Using TRILL and FabricPath

Sanjay K. Hooda
Shyam Kapadia
Padmanabhan Krishnan

ciscopress.com

Using TRILL, FabricPath, and VXLAN

Sanjay Hooda
Shyam Kapadia
Padmanabhan Krishnan

Cisco Press

800 East 96th Street

Indianapolis, IN 46240

Using TRILL, FabricPath, and VXLAN

Designing Massively Scalable Data Centers with Overlays

Sanjay Hooda
Shyam Kapadia
Padmanabhan Krishnan

Published by:
Cisco Press
800 East 96th Street
Indianapolis, IN 46240 USA

Printed in the United States of America

First Printing January 2014

Library of Congress Control Number: 2013957519

ISBN-13: 978-1-58714-393-9

ISBN-10: 1-58714-393-3

Warning and Disclaimer

This book is designed to provide information about network security. 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 international@pearsoned.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 e-mail at feedback@ciscopress.com. Please make sure to include the book title and ISBN in your message.

We greatly appreciate your assistance.

Publisher: Paul Boger

Associate Publisher: Dave Dusthimer

Development Editor: Eleanor C. Bru

Managing Editor: Sandra Schroeder

Project Editor: Seth Kerney

Editorial Assistant: Vanessa Evans

Cover Designer: Mark Shirar

Business Operation Manager, Cisco Press: Jan Cornelssen

Executive Editor: Brett Bartow

Copy Editor: Apostrophe Editing Services

Technical Editors: Narbik Kocharians, Ryan Lindfield

Proofreader: Megan Wade-Taxter

Indexer: Tim Wright

Composition: Bronkella Publishing, LLC

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

Sanjay Hooda, CCIE No. 11737, is currently a principal engineer at Cisco, where he works with embedded systems and helps define new product architectures. His current passion is to design the next-generation campus architecture, and he is focused on simplifying the design and deployment of wired and wireless infrastructure. Over the last 17 years, Sanjay's experience spans various areas including high availability; messaging in large-scale distributed systems; Supervisory Control and Data Acquisition (SCADA); large-scale software projects; and enterprise campus and LAN,WAN, and data center network design.

Shyam Kapadia, Ph.D., is currently a technical leader in the Data Center Group at Cisco. He graduated from the University of Southern California with Ph.D. and master's degrees in computer science in 2006. His research interests broadly lie in the area of networking systems including wired, wireless, ad-hoc, vehicular, and sensor networks. He has co-authored several conference and journal publications in these areas including a book chapter in the relatively nascent area of intermittently connected wireless networks (http://anrg.usc.edu/~kapadia/publications.html).

At Cisco, for the first few years, he was an integral part of the team that delivered the next-generation Catalyst 6500 Sup 2T platform. During the past few years, he has been intrinsically involved in developing solutions for data center environments with more than 25 submitted patents in this area. Over the past 12 years, Shyam has been the speakers chair for a premiere Open Source conference, Southern California Linux Exposition (SCALE), hosted in the Los Angeles area. In his spare time, he loves watching international movies and is passionate about sports like cricket, basketball, and American football.

Padmanabhan Krishnan is a software engineer in the Data Center Group at Cisco. He joined Cisco 7 years ago and has more than 12 years of experience in various areas of networking and telecommunication. He obtained his master's degree in computer science from the University of Missouri, Kansas City, and his bachelor's degree in engineering from Madras University, India. His research work for the master's degree included Diffserv, MPLS traffic engineering, and QOS routing/Connection Admission Control in ad-hoc wireless networks.

Padmanabhan has worked in many overlay technologies in Cisco such as 802.1ah, TRILL, FabricPath, and VPLS. He was responsible for the design and development of the core infrastructure used by the forwarding drivers and many Layer 2 features in the next-generation Catalyst 6500 Sup 2T Platform. Prior to joining Cisco, Padmanabhan worked in ATM signaling and DVB-RCS, an interactive on-demand multimedia satellite communication system specification.

