The mechanisms to achieve this behavior in LISP are specified in draft-ietf-lisp-predictive-rlocs. The specification also shows how intersections could be modeled as replication lists in the Mapping Database System.

The data structure in the Mapping Database System could also lend itself to modeling a three-dimensional mobility space. Thus, the model of predictive RLOCs can be extended to other less predictable systems. For example, in a Wi-Fi network, the Wi-Fi infrastructure (radio frequency controller) provides predictive information based on the signal strength it measures at different access points. With this information, the Wi-Fi controller may dynamically populate replication lists in the Mapping Database System and also provide a sense of current direction of the moving device. This information can be exchanged with the LISP Mapping Database System through application programming interfaces (APIs) to populate the Mapping Database System directly and quickly with a predictive RLOC list.

To reduce the amount of traffic replicated by an ITR, the ITR may monitor the RLOCs from which it receives return traffic. By doing so, the ITR determines which RLOC in the replication list the moving EID may be connected to and replicate only to that RLOC and the next ones in the replication list. The ITR may also limit how many future ITRs the traffic is replicated to. The flow of information is illustrated in Figure 7-8.

https://t.me/learningnets

Technet24

Additionally, the Mapping Database System can send notifications to the next N RLOCs/ETRs in the replication list so that they install local routes for the moving host ahead of time. The predictive-RLOC mechanism allows the replicated traffic received at the ETRs to be delivered to the EIDs the minute they attach to the ETR, without any convergence delay, thus minimizing the potential for packet loss and providing fast roaming.

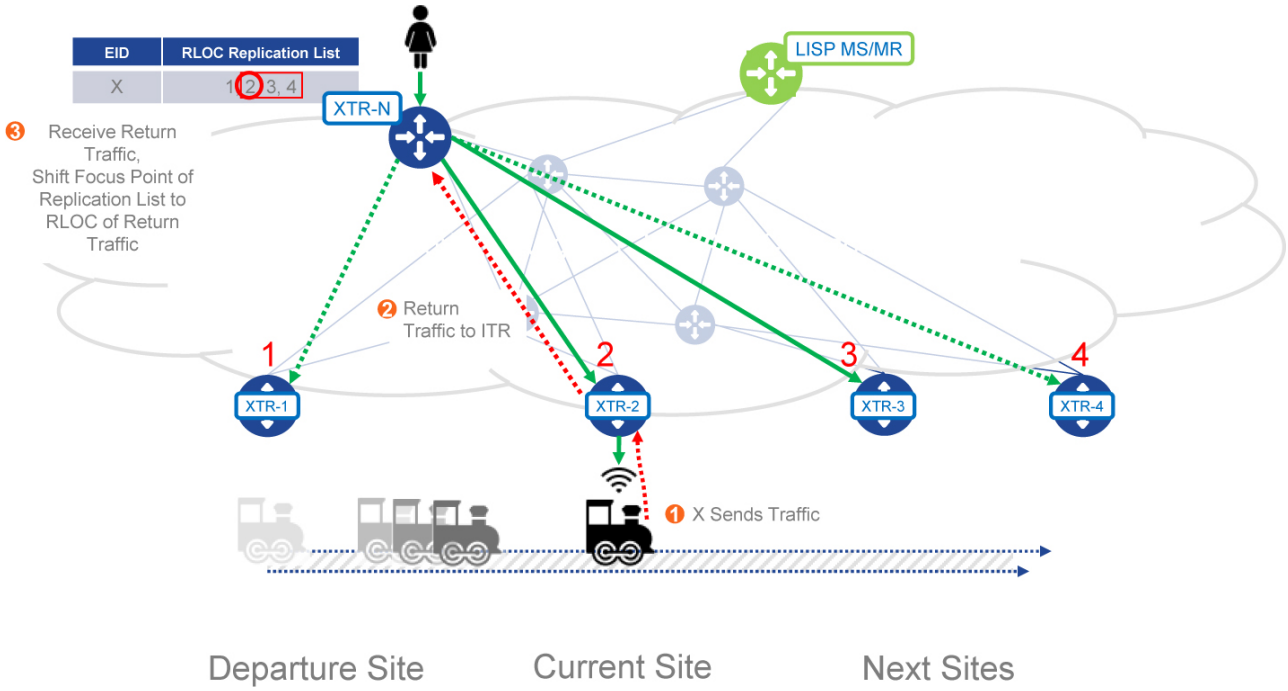


Figure 7-8 *Predictive RLOCs Replication Scoping*

An unlikely yet interesting potential race condition may arise between the mobile EID and the transmission of the replicated information for RLOC predictability. This condition leads to the reception of duplicate packets at the roaming hosts and occurs only under rare network delay conditions. In a scenario where there are significant network delays to some RLOCs but not others, an EID may receive a copy of a packet while connected at xTR-1 and move to the next RLOC (xTR-2) faster than the copy of the same packet can be replicated and transmitted to xTR-2. In this case, the roaming EID receives the packet once at xTR-1 and moves to xTR-2 before the predictive copy of the same packet is received at xTR-2. When xTR-2 receives the copy of the traffic, it forwards it to the roaming EID, causing the roaming EID to effectively receive two copies of the same packet. Although this scenario is interesting, it is unlikely. If you're concerned about this scenario, monitoring the latency levels between RLOCs may provide a good indication of the likelihood of this scenario by comparing measured network latencies to the speed at which an EID may roam. Usually, the speed at which the two types of events occur are orders of magnitude apart, which makes the situation highly unlikely.

https://t.me/learningnets

USE CASES

To this point, we have discussed the mechanisms by which LISP supports mobility in detail. How these mechanisms are used in different applications is rather interesting. The following sections examine a few of the scenarios in which LISP mobility is applied.

Use Case: High Rate Mobility

As the Internet of Things (IoT) increases its foothold, more and more environments must host a very high density of highly mobile devices. It is not clear at this point how mobile IoT devices will be implemented. However, there will be many of them, and they will be multihomed. Although individual devices may move relatively slowly, the convergence of a large number of mobile devices results in a high aggregate rate of mobility events that the network must handle.

