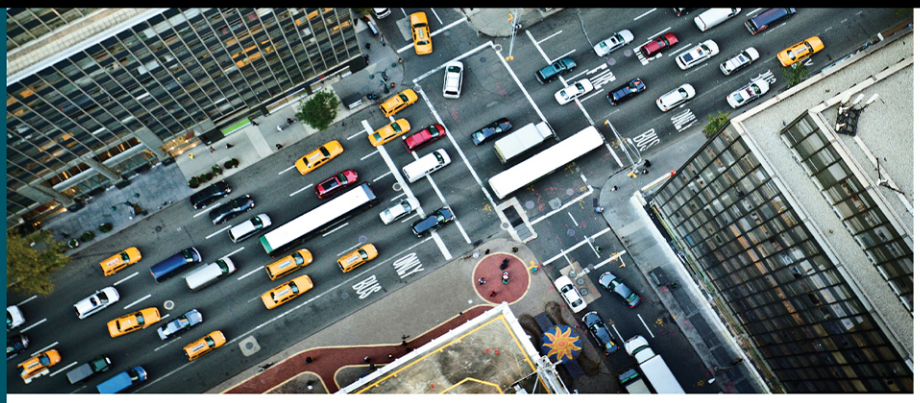


# The LISP Network

## Evolution to the Next Generation of Data Networks

Victor Moreno
Dino Farinacci

ciscopress.com

# The LISP Network

## Evolution to the Next Generation of Data Networks

Victor Moreno
Dino Farinacci

Cisco Press

# The LISP Network

Victor Moreno

Dino Farinacci



Published by:
Cisco Press



01 19

Library of Congress Control Number: 2018964522

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

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

**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:** codemantra

**Indexer:** Cheryl Lenser

**Proofreader:** Abigail Manheim

**Americas Headquarters**
Cisco Systems, Inc.
San Jose, CA

**Asia Pacific Headquarters**
Cisco Systems (USA) Pte. Ltd.
Singapore

**Europe Headquarters**
Cisco Systems International BV Amsterdam,
The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at **www.cisco.com/go/offices.**

## About the Authors

**Victor Moreno** is a Distinguished Engineer at Cisco Systems responsible for the definition of next-generation network architectures. Victor has more than 20 years of industry experience focused on enterprise and data center network design and architecture. A recognized expert in his field, Victor holds several patents which are at the foundation of the key protocols and networking technologies that have enabled the evolution of networking to its current state. He has worked directly on the designs of global enterprises and service providers and has done extensive research on the topic of network virtualization, being a driving force within Cisco and earlier Digital Equipment Corporation for new product definition and technological direction. Victor is the co-author of the Cisco Press title *Network Virtualization* and has published a multitude of technical papers and articles on behalf of Cisco Systems. Victor holds a degree in electrical engineering from the Simón Bolívar University, as well as master's degrees and specializations from the Universities of York, Cambridge, and Stanford. Victor is an active contributor to the definition, implementation, and standardization of the Locator/ID Separation Protocol (LISP).

**Dino Farinacci** is a software engineer by trade and a technology visionary by passion, advancing the state of the art in computer networking. As one of the first Cisco Fellows, Dino holds more than 40 Internet and networking-related patents and has been a major IETF contributor for nearly 30 years with approximately 50 RFCs and Internet Drafts published. Dino is the founder of lispers.net, a nonprofit engineering organization, where he now focuses on design and deployment of LISP for IoT, cryptocurrency, and 5G mobile networks.

Dino is one of the original RFC co-authors of LISP, dating back to 2007, and has had the pleasure of writing two implementations of the protocol. He currently does consulting for large startup networking vendors and helps users deploy network designs using LISP and other architectures. If you can name an Internet protocol, there is a good chance Dino has designed and implemented it in widely deployed products. Over his career working at the NSA, CDC, 3Com, Procket, and Cisco, he has worked on dozens of operating systems, network protocols, and infrastructure systems.

## About the Technical Reviewers

**Lukas Krattiger**, CCIE No. 21921 (Routing/Switching and Data Center), is principal engineer, Technical Marketing, with more than 15 years of experience in data center, Internet, and application networks. Within Cisco, he specializes in data center switching, overlay architectures, and solutions across platforms. Lukas is a double CCIE (R&S and Data Center) with several other industry certifications and has participated in various technology leadership and advisory groups. Prior to joining Cisco, Lukas was a senior network engineer with System Integrators and Service Providers, where he was responsible for data center and Internet networks. Since joining Cisco, he has covered various technologies within the data center as well as enterprise networks portfolio, and he has built foundational solutions for customers and partners. He is from Switzerland and currently lives in California with his wife and one wonderful daughter. He can be found on Twitter at @ccie21921.

**Sameer Merchant** is a data center networking consultant. He is actively involved in Data Center Switch Software, L4-L7 Network Services, and Network Analytics software design. He has more than 20 years of networking software development experience. He was a Distinguished Engineer at Cisco and a lead developer of LISP forwarding software on Cisco Nexus 7000 Data Center Switch. He has worked as a lead software engineer on the Cisco core routers, data center switches, network virtualization, and SDN solutions.

## Dedications

To my beautiful wife, Shannon, and our beloved children, Noah and Stella. Thanks for your unwavering love and support throughout the years. Thank you for your endless patience toward my "missions" to humanize technology and change the world. Above all, thanks for being the source of inspiration for every meaningful adventure in my life and for being the driving force behind every worthwhile venture I have dared pursue.

—*Victor*

I would like to dedicate this book to my wife, Nancy Farinacci. I thank her for the loving support throughout LISP's lifetime and for listening to the word *LISP* for more than a decade. She keeps asking me "When is LISP going to be done?" but doesn't really want the answer because she has witnessed how dedicated and satisfying the technology has been for me. Throughout our entire marriage, she has been my greatest support and motivation for all my endeavors. Love you and thank you, sweetie, for making my commitment your commitment. Your one and only, Dino.

—*Dino*

# Acknowledgments

Victor would like to thank Dino for making LISP the reality it is today. Thanks for the contagious conviction that the world can be a better place and that technology is at the heart of that improvement. LISP would not exist without you, Dino. Dino would like to thank Victor for making *The LISP Network* book a reality. It would not exist without you Victor!

A special thank you goes to Brett Bartow for his infinite patience and perseverance. Thanks for sticking with us and making this book a reality.

We would like to thank the LISP community for their contributions to the technology and its applications.

In particular, we would like to thank Dave Oran for planting the seeds for the original ideas for LISP. His consultation continues to provide value to the LISP community at large.

A special and appreciative thank you goes to Noel Chiappa for providing architectural impetus over the past decades on separation of location and identity, as well as detailed reviews of the LISP architecture and documents, coupled with enthusiasm for making LISP a practical and incremental transition for the Internet.

A big thank you goes to those who took the seminal ideas, gave them concrete shape, and set them in motion through the initial specifications and implementations. Specifically, thank you to David Meyer, Vince Fuller, and Darrel Lewis, who took the charge for many years alongside Dino and made LISP the focus of their careers.