About the Technical Reviewers

Jeevan Sharma, **CCIE No. 11529**, is a technical marketing engineer at Brocade, where he works with the Enterprise Networking Group focusing on Enterprise Switching Business. He has more than 16 years of worldwide work experience in data center and wide area network technologies, focusing on routing, switching, security, content networking, application delivery, and WAN optimization. During this period, Jeevan has held various technical roles in which he has worked extensively with customers all around the world to help them design and implement their data center and campus networks, in addition to helping them troubleshoot their complex network issues. Working internally with engineering teams, Jeevan has been instrumental in driving several new features and product enhancements, making products and solutions work better for customers. Prior to Brocade, Jeevan worked for Riverbed Technologies, Cisco Systems, HCL Technologies, and CMC Limited. He holds a bachelor's degree in engineering and an MBA degree from Santa Clara University. In his spare time, Jeevan enjoys spending time with family and friends, hiking, playing tennis, traveling, and photography.

Dedications

Sanjay Hooda: First of all, I would like to dedicate this book to my father (Satbir Singh) for being an inspiration and support. I would like to thank my mother (Indrawati), wife (Suman), and children (Pulkit and Apoorva) for their support during the writing of the book.

Shyam Kapadia: I dedicate this book to my family, especially my wife Rakhee and my mother who have provided and continue to provide their undying love and support.

Padmanabhan Krishnan: I would like to dedicate this book to my wife Krithiga and daughter Ishana. It would not have been possible without their understanding and support in spite of all the time it took me away from them. I would also like to dedicate this book to my parents and sister for their support and encouragement in all aspects of my life.

Acknowledgments

Sanjay Hooda: First of all, I would like to thank my co-authors, Shyam Kapadia and Padmanabhan Krishnan, who have been very supportive during the course of writing. In addition, I would like to thank my great friends Muninder Singh Sambi and Sanjay Thyamagundalu. Both of them have been a source of inspiration and thought-provoking insights into various areas.

Thanks as well to Brett Bartow, Ellie Bru, and all the folks at Cisco Press for their support, patience, and high quality work.

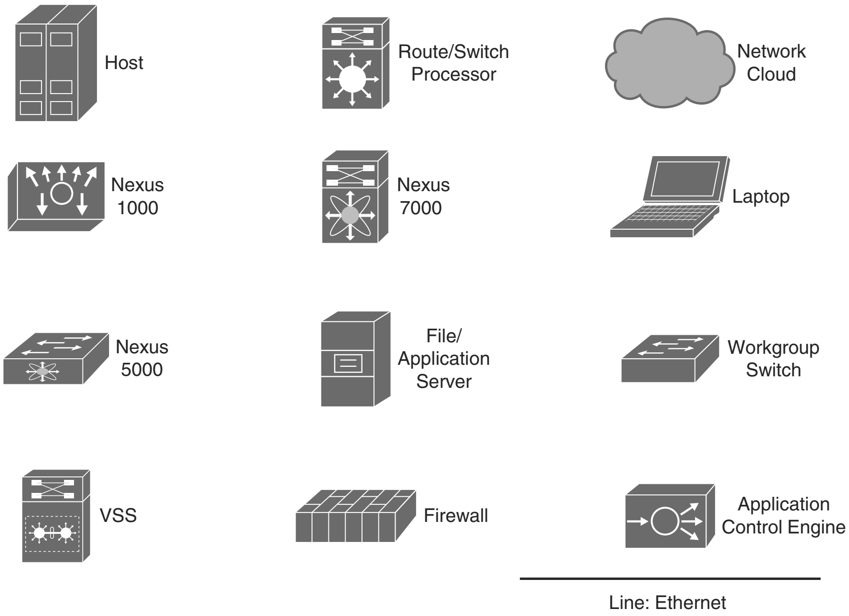
Shyam Kapadia: Special thanks to my co-authors, Padmanabhan and Sanjay, for putting in a great deal of effort in ensuring that we came up with a quality deliverable that we can all be proud of. Special acknowledgment goes to my wife Rakhee without whose help I would not have been able to complete this book on time. And last but certainly not least, special thanks to the reviewers and editors for their tremendous help and support in developing this publication.