One prime example of a densely populated environment with a high aggregate mobility rate is an automated fulfillment center for a large mail-order retailer. In this environment, tens of thousands of robots move around a warehouse and must remain reliably connected to their control application in their data center. The robots may not move quickly, but they do move frequently. Say that each robot moves about 20 times per minute. If the warehouse has 10,000 robots, the network sees 200,000 moves per minute, which basically means the control plane must process 3,333 moves every second. However, it is not uncommon for a network of this kind to have a much higher density of endpoints, often five times or ten times higher. Conceivably, in a network with 100,000 mobile endpoints, the control plane could be tasked with handling over 33,333 updates every second. In general, you will see that not all roam events cut across xTRs, but some may actually remain within a single xTR and therefore not trigger a control plane update in the overlay. For the sake of simplicity, say that only 30 percent of the roams are actually seen in the overlay. This still leaves about 10,000 updates per second.

If the automated warehouse example doesn't provide you a clear idea of the densities that the IoT can bring and the compound effect of endpoint density and mobility, consider the example of a connected city. Instead of a warehouse and robots, think about a city hosting connected vehicles. It is conceivable for a small city to have 100,000 cars in transit at any given point in time. The city network connects vehicles, and the connectivity is leveraged by other types of endpoints, many of which are mobile.

As if the high density of endpoints and high rates of mobility were not enough of a challenge, the applications controlling the mobile endpoints have low tolerance to

https://t.me/learningnets

Technet24

sustained packet loss. In many of these scenarios, it is required that the network resumes forwarding of traffic to the mobile endpoints in less than 100 milliseconds. It is also important to note that these are relatively large networks where there may be 200 or more xTRs providing connectivity for the mobile endpoints. Figure 7-9 illustrates a sample network in which a large number of endpoints exhibit a high aggregate rate of mobility. The path of traffic between applications in the data center and the mobile endpoints is highlighted.

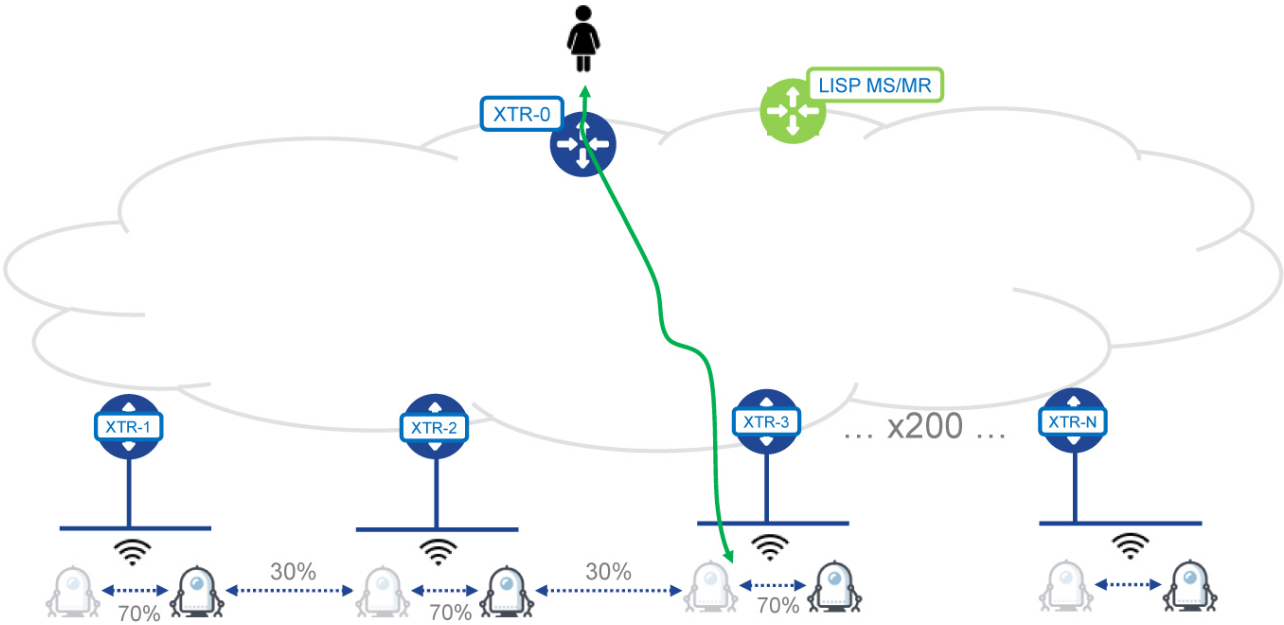


Figure 7-9 *High-Density Mobility Fabric*

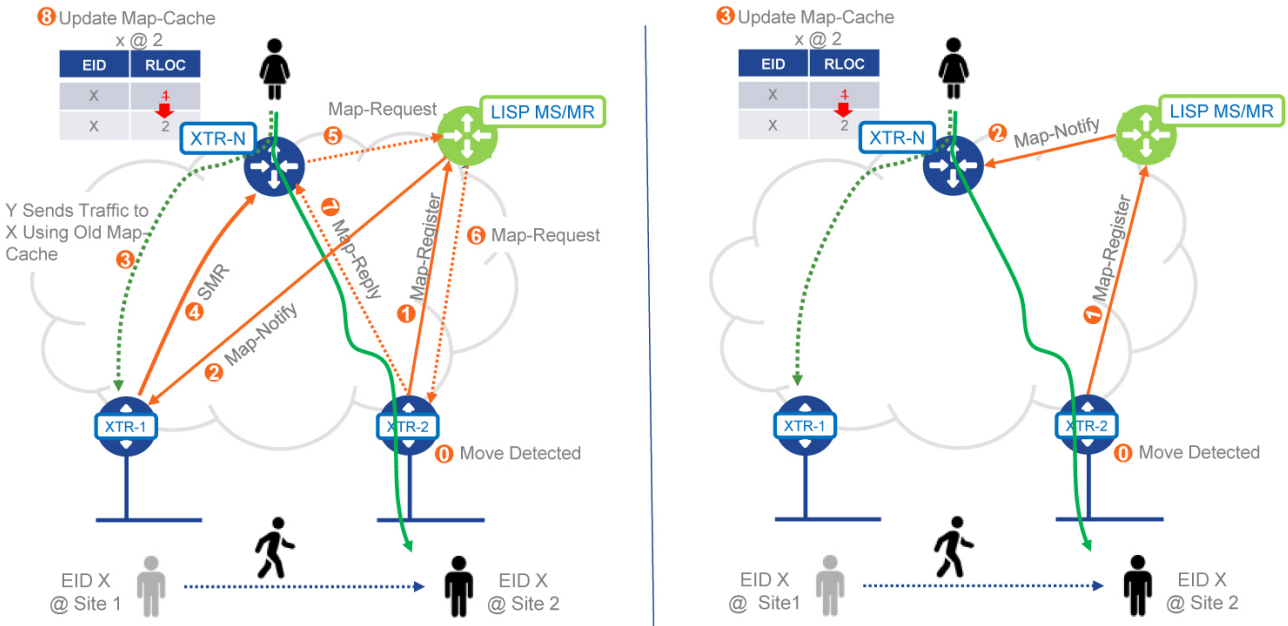
You can build this network in a few ways. One way is to make it a flat L2 network to allow hosts to move anywhere. This approach creates a large failure domain and may not be the best option. Another way to build this network is to use an overlay solution that supports mobility. As of this writing, the two leading options in the market to create such an overlay are Border Gateway Protocol–Ethernet Virtual Private Networks (BGP-EVPN) and LISP. BGP-EVPN is a push protocol, whereas LISP is demand based.