We would like to specially acknowledge the work and contributions of all of those who saw the value of the technology. They include Scott Brim, Andrew Partan, John Zwiebel, Jason Schiller, Lixia Zhang, Dorian Kim, Peter Schoenmaker, Vijay Gill, Geoff Huston, David Conrad, Mark Handley, Ron Bonica, Ted Seely, Mark Townsley, Chris Morrow, Brian Weis, Dave McGrew, Peter Lothberg, Dave Thaler, Eliot Lear, Shane Amante, Ved Kafle, Olivier Bonaventure, Luigi Iannone, Robin Whittle, Brian Carpenter, Joel Halpern, Terry Manderson, Roger Jorgensen, Ran Atkinson, Stig Venaas, Iljitsch van Beijnum, Roland Bless, Dana Blair, Bill Lynch, Marc Woolward, Damien Saucez, Damian Lezama, Attilla De Groot, Parantap Lahiri, David Black, Roque Gagliano, Isidor Kouvelas, Jesper Skriver, Fred Templin, Margaret Wasserman, Sam Hartman, Michael Hofling, Pedro Marques, Jari Arkko, Gregg Schudel, Srinivas Subramanian, Amit Jain, Xu Xiaohu, Dhirendra Trivedi, Yakov Rekhter, John Scudder, John Drake, Dimitri Papadimitriou, Ross Callon, Selina Heimlich, Job Snijders, Vina Ermagan, Fabio Maino, Chris White, Clarence Filsfils, Alia Atlas, Florin Coras, and Alberto Rodriguez.

We would like to acknowledge the work on the early implementations and deployments that put LISP on the map at Cisco: Isidor Kouvelas, Jesper Skiver, Parna Agrawal, Venu Venugopal, Dan Alvarez, Selina Heimlich, Gregg Schudel, Jayshree Ullal, Ram Velaga, Tom Edsall, and Murali Basavaiah. A special thanks to John Chambers for the incredible support during the first phase of implementations at Cisco. For the inspired willingness to carry the torch forward: Fabio Maino, Sanjay Hooda, Lukas Kratiger, Satish Kondalam, Dana Blair, Ravi Chandrasekaran, Sachin Gupta, and all the teams that followed their lead.

Recognition goes to the IETF LISP Working Group for the efforts in legitimizing and refining the technical specifications of the protocol. We cannot overstate the efforts that help LISP move through the formal standards track at the IETF. A big thanks goes to all those involved; in particular, Joel Halpern, Luigi Iannone, Deborah Brungard, Fabio Maino, Scott Bradner, Kyle Rose, Takeshi Takahashi, Sarah Banks, Pete Resnick, Colin Perkins, Mirja Kuhlewind, Francis Dupont, Benjamin Kaduk, Eric Rescorla, Alvaro Retana, Alexey Melnikov, Alissa Cooper, Suresh Krishnan, Alberto Rodriguez-Natal, Vina Ermagan, Mohamed Boucadair, Brian Trammell, Sabrina Tanamal, and John Drake. The contributions they offered greatly added to the security, scale, and robustness of the LISP architecture and protocols. We would like to acknowledge the valuable contributions of Professor Alberto Cabellos and his brain trust at the University of Catalunya.

And a big thank you to all of you who have deployed and operationalized LISP in your networks and who make LISP part of your daily jobs.

To our technical editors, Sameer and Lukas, thank you for your thorough reviews. In addition to your technical accuracy, your insight sparked ideas and inspired content that made this book so much richer.

We would like to thank the Cisco Press team for their assistance and insight throughout this project. You have been a pleasure to work with, and your attention to detail is simply amazing.

# Contents at a Glance

Introduction xv

Chapter 1 LISP and the Future of Networking 1

Chapter 2 LISP Architecture 9

Chapter 3 Data Center Trends 31

Chapter 4 The Wide-Area Network: Bringing Traffic from Access to the Data Center 57

Chapter 5 Mega-Scale Access Networks: LISP, User Access, and the Internet of Things 79

Chapter 6 Security 105

Chapter 7 LISP and the Next-Generation Mobile Network 131

Index 157

# Contents

Introduction xv

**Chapter 1 LISP and the Future of Networking 1**

A Brief History of LISP: Motivation, Base Premises, Evolution 5

LISP in the Standards and Open Community 6

Use Cases for LISP: Supporting Future Trends 7

**Chapter 2 LISP Architecture 9**

Seminal Idea: Location-Identity Separation 9

Map and Encapsulate 11

Demand-Based Routing and Caching 12

LISP Roles 14

Tunnel Routers 14

*Ingress Tunnel Routers 14*

*Egress Tunnel Routers 15*

Proxy Tunnel Routers 15

*Proxy Ingress Tunnel Routers 16*

*Proxy Egress Tunnel Routers 16*

Mapping Database System 17

An Asset-Controlled Mapping Database 21

Networking Beyond Traditional Address Types 22

The LISP Data Plane 23

Tunnel Entropy 24

Segmentation 24

Locator Status Validation 25

Path Reliability 26

Confidentiality and Authentication 27

Alternative Data Plane Formats 27

NAT Traversal 29

Summary 30

**Chapter 3 Data Center Trends 31**

A Brief History of Application Virtualization 31

Multitiered Applications, Virtualization, and the Network 34

Evolving Switching Fabrics 37

Optimizing Connectivity to the Data Center with LISP 39

Mobility: Subnets Really Don't Work 42
Segmentation: 32 Bits Needed 46
Device Segmentation 48
Control Plane Segmentation 49
Data Plane Segmentation 50
Extranet VPNs 50
Policy: The Network as an Enforcer 51
The Hybrid Cloud and Carrier Neutrality 54
Summary 56

**Chapter 4 The Wide-Area Network: Bringing Traffic from Access to the Data Center 57**

Modern WAN Services 57
Hybrid WAN: Efficient xTR Multihoming 60
Scale Considerations 65
Logical Topologies: Peer-to-Peer Connectivity and Service Insertion 67
Security: Connection Integrity and Confidentiality 70
Segmentation 71
The Access Network: Multisite Considerations 72
Manageability 76
Summary 77

**Chapter 5 Mega-Scale Access Networks: LISP, User Access, and the Internet of Things 79**

Access Networks Using LISP 81
LISP Access Network Design 81
*Connecting to External Networks 85*
Mobility and Wireless Integration 87
Segmentation 90
Zero Configuration Networking: Service Discovery 91
Situational Policy (Beyond Just Location) 92
Applications 92
Optimized Campus and Branch Access 92
Connected Home 93
Campus Dormitory Rooms: A Virtual Home 94
LISP-Based Air-to-Ground Network 95
Endpoint Tracking Applications: Geo-location 96

The Internet of Things 97
Security and Integrity 98
Sensors: Mega-Scale Aggregation of Very Little Data 99
A Protocol Fitted for Low-Power, Light-Footprint Applications 102
A Lightbulb for Utopia 103
Summary 104