Padmanabhan Krishnan: First and foremost, I would like to thank the editors Ellie and Brett for their helpful reviews, patience, and understanding our work-related priorities. I would like to sincerely acknowledge Rajagopalan Janakiraman for many of our technical discussions. His insights and deep technical expertise in networking helped me immensely. I would like to thank Sridhar Subramanian for sharing his expertise and materials in TRILL deployment, which were extremely helpful. A special thanks to the technical reviewer Jeevan Sharma for his thorough reviews and providing comments that added value to the chapters. I would like to express my sincere gratitude to my co-authors, Shyam and Sanjay, for their invaluable comments and support. Last, but not the least, I would like to thank my manager, Milton Xu, for giving me the opportunity to work in different overlay technologies, which gave me the needed practical exposure.

Contents at a Glance

Contents

Icons



Command Syntax Conventions

The conventions used to present command syntax in this book are the same conventions used in the IOS Command Reference. The Command Reference describes these conventions as follows:

- Boldface indicates commands and keywords that are entered literally, as shown. In actual configuration examples and output (not general command syntax), boldface indicates commands that are manually input by the user (such as a show command).
- Italics indicate arguments for which you supply actual values.
- Vertical bars (|) separate alternative, mutually exclusive elements.
- Square brackets [ ] indicate optional elements.
- Braces { } indicate a required choice.
- Braces within brackets [{ }] indicate a required choice within an optional element.

Introduction

Over the past few years, virtualization and the cloud have become exceedingly popular. The recognition that server resources including memory, CPU, and so on are severely underutilized in large data centers has led to virtualized data center deployments. Physical servers now constitute a number of virtual servers that each cater to different application needs. Architectures are sought for deployment of public clouds, private clouds, and more recently hybrid clouds. Network architects are thus faced with challenges in the design and implementation of massive scale data centers that serve these challenging requirements. To address the requirements, this book describes data center deployments using overlay technologies with emphasis on the three most popular ones: FabricPath, TRILL, and VXLAN. Data center architects are looking for innovative solutions to (a) simplify their data centers vis-à-vis, (b) retain the functionality to add new PODs without making large-scale changes to their existing DC network, and (c) ensure data center designs allow for scalability, mobility, agility, extensibility, and easier management and maintenance.

Because the book's approach is to deploy these technologies in MSDCs, the focus is to divide the chapters in the book based on understanding the overlay technology, followed by a description of some representative deployments. The final chapter is dedicated toward interconnecting two or more data centers using overlay technologies.

Goals and Methods

The goal of this book is provide a resource for readers who want to get familiar with the data center overlay technologies. The main goal is to provide a methodology for network architects and administrators to plan, design, and implement massive scale data centers using overlay technologies such as FabricPath, TRILL, and VXLAN. Readers do not have to be networking professionals or data center administrators to benefit from this book. The book is geared toward the understanding of current overlay technologies followed by their deployment. Our hope is that all readers from university students to professors to networking experts benefit from this book.

Who Should Read This Book?

This book has been written with a broad audience in mind. Consider CTOs/CIOs who want to get familiar with the overlay technologies. This book helps them by providing information on all the major overlay technology options for data centers. For the network professional with the in-depth understanding of various networking areas, this book serves as an authoritative guide explaining detailed control and data plane concepts with popular overlays, specifically, FabricPath, TRILL, and VXLAN. In addition, detailed packet flows are presented covering numerous deployment scenarios.

Regardless of your expertise or role in the IT industry, this book has a place for you; it takes various overly technology concepts and and explains them in detail. This book also provides migration guidelines as to how today's networks can move to using overlay deployments.

How This Book Is Organized

Although you could read this book cover-to-cover, it is designed to be flexible and allow you to easily move between chapters and sections of chapters to cover only the material you need. The first two chapters target the CTO/CIO–level executives and describe the need for overlays and provide a brief description of the existing overlay technology options. Chapter 3 forms the foundation for the subsequent FabricPath and TRILL chapters and describes Layer 2 IS-IS with an emphasis on the extensions for supporting Layer 2 multipath overlay schemes. Chapter 4 through Chapter 9 describes the design, innerworkings, and deployment of the most popular data center overlay technologies; namely, FabricPath, TRILL, and VXLAN.