In the case of a push protocol, updates triggered by the moves are sent to every edge device in the network. This means that the control plane not only handles more than 10,000 to 30,000 transactions per second but also replicates these updates to all edge nodes in the network. In short, every single router involved must process all 10,000+ transactions. In Ethernet Virtual Private Network (EVPN), mobility is completed through a handshake of advertisement of routes with a mobility attribute and eventual withdrawal of the route from the previous originator. This means the number of messages per transaction is twice what it could be (one set of messages for the advertisement and another for the withdrawal), so effectively the load is equivalent to over 60,000 transactions per second and must go to every endpoint to function properly. Because every device must process the transactions, every device must have

https://t.me/learningnets

an appropriate amount of processing capacity. And this is where things get challenging when using a push routing protocol for the purposes of handling location information in an overlay. The routing protocol runs path calculations on each message received, which is processor intensive and induces latency in the convergence process. Thus, a traditional push protocol is challenged in terms of the amount of signaling involved in each transaction, the replication of the signaling to a large distribution of edge routers, and the path computation it must complete for each transaction across every network element. You can observe the effects of these factors empirically by measuring system convergence at different mobility rates. Lab testing found that a highly optimized and multithreaded BGP implementation running on dedicated multicore hardware reliably supports a rate of 100 roams per second in a 200-edge router network supporting 16,000 endpoints. Beyond this mobility rate, the central processing unit utilization on the devices is high enough to render the system unstable.

In the case of LISP, the protocol uses an on-demand model and updates are sent only where they are required. In the warehouse example, the only xTRs needing the updates are those that connect the data centers to the fabrics. Furthermore, researchers studied how convergence can be improved by using a combination of EID subscriptions and predictive RLOCs. Using EID subscriptions in this scenario reduces the signaling and the signaling delay dramatically from six events taking 3× round-trip time (RTT) (Register, Notify, Encapsulate Traffic to the departure ETR, SMR, Map-Request, Map-Reply) to one event taking 1× RTT (Map Register → Map Notify). Figure 7-10 illustrates the two scenarios for EID mobility with data-driven SMRs versus subscription-based notifications. The automated warehouse example is an ideal scenario in which to leverage the publish/subscribe optimization for mobility because the subscriber to the EIDs is only the fabric border.



https://t.me/learningnets

Figure 7-10 *Message Reduction and Signaling Delay in EID Mobility*

Additionally, because LISP is not a routing protocol, no path computation loads the CPU of the control plane nodes or causes processing delay. What was found in lab testing is that a single Map-Server comfortably supports 300 roams per second in a nonoptimized monolithic and single-threaded implementation. You can expect that optimizing the LISP Map-Server implementation to leverage multiple cores and parallel processing should improve this performance number significantly.

At this point, you are probably wondering how a mobility rate of 300 roams/sec in a LISP fabric may apply to the high-density case in which more than 10,000 or even 30,000 roams/sec may be required. The answer is horizontal scaling. The LISP control plane is scaled horizontally by using the LISP Delegated Database Tree (DDT). In LISP DDT, the EID space is segmented, and each segment is scoped to different Map-Servers; thus, each Map-Server has to handle a subset of the roam events. The EID space could be scoped to different Map-Servers by prefix, where specific Map-Servers handle registrations, resolution, and mobility events for specific prefixes. Figure 7-11 illustrates how to use LISP DDT to achieve horizontal scaling. Note how EIDs in the red prefix are registered with one Map-Server and EIDs in the green prefix are registered with a different Map-Server. Notifications and updates on red or green prefixes are sent only to interested parties, which means all processing and state handling are fully distributed and no bottlenecks are created.