**Chapter 6 Security 105**

Attack Surfaces, Lateral Moves, and Bot-nets 105
Policy, Segmentation, and the Virtual Perimeter 106
Macro-segmentation 109
Micro-segmentation 111
Process-Level Segmentation 113
How to Integrate the Control Plane into the Assurance Loop 116
Traffic Steering and Service Chains 117
Cryptography in LISP 117
Public-Key Cryptography 117
Symmetric Cryptography 119
Integrated Key Exchange 120
How the LISP Control Plane Is Secured 123
Enhanced Control Plane Security 124
*LISP-SEC 124*
*Threats Addressed by LISP-SEC 126*
*LISP Elliptic Curve Digital Signature Algorithm (ECDSA) Authentication and Authorization 127*
Anonymity in LISP 129
Summary 130

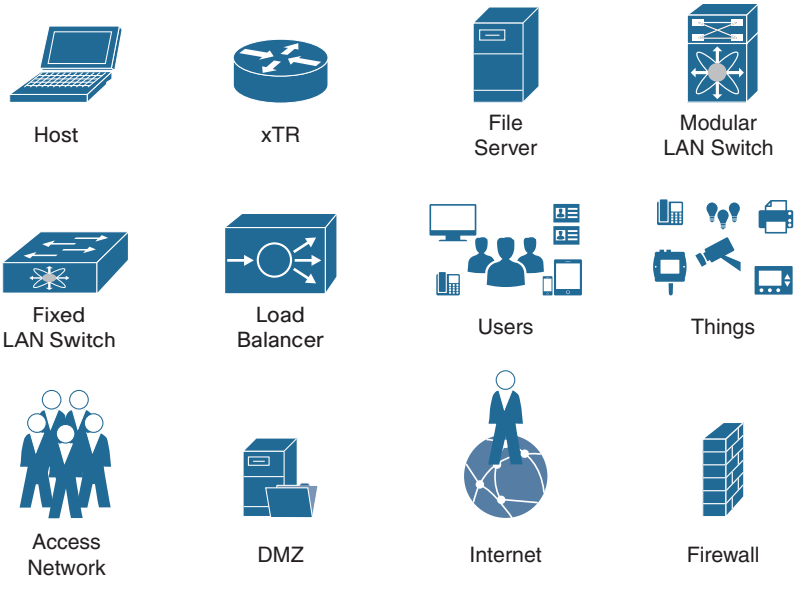
**Chapter 7 LISP and the Next-Generation Mobile Network 131**

LISP EID Mobility and LISP Mobile Node 131
LISP EID Mobility 132
*LISP EID Mobility Mechanics 133*
LISP Mobile Node 136
*LISP Mobile Node Mechanics 137*
Mobility Convergence Optimization 138
Redirection 138
Pub-Sub 139
Predictive RLOCs 141

Use Cases 143
Use Case: High Rate Mobility 143
Use Case: Aeronautical Telecommunications Network (ATN) 148
Use Case: Next-Generation Cellular Networks 150
*Network Slicing 151*
*Ultra-Low Latency 152*
*High Endpoint Density 152*
*Fixed-Mobile Convergence (FMC) Multihoming 153*
*Security 153*
Use Case: Mobile Environment for Media Broadcasting 153
Use Case: Blockchain Network 154
Summary 155

**Index 157**

## Icons Used in This Book




## Reader Services

**Register your copy** at www.ciscopress.com/title/9781587144714 for convenient access to downloads, updates, and corrections as they become available. To start the registration process, go to www.ciscopress.com/register and log in or create an account.* Enter the product ISBN 9781587144714 and click Submit. When the process is complete, you will find any available bonus content under Registered Products.

*Be sure to check the box that you would like to hear from us to receive exclusive discounts on future editions of this product.

# 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 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**

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 planes 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.

*This page intentionally left blank*

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.

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 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

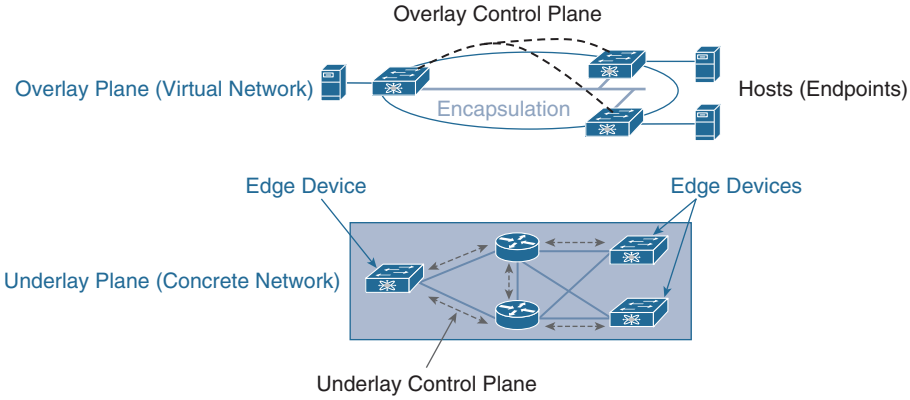



**Figure 2-1** *Functional Components of a Network Overlay*

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

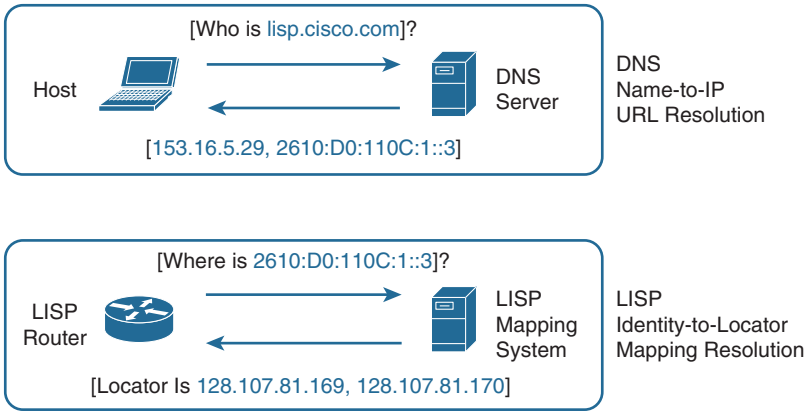



**Figure 2-2** *LISP and DNS*

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 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.