Chapters 1 through 9 cover the following topics:

- **Chapter 1, "Need for Overlays in Massive Scale Data Centers":** This chapter describes the major requirements of massive scale data centers and the associated deployment challenges. Popular data center architectures are introduced, and the case for overlays in data center networks is firmly established.
- **Chapter 2, "Introduction to Overlay Technologies":** This chapter provides a brief survey of various overlay technologies employed in data center environments.
- **Chapter 3, "IS-IS":** This chapter provides a brief introduction to IS-IS. It ex-plains in detail the extensions that were introduced in IS-IS to support TRILL.
- **Chapter 4, "FabricPath":** This chapter introduces FabricPath, a novel Cisco overlay solution, and provides details of the architecture and innerworkings of FabricPath, both from the point of view of control plane and data plane. Detailed end-to-end packet flows are presented in a FabricPath network.
- **Chapter 5, "TRILL":** This chapter introduces TRILL, an IETF standard, and provides details of the architecture and innerworkings of TRILL. Both control and data plane aspects are described. This chapter also covers in detail the different areas of development in the TRILL community as of this writing. Detailed end-to-end packet flows are presented in a TRILL network.
- **Chapter 6, "VXLAN":** This chapter provides a detailed description of VXLAN, a popular MAC over IP/UDP overlay deployed in data center environments. Details of the VXLAN architecture are presented coupled with step-by-step packet flows covering unicast, multicast, and broadcast cases in VXLAN clouds. Both multicast as well as multicast-less VXLAN deployment options are presented.

- **Chapter 7, "FabricPath Deployment, Migration, and Troubleshooting":** This chapter covers the different deployment possibilities with FabricPath along with representative examples. Migration strategies to FabricPath including (Classical Layer 2 to FabricPath and vPC to vPC+) are covered. In addition, some common FabricPath deployment topologies are presented. The chapter concludes with a brief description of troubleshooting and monitoring tools for FabricPath networks.
- **Chapter 8, "TRILL Deployment, Migration and Troubleshooting":** This chapter explains how current data center deployments can be migrated to TRILL. Various deployment scenarios along with some case studies are explained in detail. A brief introduction to troubleshooting in TRILL networks is also provided.
- **Chapter 9, "Interoperability of Other Technologies":** This chapter describes some specific deployments where multiple overlay technologies may be employed to realize an end-to-end solution in data center environments. Three representative case studies are presented that cover both intra-DC and inter-DC deployments.

This page intentionally left blank

Chapter 2

Introduction to Overlay Technologies

This chapter covers the following objectives:

- **FabricPath:** This section starts with an introduction to FabricPath and its high-level architecture followed by frame format details and then delves into data plane operations with FabricPath. For in-depth details on FabricPath, refer to Chapter 4, "FabricPath."
- **Transparent Interconnection of Lots of Links (TRILL):** This section provides an overview of the requirements and benefits of TRILL along with the frame format and high-level data plane operations. For more details on TRILL refer to Chapter 5, "TRILL."
- **Locator/ID Separation Protocol (LISP):** This section provides an overview of LISP frame format details and LISP high-level data plane operations, and discusses LISP mobility.
- **Virtual Extensible LAN (VXLAN):** This section provides an overview of VXLAN along with frame format followed by a brief description of VXLAN operation. For more details, refer to Chapter 6, "VXLAN."
- **Network Virtualization using Generic Routing Encapsulation (NVGRE):** This section provides an overview of NVGRE along with the frame format followed by NVGRE data plane operations.
- **Overlay Transport Virtualization (OTV):** This section provides an overview of OTV followed by frame format details and data plane operations.
- **Provider Backbone Bridging (PBB):** This section provides an overview of IEEE 802.1ah followed by frame format details and data plane operations.