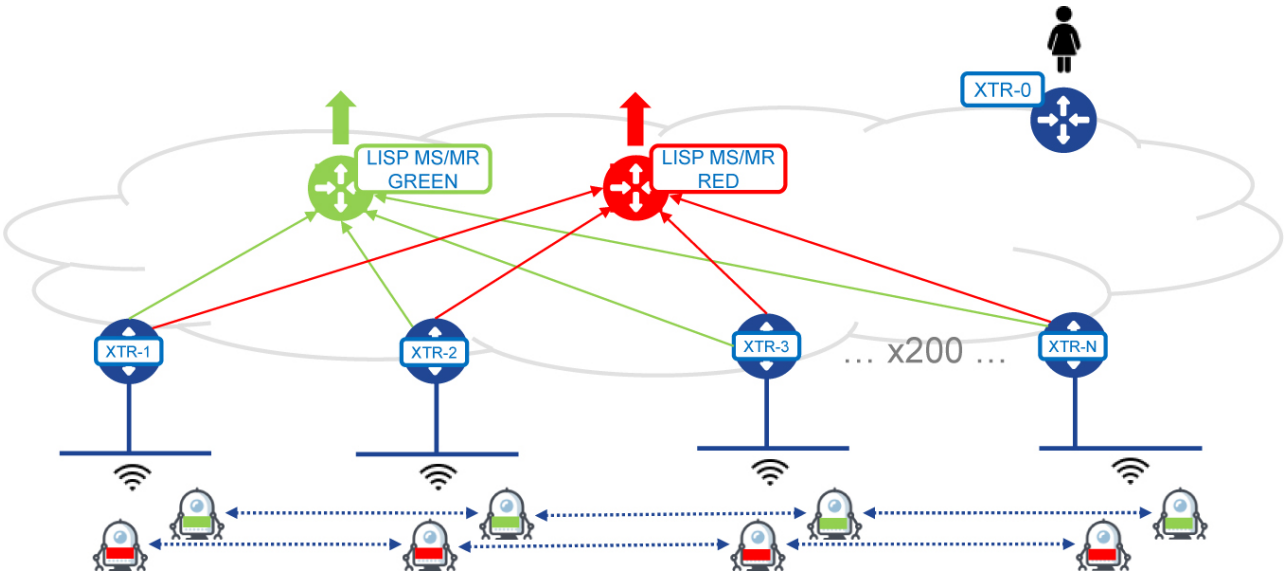


Figure 7-11 *Horizontal Scaling of the Control Plane with LISP Delegated Database Tree (DDT)*

Similar considerations to those discussed so far apply to a scenario in which LISP mobile nodes are used. The main difference is that the xTR detects changes in its connected RLOCs rather than changes in its connected EIDs. The amount of signaling

https://t.me/learningnets

and the scale considerations are similar. The mechanisms proposed for horizontal scale are the same in both the Mobile EID and LISP mobile node cases.

In the extreme case of 30,000 roams/sec, with the current nonoptimized implementations of the LISP Map-Server, you would need 100 Map-Servers. In an optimized case, it is reasonable to expect that a single Map-Server leveraging multiple cores and parallel processing could handle 10 or 20 times the capacity of the nonoptimized implementation, reducing the number of Map-Servers required by one or two orders of magnitude. So, you could build this network with five Map-Servers. Processing capacity continues to grow, and the density of endpoints in the network grows as well. The LISP architecture provides the necessary mechanisms to lighten the load on the control plane and also seamlessly support the horizontal expansion of the capacity of the network.

Achieving this horizontal scale in BGP-EVPN is not simple and may not even be possible. Mechanisms such as RT constraint can achieve some level of horizontal scale of route reflectors if the right set of restrictions are imposed on the organization. Based on the test findings, such a deployment might use 300 BGP route reflectors (each handling 100 roams/sec). Remember, the BGP route reflector code was already optimized, so you can expect these BGP route reflectors to maintain their level of performance. However, horizontally scaling the route reflectors is not sufficient. The edge routers quickly become the bottleneck because BGP sends all updates everywhere, and the updates that were horizontally distributed across multiple route reflectors reconverge at every provider equipment (PE) router receiving the reflected updates. The load on the BGP routers at the edge is the full load of all mobility events, so all elements participating in the BGP fabric must support the large capacity. The conclusion is simple: there isn't a good way of horizontally scaling a BGP deployment in this kind of fabric where every host may move to every edge router. Even if you are able to scale the control plane nodes horizontally, the processing demands reconverge at the edge routers to re-create the scalability problem. Figure 7-12 illustrates the possible options and pitfalls of attempting horizontal scaling for a mobility network with push protocols. In the scenario illustrated, filters are put in place at every PE so that green prefix advertisements are sent only to the green route reflector (Green RR) and red prefix advertisements are sent only to the red route reflector (Red RR). This filtering effectively distributes the load across the different route reflectors. However, to guarantee reachability of every prefix from every PE, the route reflectors reflect the updates received to every PE without filtering. So, each PE effectively receives all mobility updates for all prefixes (in the example, red and green). Each PE must process all updates and run path computation on each update. The PE processing the updates is now the bottleneck because its processing capabilities are usually less than those on the

https://t.me/learningnets
Technet24

route reflectors.

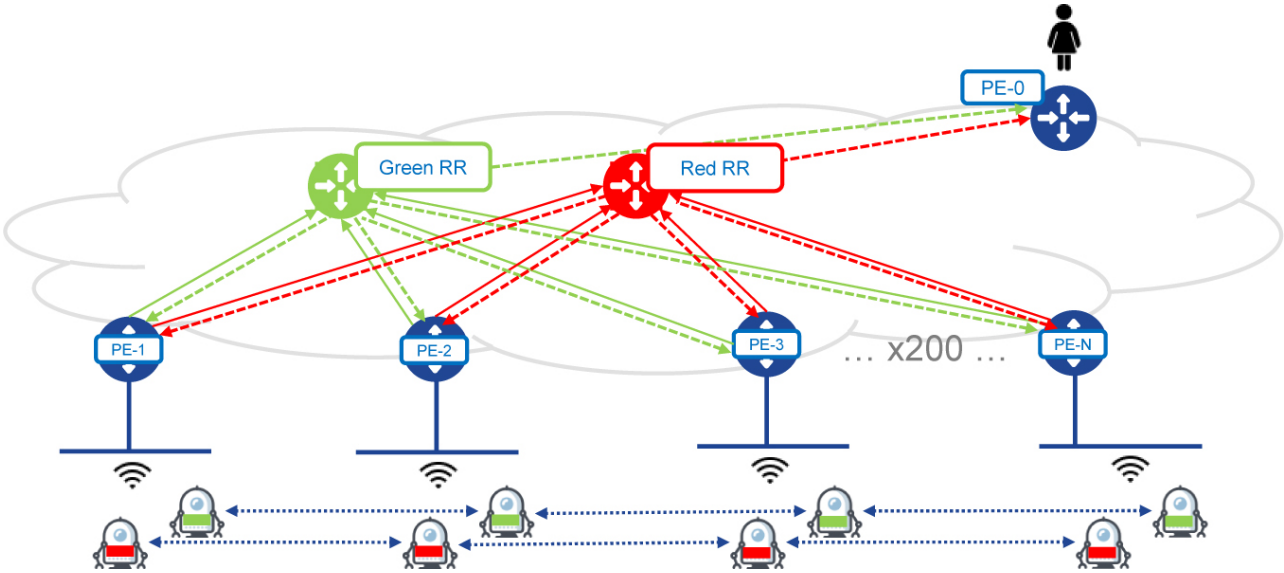


Figure 7-12 *Horizontal Scaling of Control Plane Elements in Push Protocols*

Use Case: Aeronautical Telecommunications Network (ATN)