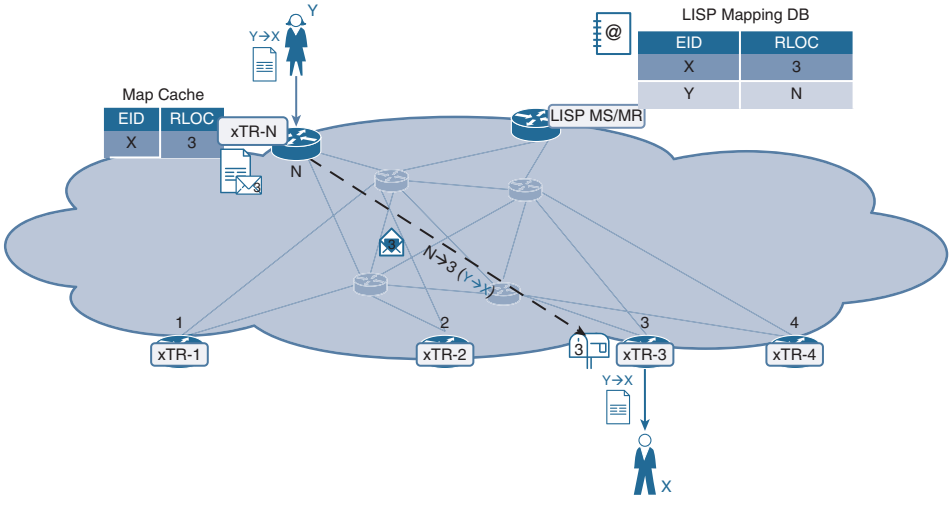



**Figure 2-3** *Location Resolution on Demand*

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 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.

### 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, and yes, the phone book used to also include the addresses of the subscribers listed in it).

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-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.

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 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.

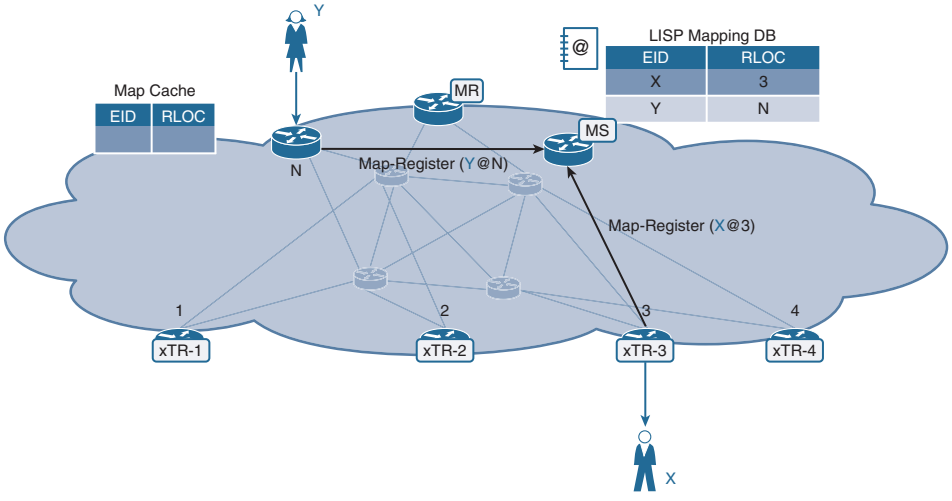



**Figure 2-4** *Registration of EID to RLOC Mappings*

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.

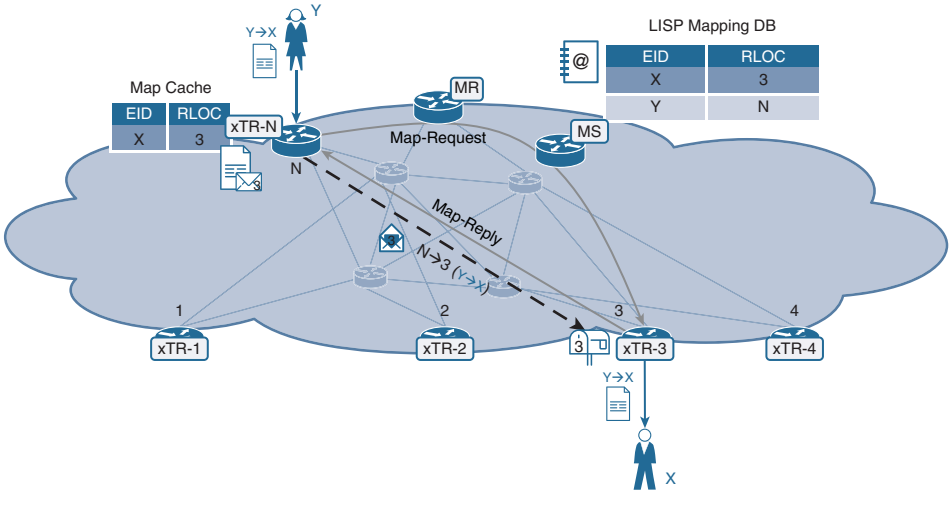



**Figure 2-5** *Mapping Resolution*

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.

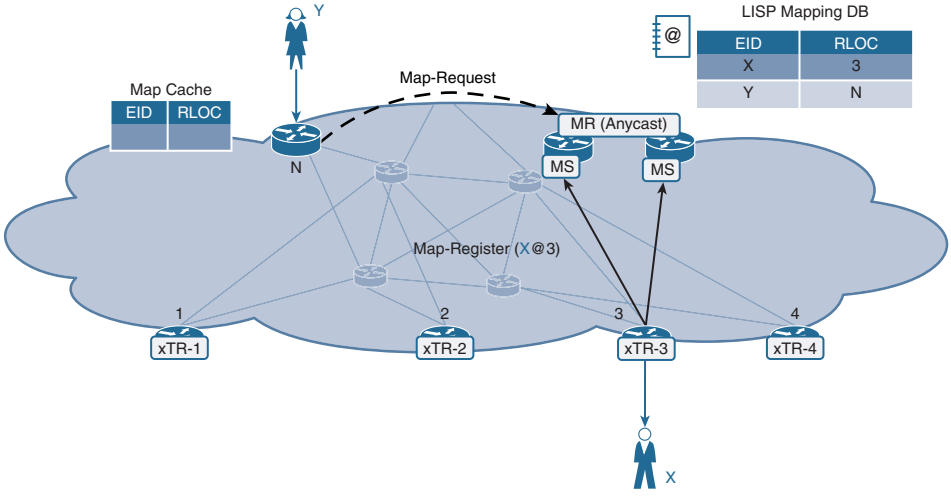



**Figure 2-6** *Map-Server/Map-Resolver Resiliency*

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

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.