- **Shortest Path Bridging (SPB):** This section provides an overview of SPB (including Shortest Path Bridging VID [SPBV] and Shortest Path Bridging - MAC [SPBM]) and data plane operations.

This chapter covers the various overlay technologies, which have become extremely popular in data center and enterprise networks. Because the underlying control protocol for both FabricPath and TRILL is IS-IS[1], if you want an in-depth understanding of IS-IS, refer to Chapter 3, "IS-IS," for details on IS-IS. This chapter, in addition to providing an executive-level overview of the different overlay technologies, also enables you to get a quick grasp of each of these technologies. This chapter builds the foundation for further discussion of these technologies in subsequent chapters.

Overlay Technologies Overview

Table 2-1 gives an overview of the different overlay technologies along with their benefits.

Table 2-1 *Different Overlay Technologies Overview*

Technology	Description	Benefit
FabricPath	FabricPath is a Layer 2 technology that provides Layer 3 benefits such as multipathing to the classical Layer 2 networks by using link state protocol (IS-IS) at Layer 2. This enables the network to be free of the spanning tree protocol, thereby avoiding its pitfalls especially in a large Layer 2 topology.	Provides plug-and-play features of classical Ethernet Networks. Multipath Support (ECMP) provides high availability to Ethernet networks. Conversational MAC learning provides MAC scalability. Enables larger Layer 2 domains because it doesn't run spanning tree.
TRILL	TRILL is an IEEE standard that, like FabricPath, is a Layer 2 technology, which also provides the same Layer 3 benefits as Fabric Path to the Layer 2 networks by using the link state protocol (IS-IS) over Layer 2 networks.	Provides plug-and-play features of classical Ethernet networks. MAC-in-MAC encapsulation enables MAC address scalability in the TRILL networks.

Technology	Description	Benefit
LISP	LISP separates the location and the identifier (EID) of the network hosts thus allowing virtual machine mobility across subnet boundaries while keeping the endpoint identification (IP address for IP networks).	Optimal shortest-path routing. Support for both IPv4 and IPv6 hosts. There is a draft for Layer 2 LISP that supports MAC addresses as well. Load balancing and multi homing support.
VXLAN	Virtual Extensible LAN (VXLAN) is a LAN extension over a Layer 3 network. This encapsulation with its 24-bit segment-ID enables up to 16 million VLANs in your network.	Large number of Virtual LANs (16 million). The extension of the VXLAN across different Layer 3 networks, while enabling communication at Layer 2 enables elastic capacity extension for the cloud infrastructure. Enables VM mobility at Layer 2 across Layer 3 boundaries.
NVGRE	NVGRE, like VXLAN, is an encapsulation of a Layer 2 Ethernet Frame in IP, which enables the creation of virtualized Layer 2 subnets. With an external IP header, these virtualized Layer 2 subnets can span physical Layer 3 IP networks.	Compatible with today's data center hardware infrastructure because it doesn't require an upgrade of data center hardware because GRE support is common. Like VXLAN the Tenant Network Identifier (TNI) in the GRE frame enables 16 million logical Layer 2 networks. Enables VM mobility at Layer 2 across Layer 3 boundaries.
OTV	Overlay transport virtualization (OTV) is a Cisco-proprietary innovation in the Data Center Interconnect (DCI) space for enabling Layer 2 extension across data center sites.	OTV, being an overlay technology, is transparent to the core network and the sites. Failure boundary and site independence are preserved in OTV networks because OTV uses a control plane protocol to sync MAC addresses between sites and avoIDs any unknown unicast floods.

Technology	Description	Benefit
Shortest Path Bridging (SPB)	Like TRILL, SPB uses IS-IS to advertise topology information. SPB is an IEEE counterpart to TRILL but differs in the use of the tree structures. At the edge devices, the packets are either encapsulated in MAC-in-MAC (802.1ah) or tagged (802.1Q/802.1ad) frames.	The benefits are similar to FabricPath and TRILL networks including: ■ Multipath support (ECMP) provides high availability for Ethernet networks. ■ Failure/recovery is handled by standard IS-IS behavior.