The civil aviation space presents an interesting set of mobile connectivity requirements. Commercial aircraft need to communicate with Air Traffic Control (ATC) and Aeronautical Operations Control (AOC) resources and applications on the ground. ATC traffic supports flight operations, and AOC refers to the communication between the aircraft and operating organizations on the ground, such as the airline and airline handling partners (for example, fuel suppliers, de-icing companies, aircraft manufacturers). This mission-critical communication is sensitive to packet loss. A reliable and flexible mobility architecture is required to support this environment, and LISP provides the necessary functionality to address the space.

Commercial aircraft connect to the ground network using a variety of radio access networks. The different networks include the General Terrestrial Data Link (LDACS); Satellite Communications (SATCOM) provided by INMARSAT, INTELSAT, or others; and alternative radio links to ground such as WiMAX. Commercial aircraft usually connect simultaneously to more than one of these radio networks and have well-defined policies on which applications should travel over which kind of radio links. This simultaneous connectivity is referred to as multihoming of the aircraft.

The different radio networks connect to an IP core known as the Aeronautical Telecommunications Network/Internet Protocol Suite (ATN/IPS, the ATN using the Internet Protocol Suite). This IP core must support mobility and enable the multihoming of the aircraft. Figure 7-13 shows the proposed architecture for the ATN using LISP to enable the mobility services in the core.

https://t.me/learningnets

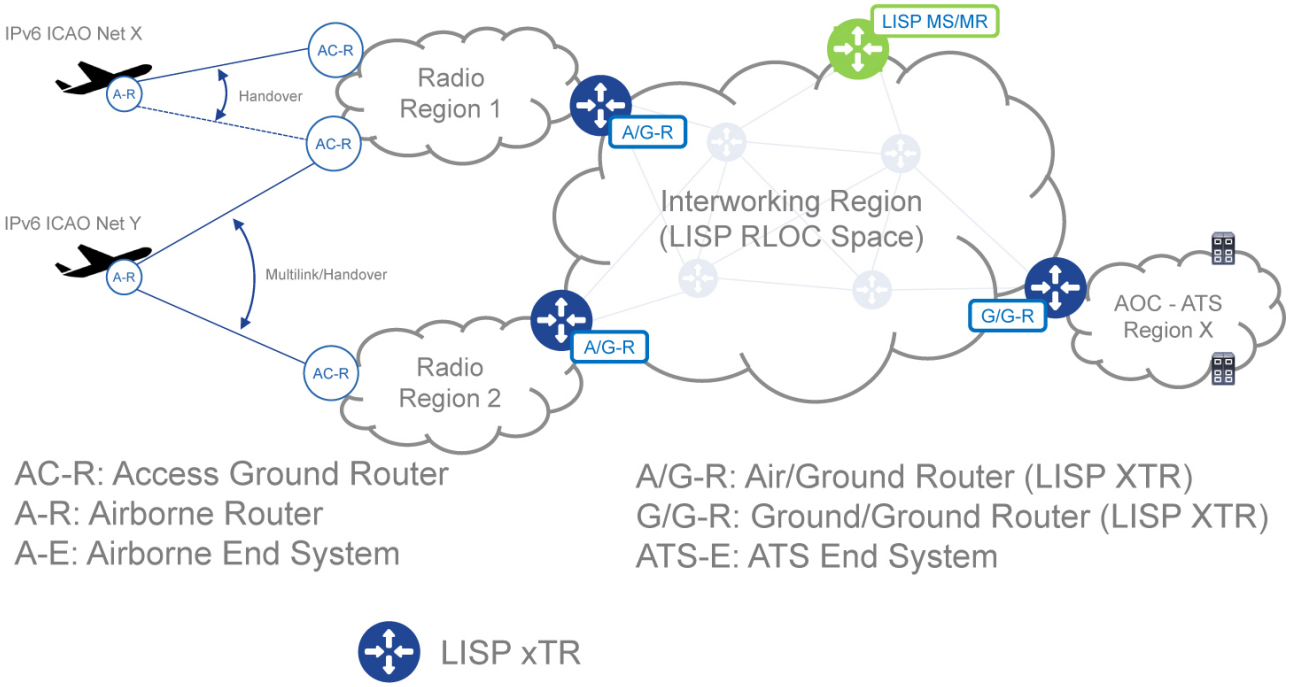


Figure 7-13 *Aeronautical Telecommunications Network Using LISP*

Figure 7-13 illustrates a reference design for the ground-based aeronautical communications network. The aircraft is equipped with an airborne router (A-R) that connects to the different radio regions via the access ground routers (AC-R), and the radio regions connect to the IP core via the air/ground routers (A/G-R). The ground regions connect to the IP core using the ground/ground routers (G/G-R). LISP is deployed over the IP core as xTRs at the A/G-Rs and G/G-Rs, leaving the AC-Rs and aircraft A-Rs untouched. Each aircraft is assigned an IPv6 prefix that uniquely identifies it; this IPv6 prefix does not change as an aircraft roams and is handled as a mobile EID.