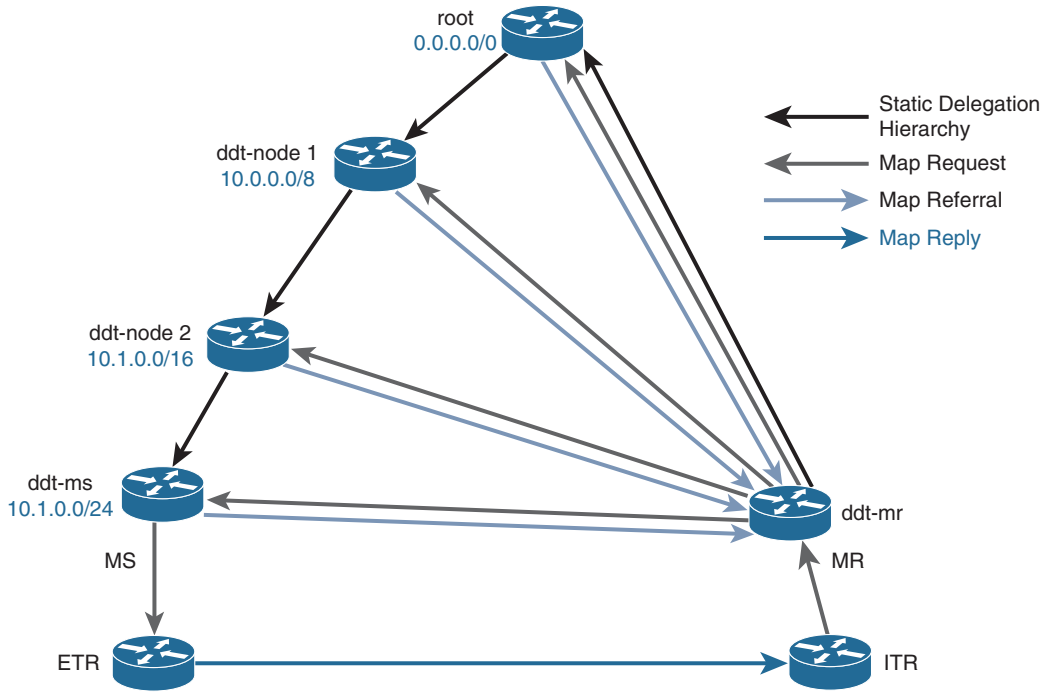



**Figure 2-7** *Delegated Database Tree (DDT)*

## An Asset-Controlled Mapping Database

The LISP mapping database naturally embodies the declarative model of object 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 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 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 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 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.

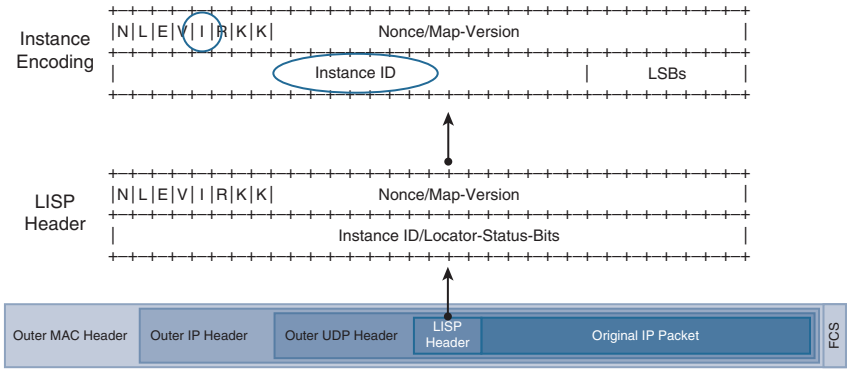



**Figure 2-8** *LISP Header with Instance-ID*

## 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 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.

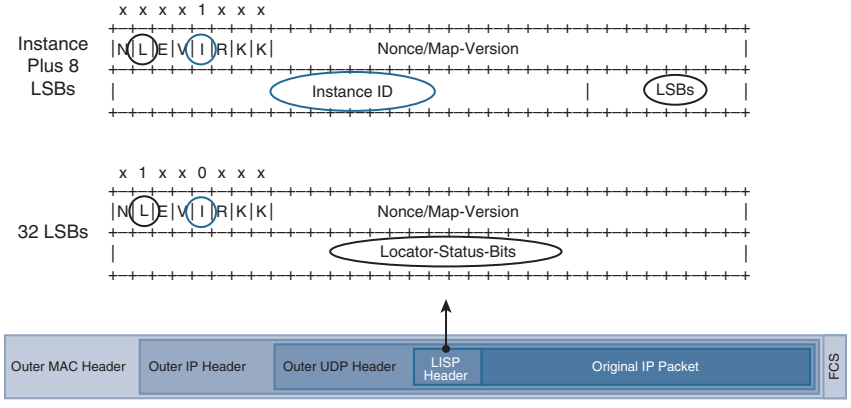



**Figure 2-9** *LISP Header Locator Status Bits*

## 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 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.

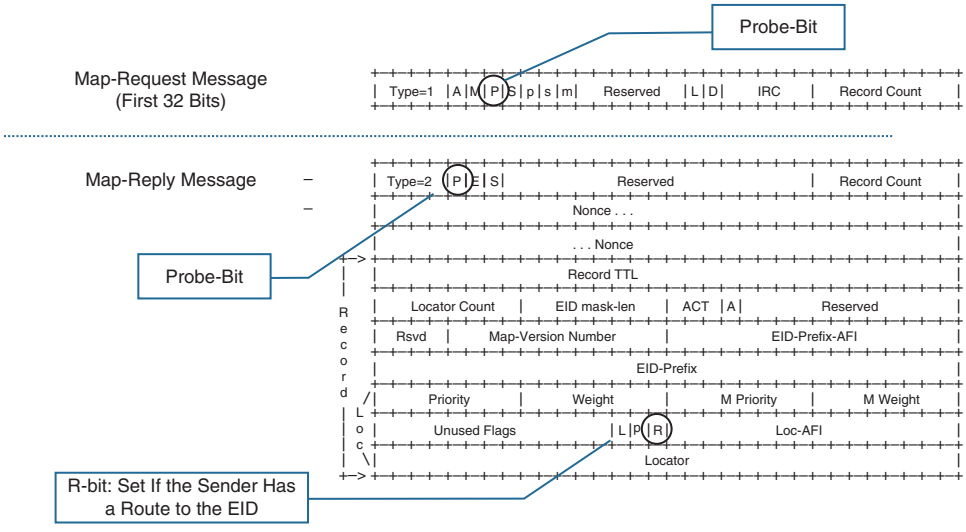



**Figure 2-10** *LISP Message Fields Relevant to RLOC Probes*

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

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 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.

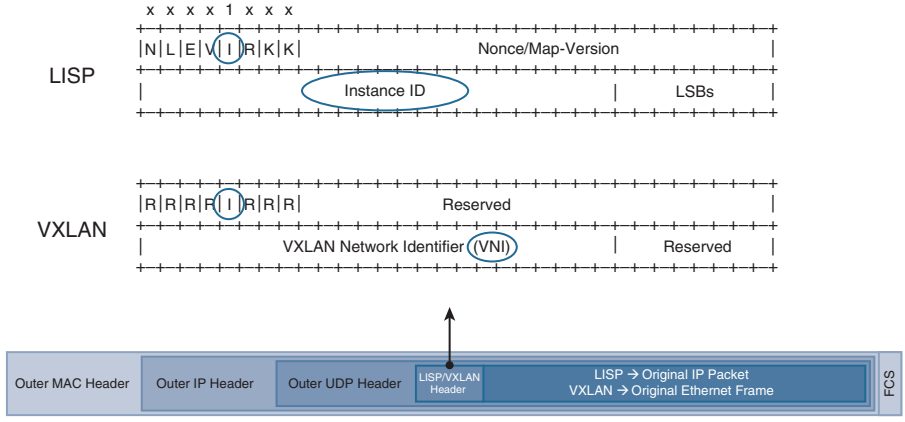



**Figure 2-11** *LISP and VXLAN Headers Compared*

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 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 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.