As planes multihome and move across radio regions, the xTRs that function as A/G-Rs detect the EIDs of the moving aircraft and register the EIDs with the Mapping Database System. The aircraft report over the radio network their EIDs and the corresponding preferences for the different radio networks, which are translated by the A/G-Rs to priority and weight values in the LISP registration. Based on these priorities and weights, the LISP network forwards traffic from ground to aircraft via the preferred radio network. All the A/G-Rs with an active connection to the aircraft register the aircraft's EIDs. As the aircraft move and change their attachment radio network, the registration changes. The solution leverages the pub-sub mechanisms so that any of the changes in the registration result in an update being published to any ITRs that were sending traffic to the roaming EIDs (in the example illustrated in Figure 7-14, it is G/G-R-X, which subscribes to the Net Y EID). In addition, the ETRs that previously registered the EIDs have subscriptions to the EIDs and get the notification as well. As a result, any moves are notified as quickly as possible to the ITRs so that they can redirect

https://t.me/learningnets

traffic to the new ETRs. If traffic travels to the old ETRs before the ITRs reconverge, the old ETRs are also notified so that they can redirect traffic to the new location. This redirection is also useful in minimizing the amount of packet loss when there is a failure downstream of an ETR. The scenarios of fast convergence and redirection are illustrated in Figure 7-14.

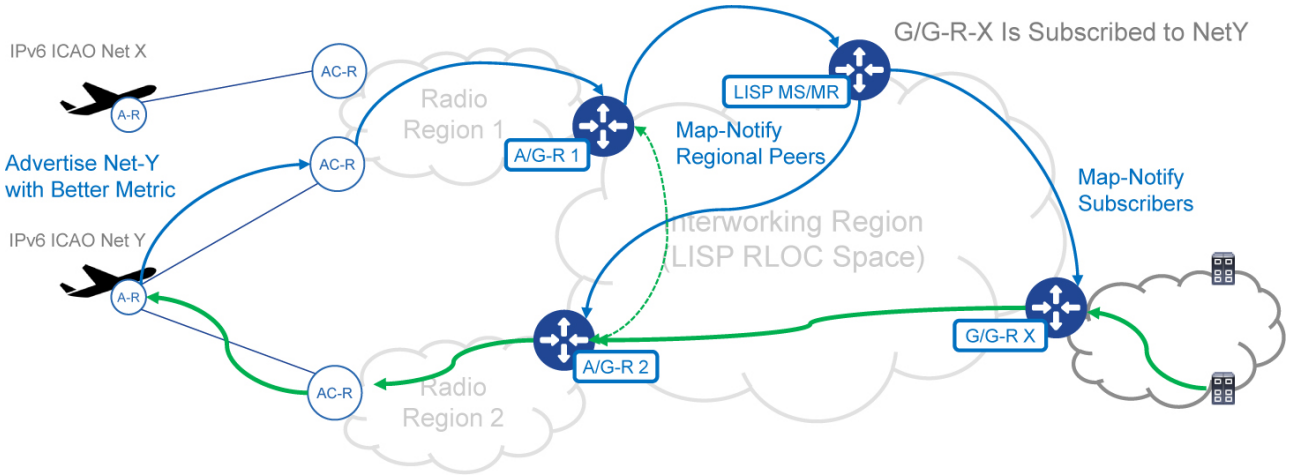


Figure 7-14 *Multihoming and Mobility in the Aeronautical Telecommunications Network Using LISP*

The pub-sub service and scale characteristics of LISP enable an Aeronautical Telecommunications Network service that is virtually lossless, allows aircraft to multihome, provides optimal path forwarding, and is able to support a large number of aircraft with a high rate of aggregate mobility.

An interesting constraint in providing mobile network solutions for aviation is that the hardware and software installed on the aircraft must be kept as minimal and simple as possible. The certification of hardware and software that is used on board an aircraft is a demanding and slow process, so any changes or new requirements for functionality to run on the aircraft must be carefully evaluated. In the ideal scenario, the solutions proposed shouldn't require any changes or additions to the hardware and software already installed on the aircraft. So, planes and other aircraft are equipped with hardware and software certified to connect to the different radio networks as they currently exist.

Use Case: Next-Generation Cellular Networks

The 3rd Generation Partnership Project (3GPP) unites seven telecommunications standard development organizations and provides these organizations with a stable environment to produce the necessary standards to evolve cellular telecommunications network technologies. As of this writing, 3GPP is heavily focused on the definition of the standards for the 5th Generation (5G) cellular telecommunications network. The

https://t.me/learningnets

LTE/5G mobile network is structured as illustrated in Figure 7-15.