*This page intentionally left blank*

# Index

## Numbers


**3GPP (3rd Generation Partnership Project), 150–151**


## A


**access control lists (ACLs), 52–53**

**access networks**

applications, 92–97

*air-to-ground networking, 95–96*

*campus dormitory networking, 94*

*geo-location, 96–97*

*home networking, 93–94*

*network design normalization, 92–93*

in mobile computing

*endpoints in, 81*

*external network connections, 85–87*

*network design, 81–87*

situational policy, 92

for WANs, 72–76

zero configuration networking, 91–92

**ACLs (access control lists), 52–53**

**addresses. *See* Internet addresses; IP addresses**

**Aeronautical Telecommunications Network (ATN) use case, 148–150**

**aggregation of sensor data, 99–102**

**air-to-ground networking, 95–96**

**ALT topology, 20**

**anonymity, 129–130**

**antireplay protection, 127**

**applications**

for access networks, 92–97

*air-to-ground networking, 95–96*

*campus dormitory networking, 94*

*geo-location, 96–97*

*home networking, 93–94*

*network design normalization, 92–93*

cloud-ready, 37

multitiered, 34–37

**architecture of LISP, 9–30**
data plane, 23–30
*alternative formats, 27–29*
*confidentiality and authentication, 27*
*locator status validation, 25–26*
*NAT traversal, 29–30*
*path reliability, 26–27*
*segmentation, 24–25*
*tunnel entropy, 24*
declarative object model, 21–22
demand-based routing and caching, 12–14
extensibility, 22–23
location-identity separation, 9–11
map and encapsulate, 11–12
Mapping Database System, 17–20
proxy tunnel routers, 15–17
tunnel routers, 14–15
**assurance loop, control plane integration into, 116–117**
**asymmetric encryption, 117–119**
**ATN (Aeronautical Telecommunications Network) use case, 148–150**
**attack surfaces, 105–106**
**authentication, 27, 126–129.** ***See also*** **cryptography**
**authorization, 127–129**

# B

**Bacon, Sir Francis, 103**
**blockchain network use case, 154–155**
**bot-nets, 105–106**

# C

**campus dormitory networking, 94**
**cellular network use case, 150–153**
**Cisco Campus Fabric, group-based policies in, 52–53**
**cloud services**
hybrid clouds, 54–56
private clouds, 54
public clouds, 54
for WANs, 76
**cloud-ready applications, 37**
**Cocks, Clifford, 121**
**co-locations**
for hybrid cloud connectivity, 54–55
security policy enforcement, 56
**confidentiality, 27, 70**
**connection integrity in WANs, 70**
**connectivity of data centers**
in hybrid clouds, 54–56
mobility in, 42–46
optimizing, 39–41
**containers, 33, 37**
**control plane**
access network design, 84
declarative object model, 21–22
demand-based routing and caching, 12–14
extensibility, 22–23
integration into assurance loop, 116–117
macro-segmentation, 110
map and encapsulate, 11–12
Mapping Database System, 17–20
micro-segmentation, 112–113
process-level segmentation, 115–116
proxy tunnel routers, 15–17

security of, 123–129
*ECDSA authentication and authorization, 127–129*
*LISP-SEC, 124–127*
security policy enforcement, 51–54
segmentation, 49–50
switching fabrics, 37–38
tunnel routers, 14–15
**convergence optimization, 138–143**
predictive RLOCs, 141–143
publish/subscriber model, 139–141
redirection, 138–139
**CP/CMS operating system, 32**
**cryptography, 56, 117**
integrated key exchange, 120–123
in Internet of Things, 98
process-level segmentation, 116
public-key cryptography, 117–119
of sensor data, 100–101
symmetric cryptography, 119–120
in WANs, 70

# D

**data center trends, 31–56**
hybrid clouds, 54–56
mobility, 42–46
optimizing connectivity, 39–41
security policy enforcement, 51–54
segmentation, 46–48
*control plane segmentation, 49–50*
*data plane segmentation, 50*
*device segmentation, 48–49*
*extranet VPNs, 50–51*
switching fabrics, 37–38
virtualization
*history of, 31–33*
*multitiered applications and, 34–37*
**data plane, 23–30**
alternative formats, 27–29
confidentiality and authentication, 27
locator status validation, 25–26
macro-segmentation, 110
micro-segmentation, 113
NAT traversal, 29–30
path reliability, 26–27
segmentation, 24–25, 50
tunnel entropy, 24
**DDoS (distributed denial of service) attacks, 127**
**DDT (Delegated Database Tree), 20, 72–73, 75–76**
**declarative object model, 21–22**
**Deep Blue, 79–80**
**demand-based routing and caching, DNS analogy, 12–14**
**demilitarized zone (DMZ), 106–108**
**denial of service (DoS) attacks, 127**
**device mobility, 2**
**device segmentation, 48–49**
**DH (Diffie-Hellman) key exchange, 121–123**
**Diffie, Whitfield, 121**
**distributed denial of service (DDoS) attacks, 127**
**DMVPN (Dynamic Multipoint VPNs), 59**
**DMZ (demilitarized zone), 106–108**
**DNS (Domain Name System), analogy for demand-based routing and caching, 12–14**
**dormitory networking, 94**
**DoS (denial of service) attacks, 127**

# E

**ECDSA (Elliptic Curve Digital Signature Algorithm), 127–129**
**echo nonce, 124**
**egress tunnel routers (ETRs).** ***See*** **ETRs (egress tunnel routers)**
**EID-prefix overclaim protection, 127**
**EIDs (endpoint IDs), 4–5**
demand-based routing and caching, 12–14
ECDSA authentication and authorization, 127–129
ephemeral EIDs, 129–130
extensibility, 23
external network connections, 86–87
in federated network model, 75
geo-location, 96–97
in hybrid WANs, 62–64, 65
Instance-IDs in, 48
ITRs and, 12
mobility, 131, 132–136
path reliability, 26–27
PETRs and, 16–17
PITRs and, 16
private addresses, 30
role in LISP, 6, 7–8, 10–11
with RTRs, 74–75
security of, 123–124
security policy enforcement, 51–54
segmentation, 24–25
service insertion, 69
situational policy, 92
**Elliptic Curve Digital Signature Algorithm (ECDSA), 127–129**
**Ellis, James H., 121**
**encapsulation**
confidentiality and authentication with, 27
map and encapsulate functions, 11–12
VXLAN, 27–29
**encryption.** ***See*** **cryptography**
**endpoint IDs (EIDs).** ***See*** **EIDs (endpoint IDs)**
**endpoints**
high density, 152
micro-segmentation, 111–113
in mobile computing, 81
xTR functionality on, 102–103
**ephemeral EIDs, 129–130**
**ETRs (egress tunnel routers), 12, 15.** ***See also*** **xTRs (ingress/egress tunnel routers)**
cryptography in, 121–123
in declarative object model, 22
ECDSA authentication and authorization, 127–129
security of, 123–124
tunnel entropy, 24
**explicit locator path (ELP), 46**
**extensibility of LISP, 22–23**
**external network connections, 85–87**
**extranet VPNs, 50–51, 71**

# F

**federated network design, 73–76**
**firewalls**
DMZ (demilitarized zone), 106–108
micro-segmentation, 113
in multitiered applications, 34–35
security policy enforcement, 53
in WANs, 76

**FMC (fixed-mobile convergence) multihoming, 153**
**Frame Relay, 58**

## G

**geo-location, 96–97**
**GETVPN (Group Encrypted Transport VPN), 59, 70**
**group-based policies in Cisco Campus Fabric, 52–53**

## H

**Hellman, Martin, 121**
**high rate mobility use case, 143–148**
**history**
of the Internet, 1–2
of LISP, 5–6
of virtualization, 31–33
of WANs, 57–60
**home networking, 93–94**
**hub-and-spoke topologies in WANs, 57, 67**
**human addresses, 2–3**
**hybrid clouds, 54–56**
**hybrid WANs, 60–65**

## I

**IAB (Internet Architecture Board), 6**
**IANA (Internet Assigned Numbers Authority), 3**
**identity**
in IP addresses, 3–5
separation from location, 9–11
**IETF (Internet Engineering Task Force), 6**
**Information-Centric Networking research group, 23**
**ingress tunnel routers (ITRs). *See* ITRs (ingress tunnel routers)**
**inner-headers, 24**
**Instance-IDs, 24–25, 47, 48**
**integrated key exchange, 120–123**
**integrity. *See also* cryptography**
of EID-prefixes, 126–127
in Internet of Things, 98
**Internet, history of, 1–2**
**Internet addresses, 2–3**
**Internet Architecture Board (IAB), 6**
**Internet Assigned Numbers Authority (IANA), 3**
**Internet Engineering Task Force (IETF), 6**
**Internet of Things (IoT), 79**
lightbulbs in, 103–104
purpose of, 97–98
security and integrity in, 98
sensor data aggregation, 99–102
xTR functionality on endpoints, 102–103
**Internet Research Task Force (IRTF), 6, 23**
**Internet traffic, access network connections for, 85**
**IP addresses**
anonymity, 129–130
EIDs (endpoint IDs), 4–5
location and identity in, 3–5
mapping to Internet addresses, 2–3
mobility of, 80, 87–88
RLOCs (routing locators), 4–5
telephone number analogy, 2–3
zero configuration networking, 91
**IPsec VPNs, 59**

**IRTF (Internet Research Task Force), 6, 23**
**ITRs (ingress tunnel routers), 12, 14–15.** ***See also*** **xTRs (ingress/egress tunnel routers)**
cryptography in, 121–123
demand-based routing and caching, 13–14
ECDSA authentication and authorization, 127–129
in hybrid WANs, 64–65
LISP mobile node, 137–138
security of, 124
tunnel entropy, 24

# K

**Kasparov, Garry, 79–80**
**key exchange, 120–123**

# L

**latency, 40, 152**
**Layer 2 access network design, 82**
**LCAF (LISP Canonical Address Format), 23**
**lightbulbs in Internet of Things (IoT), 103–104**
**LISP (Locator/ID Separation Protocol)**
advantages of, 5
architecture, 9–30
*data plane, 23–30*
*declarative object model, 21–22*
*demand-based routing and caching, 12–14*
*extensibility, 22–23*
*location-identity separation, 9–11*
*map and encapsulate, 11–12*
*Mapping Database System, 17–20*
*proxy tunnel routers, 15–17*
*tunnel routers, 14–15*
history of, 5–6
as open source, 6
specifications and implementations, 6
use cases, 7–8
**LISP Beta Network, 6**
**LISP Canonical Address Format (LCAF), 23**
**LISP cryptography, 119–120, 121–123**
**LISP mobile node, 131, 136–138**
**Lisp programming language, 32**
**LISP-SEC, 124–127**
**load balancers**
in multitiered applications, 34–35
process-level segmentation, 116
**location**
in IP addresses, 3–5
separation from identity, 9–11
**locator status validation, 25–26**
**Locator/ID Separation Protocol.** ***See*** **LISP (Locator/ID Separation Protocol)**
**logical topologies**
process-level segmentation, 113–114
in WANs, 67–70

# M

**macro-segmentation, 109–111**
**malware, 105–106**
**manageability of WANs, 76–77**
**map and encapsulate, 11–12**
**Map-Caches, 14, 135**

**Map-Notify, 134**
**mapping**
Internet and IP addresses, 2–3
map and encapsulate functions, 11–12
**Mapping Database System, 8, 13, 17–20**
DDT model, 72–73
declarative object model, 21–22
ETRs and, 15
in federated network model, 75
ITRs and, 14–15
logical WAN topologies, 67–68
PETRs and, 16–17
with RTRs, 74–75
security of, 123–124
security policy enforcement, 51–54
**Map-Referrals, 20**
**Map-Registers, 15, 17–18**
**Map-Replies, 15**
**Map-Requests, 14, 15, 18**
**Map-Resolver (MR), 17–20**
**Map-Server (MS), 17–20**
**Massachusetts Institute of Technology (MIT), 31–32**
**McCarthy, John, 32**
**media broadcasting mobility use case, 153–154**
**micro-segmentation, 109, 111–113**
**micro-services, 37**
**MIT (Massachusetts Institute of Technology), 31–32**
**mobile computing, 79–80**
access networks in
*endpoints in, 81*
*external network connections, 85–87*
*network design, 81–87*
number and power of devices, 79–80
wireless integration and, 87–90
**mobility, 131–155**
convergence optimization, 138–143
*predictive RLOCs, 141–143*
*publish/subscriber model, 139–141*
*redirection, 138–139*
in data center, 42–46
EID mobility, 131, 132–136
of IP addresses, 80, 87–88
LISP mobile node, 131, 136–138
models for, 131–132
use cases
*Aeronautical Telecommunications Network (ATN), 148–150*
*blockchain network, 154–155*
*high rate mobility, 143–148*
*media broadcasting, 153–154*
*next-generation cellular networks, 150–153*
**MPLS (Multiprotocol Label Switching), 58–59**
**MS/MR nodes, 19–20, 84**
**MultiCS, 31–32**
**multihoming**
FMC (fixed-mobile convergence) multihoming, 153
with hybrid WANs, 60–65
**multisite access networks**
in mobile computing, 84–85
in sensor network design, 102
for WANs, 72–76
**multitiered applications, virtualization and, 34–37**