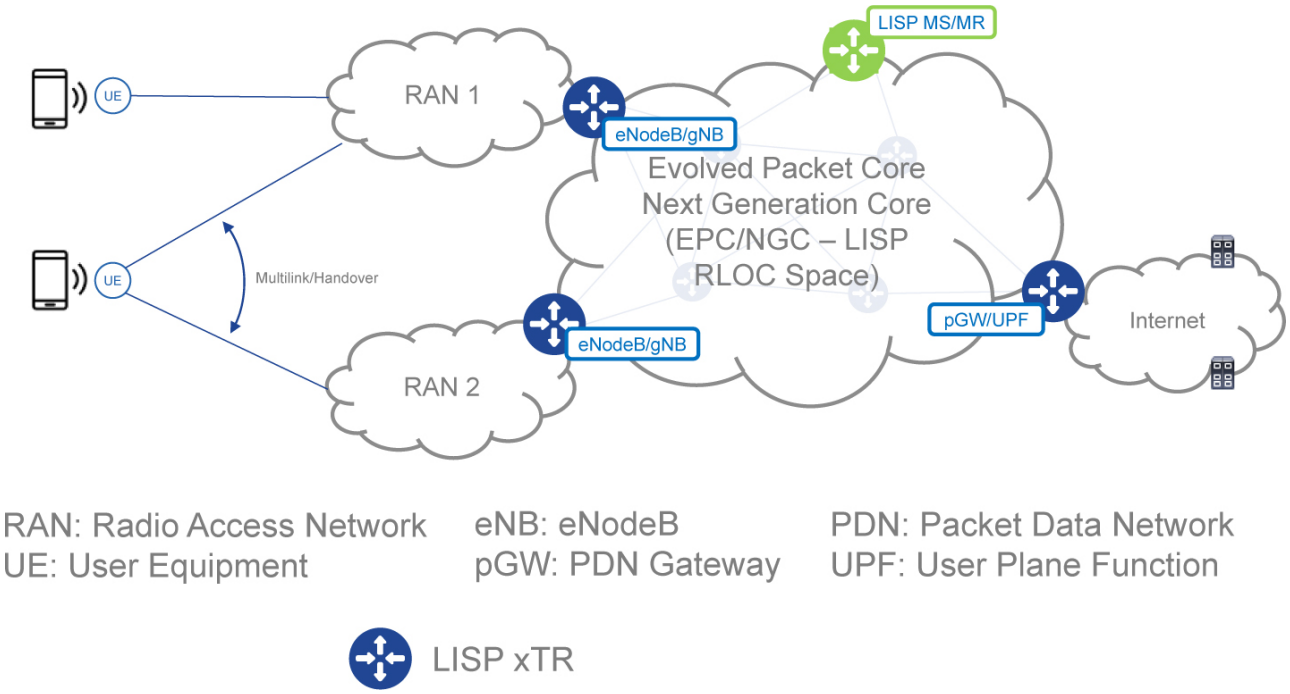


Figure 7-15 *LTE/5G Mobile Network Architecture*

Similar to the mobility network defined for aeronautical telecommunications, the LTE network is designed as a collection of radio access networks that are aggregated in an IP core. The nomenclature that 3GPP uses is rather specific and captured in Figure 7-15. As seen in the figure, user equipment (UE) connects to the network via the radio access networks (RAN). RANs are aggregated into the core, which provides connectivity between the RANs and to the Internet. The core is referred to as the evolved packet core (EPC) in the 4G/LTE network specification and is referred to as the next-generation core (NGC) in the 5G specification. We refer to the core generically as the *core* or the *EPC/NGC*. The devices connecting the RAN to the core are referred to as eNodeB (eNB) and gNB in the 4G and 5G specifications, respectively. The devices connecting the core to the Internet are referred to as PDN-Gateway (pGW) and UPF in the 4G and 5G specifications, respectively.

Similar to the mobility architecture proposed for the aeronautical network, LISP is deployed in the core and is not extended to the radio access networks. In the specific case of the 4G/5G networks, this means that the eNB/gNB devices, as well as the pGW/UPF devices, are LISP xTRs. Traffic is LISP encapsulated over the core (EPC/NGC) where mobility is implemented according to the LISP procedures described in the ietf-lisp-eid-mobility specification. The mapping of LISP roles to the cellular core is described in detail in draft-farinacci-lisp-mobile-network.

As the mobile network evolves from 4G to 5G, the 3GPP organization is committed to address a comprehensive and ambitious set of requirements. These requirements are

https://t.me/learningnets

captured in the 3GPP Technical Specification TS 22.26. Some of the requirements impacting the evolved packet core that the 3GPP organization has set to fulfill include

• Network slicing

• Ultra-low latency

• High data rates

• High density

• Fixed-mobile convergence (FMC) multihoming

• Security

Next, we explore these requirements and see how a LISP-enabled Evolved Packet Core (EPC) addresses them.

Network Slicing

Network slicing refers to the network's ability to deliver multiple logical networks or slices in the EPC. Different operations and services, as well as tenant-specific services, are key to the commercial viability of the 5G network. The ability to provide robust, scalable, and feature-rich network slicing is a key requirement in the 5G charter.

Previous chapters discussed in detail how LISP delivers virtual networks by segmenting the control plane and the data plane. Some of the advantages of the LISP-based segmentation include the preservation of a uniform operational mode in the presence or absence of virtual network slices, the ability to provide consistent scale independent of the number of network slices because these are implemented simply as an extension of the IP name space and don't require separate peerings or queues for the different slices, and the ability to support all LISP features such as optimized multicast across multiple slices just as it is supported without any segmentation.

Ultra-Low Latency

Among the notable requirements being pursued for the 5G network is an extreme requirement for low latency. The latency budget proposed is expected to be sub 1 millisecond. This requirement is aimed at addressing applications such as those of the tactile Internet, in which any delay perceivable by humans trying to extend tactile interactions over the network may be unacceptable. At the distances proposed, sub-millisecond latency is physically impossible; however, the experience of sub-

https://t.me/learningnets

millisecond latency may be emulated through a combination of signal prediction, content caching, and optimized network efficiency. It is this last item where LISP can play a role in the 5G latency race by guaranteeing shortest path communications in the face of mobility. The shortest path forwarding achieved using LISP is in contrast with the established mobility models in which a home agent is required. Detouring traffic to a home agent adds unnecessary latency to the communications. Meanwhile, LISP provides optimal path connectivity while supporting high rates of mobility in dense environments. LISP provides the shortest path to content and applications that are hosted in a multitude of highly distributed co-location and data center facilities that are part of the 5G network. This distribution of content and applications is geared toward minimizing the distance between the content and the endpoints. By guaranteeing shortest path connectivity to these resources, LISP helps enable the vision of the tactile ultra-low-latency Internet.

High Endpoint Density

The 5G network is being designed with the Internet of Things in mind. The densities of endpoints expected to be supported by the 5G network is orders of magnitude higher than what is currently supported by the 4G standards.

The right choice of a control plane for the evolved packet core is key to its success. The combination of a large number of endpoints with relatively high rates of mobility and entropy is an interesting challenge. In an earlier section on high rate mobility, we discussed how LISP excels in fulfilling these requirements versus traditional push protocols. The 5G case is an even larger scale example of a scenario in which the demand nature of LISP is critical to the success of the future network.

Fixed-Mobile Convergence (FMC) Multihoming

5G combines a multitude of radio access networks. Of particular interest is the combination of cellular and Wi-Fi for multihoming as well as offloading. Next-generation mobile networks must support the seamless combination of the cellular network infrastructure and the fixed network infrastructure. A seamless user experience requires that voice and data traffic be handed off between the cellular and fixed networks without any perceivable loss of information. To achieve this effect, an endpoint must be able to roam seamlessly between the 3GPP and Wi-Fi networks, for example. To support this type of seamless and lossless roaming, LISP supports flexible multihoming. As discussed in the section on Aeronautical Telecommunications Network design, the multihoming is regulated by dynamic policy that allows the traffic to be load-balanced across the multiple access networks or fully moved from one to the other.

https://t.me/learningnets

Multihoming is an intrinsic characteristic of LISP. As discussed in previous chapters, LISP provides a streamlined environment in which endpoints are easily multihomed while following a simple but effective policy framework. As described in this chapter, the LISP multihoming mechanisms are well fitted to the mission of providing seamless roaming.