# N

**NAT (Network Address Translation) traversal, 29–30**
**network design**
in mobile computing, 81–87
normalization of, 92–93
**Network of Things.** ***See*** **Internet of Things (IoT)**
**network slicing, 151–152**
*The New Atlantis* **(Bacon), 103**
**next-generation cellular network use case, 150–153**

# O

**Open Overlay Router project, 138**
**open source, LISP as, 6**
**optimizing**
data center connectivity, 39–41
mobility convergence, 138–143
*predictive RLOCs, 141–143*
*publish/subscriber model, 139–141*
*redirection, 138–139*
**origin authentication, 126–127**
**OTT (over-the-top) WANs, 59–60**
**outer-headers, 24**
**overclaim attacks, 127**
**overlays, 11–12**
access network design, 83–84
for hybrid cloud connectivity, 55
process-level segmentation, 114–115
security policy enforcement, 56
switching fabrics, 37–38
**over-the-top (OTT) WANs, 59–60**

# P

**path reliability, 26–27**
**PBR (policy-based routing), 64–65**
**peer-to-peer connectivity in WANs, 67–70**
**PETRs (proxy egress tunnel routers), 16–17**
**PITRs (proxy ingress tunnel routers), 16**
**point-to-point WANs, 58**
**policy enforcement in data center, 51–54**
**policy-based routing (PBR), 64–65**
**predictive RLOCs, 141–143**
**privacy.** ***See*** **cryptography**
**private addresses in EIDs, 30**
**private clouds, 54**
**process-level segmentation, 109, 113–116**
**Project MAC, 31–32**
**proxy egress tunnel routers (PETRs), 16–17**
**proxy ingress tunnel routers (PITRs), 16**
**proxy tunnel routers, 15–17**
**public clouds, 54**
**public-key cryptography, 117–119**
**publish/subscriber model, 139–141**

# R

**redirection, 138–139**
**re-encapsulating tunnel routers (RTRs), 74–75**
**RLOC-probing, 26**
**RLOCs (routing locators), 4–5**
extensibility, 23

external network connections, 86–87
geo-location, 96–97
in hybrid WANs, 62–64
NAT traversal, 29–30
path reliability, 26–27
PETRs and, 16–17
PITRs and, 16
role in LISP, 6, 7–8, 10–11
service insertion, 69
situational policy, 92
status validation, 25–26
**routed access network design, 82**
**routes, defined, 5**
**routing locators (RLOCs). *See* RLOCs (routing locators)**
**RTRs (re-encapsulating tunnel routers), 74–75**

# S

**SaaS (software as a service) model, 64–65**
**scalable group ACL (SGACL), 52–53**
**scalable group tags (SGTs), 52–53**
**scaling WANs, 65–67**
**SD-WAN (software-defined WAN), 59–60**
hybrid WANs, 60–65
logical topologies, 67–70
manageability, 76–77
multisite access networks, 72–76
scaling, 65–67
security, 70
segmentation, 71
**security, 105–130. *See also* cryptography**
anonymity, 129–130
attack surfaces, 105–106
bot-nets, 105–106
of control plane, 123–129
*ECDSA authentication and authorization, 127–129*
*LISP-SEC, 124–127*
DMZ (demilitarized zone), 106–108
in Internet of Things, 98
malware, 105–106
of next-generation cellular network, 153
segmentation, 106–109
*control plane integration into assurance loop, 116–117*
*macro-segmentation, 109–111*
*micro-segmentation, 111–113*
*process-level segmentation, 113–116*
*traffic steering, 117*
of sensor data, 100–101
virtual perimeter, 106–109
**security policy enforcement**
in cloud services, 56
control plane integration into assurance loop, 116–117
in data center, 51–54
in WANs, 70
**segmentation, 24–25, 46–48**
advantages of, 90–91
control plane segmentation, 49–50
data plane segmentation, 50
device segmentation, 48–49
extranet VPNs, 50–51
as security policy, 106–109
*control plane integration into assurance loop, 116–117*
*macro-segmentation, 109–111*

*micro-segmentation, 111–113*
*process-level segmentation, 113–116*
*traffic steering, 117*
in WANs, 71
**sensor data aggregation, 99–102**
**service chaining, 117**
**Service Function Chaining workgroup, 54**
**service insertion in WANs, 67–70, 76**
**SGACL (scalable group ACL), 52–53**
**SGTs (scalable group tags), 52–53**
**shared computing, history of, 31–33**
**situational policy, 92**
**software as a service (SaaS) model, 64–65**
**software-defined WAN (SD-WAN), 59–60**
hybrid WANs, 60–65
logical topologies, 67–70
manageability, 76–77
multisite access networks, 72–76
scaling, 65–67
security, 70
segmentation, 71
**subnets, avoiding, 42–46**
**switching fabrics, 37–38, 54–55**
**symmetric cryptography, 119–120**

## T

**TCP/IP (Transmission Control Protocol/Internet Protocol), 1–2**
**telephone number analogy for IP addresses, 2–3**
**traffic steering, 117**
**trust relationships in declarative object model, 21–22**
**tunnel entropy, 24**
**tunnel routers, 12. *See also* ETRs (egress tunnel routers); ITRs (ingress tunnel routers); xTRs (ingress/egress tunnel routers)**
proxy tunnel routers, 15–17
segmentation, 111

## U

**ultra-low latency, 152**
**underlays, 11**
**use cases**
for LISP, 7–8
for mobility
*Aeronautical Telecommunications Network (ATN), 148–150*
*blockchain network, 154–155*
*high rate mobility, 143–148*
*media broadcasting, 153–154*
*next-generation cellular networks, 150–153*

## V

**VDI (virtual desktop infrastructure), 32**
**virtual extensible LAN (VXLAN), 36–37, 47–48**
**virtual perimeter, 106–109**
**virtualization**
history of, 31–33
multitiered applications and, 34–37
**VLAN stitching, 34–36**
**VMs (virtual machines), 33**
**VPNs (Virtual Private Networks), 24–25**

extranet VPNs, 50–51
macro-segmentation, 109–111
MPLS VPNs, 58–59
in WANs, 71
**VXLAN (virtual extensible LAN), 36–37, 47–48**
**VXLAN encapsulation, 27–29**

# W

**WANs (wide-area networks), 57–77**
access network connections for, 85–87
history of, 57–60
hybrid WANs, 60–65
logical topologies, 67–70
manageability, 76–77
multisite access networks, 72–76
scaling, 65–67
security, 70
segmentation, 71
**Williamson, Malcolm J., 121**
**wireless integration, mobile computing and, 87–90**

# X

**xTRs (ingress/egress tunnel routers), 12, 14**
data center mobility, 42–46
ECDSA authentication and authorization, 127–129
EID mobility, 132–136
external network connections, 86–87
functionality on endpoints, 102–103
macro-segmentation, 111
multihoming with hybrid WANs, 60–65
sensor data aggregation, 100

# Z

**zero configuration networking, 91–92**