LISP can implement FMC in two ways: one is with LISP mobile node, by simply being multihomed to LTE and Wi-Fi. The other way LISP implements FMC is by deploying an xTR on the eNodeB and an xTR on the Wi-Fi AP (or its first hop-switching element), where the mobile phone is a multihomed mobile endpoint with an EID (and the RLOC is not co-located with it).

Security

As you place more trust in digital assets and digital environment, security is paramount. This book examined in detail the security attributes of LISP. These attributes align tightly with the security requirements of the 5G core and in many cases exceed the requirements of the 3GPP partnership.

Use Case: Mobile Environment for Media Broadcasting

Media companies cover events in a multitude of locations around the world. The equipment setup and connectivity requirements required at the event locations from the media company do not change across different venues. Media companies benefit from normalizing the setup of their field units so that they can deploy a tested and reliable broadcasting environment to any location they may need to deploy to. In many cases, media companies set up their field units in shipping containers that can be physically moved to the different locations. The goal is to plug the shipping container to any provider, and things should just work.

For this model to succeed, it is critical that the field units operate in the same way regardless of the services and characteristics of the connectivity providers in the different locations. A provider in one location may offer IPv6 and multicast services, whereas a provider in a different location may not offer multicast or IPv6. Additionally, it is highly desirable that the IP addressing of the equipment in the field unit be totally independent of the connectivity provider being used.

A LISP overlay gives the media field units the provider independence required. The overlay allows the addressing of the field unit devices to be totally independent of the connectivity provider by simply including a LISP xTR in the field unit or shipping container as the router that connects the field unit to the local network provider. So, the

https://t.me/learningnets

connectivity provider supports the RLOC space while the EID space simply travels in the container with the field unit. This is a clear example of the multihomed LISP mobile site model that was discussed earlier. The RLOCs change from one event to another, but the EIDs of the broadcasting equipment in the shipping container remain constant from one event to the next. The network services that the field unit relies on, such as multicast or IPv6, are properties of the overlay and therefore independent of the local network provider.

The use of LISP for these broadcasting field units allows media companies to provide a stable and predictable network environment that is quickly and effectively deployed in the field. It also allows media companies to experiment with new broadcasting equipment without being constrained by any limitations that the local connectivity provider may have. Rather than spinning cycles solving networking problems, the media company can focus on its core business concern, which is processing and transmitting media.

Use Case: Blockchain Network

In 2009, a group of anonymous contributors released Bitcoin into the open-source community. Beyond becoming the most popular cryptocurrency in circulation, Bitcoin set the stage for the community to learn many valuable lessons around deploying a blockchain to achieve decentralized consensus and enable a digital ledger in which value could be exchanged in a trusted manner. Since then, cryptocurrency and decentralized consensus using blockchain technology have evolved significantly, and there are many efforts to produce next-generation blockchain networks to serve applications well beyond cryptocurrency. Some of these applications include storing and exchanging intellectual property rights, health records, or other information of value. Smart contracts deployed on the blockchain can enable applications such as escrow, fulfillment of insurance claims, and supply chain transaction brokering for logistics operations. Communities in need of an election and expression system free of governmental control and corrupt tampering can use the blockchain to run elections and polls without having to rely on an electoral body that may be illicitly influenced. The applications are many, and they all are centered around the ability to provide trust between parties involved in a transaction without involving a third party to broker these transactions.

The blockchain network is a highly distributed ecosystem of mining nodes and digital wallets. Optimal connectivity between these nodes and the wallets is required. The wallets are mobile because they are deployed in people's portable computers and smartphones. The mining nodes were traditionally stationary, but in the spirit of

https://t.me/learningnets

providing infrastructure that can be accessed anywhere in the world, efforts are underway to launch a network of mining nodes as Low Earth Orbit (LEO) satellites, which would be mobile as well. The mobility characteristics of the LISP mobile node are particularly relevant to the blockchain network where LISP mobile node can be installed on mining nodes as well as wallet nodes to provide a network overlay that is fully aligned with the needs of the blockchain.

The LISP network overlay provided by the LISP mobile node would optimize connectivity in this highly mobile network, as described in this chapter, to avoid suboptimal routing and minimize packet loss on reconvergence. The LISP mobile node would also enable optimal overlay multidestination replication with multicast to enable the efficient distribution of much of the consensus-negotiating traffic that is critical to the blockchain. The preservation of the EIDs that identify the mining and wallet nodes is a key characteristic of the LISP mobile network to be leveraged by the blockchain where the hashes in the chain are linked to the EIDs of the nodes to provide an enhanced level of security. In addition to all these benefits, LISP provides a secure environment where the communications can be encrypted and the control plane secured with modern hashing and signing so that the exchange of consensus messaging in the chain is effectively secured at the network level.

The blockchain in conjunction with the space network of LEO satellites is one example of the applicability of LISP mobility (and LISP in general) to a multitude of spaces.

SUMMARY

The next-generation network must support mobility at very high rates and with unprecedented endpoint density. The requirements are stringent, and a new paradigm in networking must be put in place to achieve the required scale and speeds of mobility.

This chapter examined the different mobility mechanisms present in LISP and how the LISP architecture can adapt to the reference architectures used for global mobility networks such as the 3GPP 4G/5G and the Global Aeronautical Telecommunications Network. It also analyzed a scenario in which a high density of endpoints connected to an access network could push the limits in terms of the network scale currently possible and how LISP provides the necessary functionality to address the requirements and unlock the dead end that current networking technology may be up against.

Possibilities for applications of LISP mobility in current and future networks are many. With global initiatives to run consensus networks in space that are independent of regulation, with LISP, the sky is no longer the limit.

https://t.me/learningnets