FabricPath

The trend toward virtualization of physical servers especially in large data centers began a few years ago. VMware became the leader on the server virtualization front; the benefits from server virtualization and commodities of scale led to the emergence of "mega data centers" hosting applications running on tens of thousands of servers. This required support for distributed applications at a large scale and having the flexibility to provision them in different zones of data centers. This necessitated the need to develop a scalable and resilient Layer 2 fabric enabling any-to-any communication. Cisco pioneered the development of FabricPath to meet these new demands. FabricPath provides a highly scalable Layer 2 fabric with a required level of simplicity, resiliency, and flexibility.

FabricPath Requirements

The evolution of large data centers with more than 1000 servers, with a design that enables scaling in size and computing capacity aka Massively Scalable Data Centers (MSDC) and virtualization technologies, has led to the need for large Layer 2 domains. The well-known Spanning Tree Protocol (STP) on which Layer 2 switching relies introduces some limitations, which led to the evolution of technologies such as TRILL and FabricPath. Before delving into further details, you need to consider the limitations of current Layer 2 networks based on STP, which were the drivers for FabricPath:

- **No multipathing support:** STP creates loop-free topologies in the Layer 2 networks by blocking redundant paths. To achieve this, STP uses the well-known root election process. After the root is elected, all the other switches build shortest paths to the root switch and block other ports. This yields a loop-free Layer 2 network topology. The side effect of this is that all redundant paths are blocked in the Layer 2 network. Although some enhancements were done specially with the use of Per VLAN Spanning Tree Protocol (PVSTP), PVST enables per VLAN load balancing, but it also suffers from multipathing support limitations.

- **STP leads to inefficient path selection:** As the shortest path is chosen for the root bridge, the available path between switches depends upon the location of the root bridge. Hence, the selected path is not necessarily a shortest path between the switches. As an example, consider two access switches that connect to the distribution and to each other. Now if the distribution switch is the STP root bridge, the link between the two access switches is blocked, and all traffic between the two access layer switches takes the suboptimal path through the distribution switch.
- **Unavailability of features like Time-To-Live (TTL):** The Layer 2 packet header doesn't have a TTL field. This can lead to a network meltdown in switched networks because a forwarding loop can cause a broadcast packet to exponentially duplicate thereby consuming excessive network resources.
- **MAC address scalability:** Nonhierarchical flat addressing of Ethernet MAC addressing leads to limited scalability as MAC address summarization becomes impossible. Also, all the MAC addresses are essentially populated in all switches in the Layer 2 network leading to large requirements in the Layer 2 table sizes.

These shortcomings of Layer 2 networks are resolved by the Layer 3 routing protocols, which provide multipathing and efficient shortest path among all nodes in the network without any limitations. Although the Layer 3 network design solves these issues, it has the side effect of making the network design static. As the static network design limits the size of the Layer 2 domain, it limits the use of virtualization technologies. FabricPath marries the two technologies to provide flexibility of Layer 2 networks and scaling of the Layer 3 networks.

FabricPath Benefits

FabricPath is a new technology that enables the data center architects and administrators to design and implement a scalable Layer 2 fabric. FabricPath offers the following benefits:

- **Preserves the plug-and-play features of classical Ethernet:** Because the configuration requirements are minimal and the administrator needs to include the interfaces belonging to the FabricPath core network, it significantly reduces the administrative effort to configure the network. FabricPath also uses a single control protocol (IS-IS) for unicast forwarding, multicast forwarding, and VLAN pruning. In addition, ping and trace route are now available in FabricPath operations, administration, and management (OAM), enabling the network administrators to debug problems in the Layer 2 FabricPath network similar to common troubleshooting techniques employed for Layer 3 networks.
- **Provides high performance using multipathing:** The N-way (more than one paths) multipathing enables the data center network architects to build large, scalable networks. It also enables network administrators to incrementally add additional devices to the existing network topology as the need arises. This enables the MSDC networks to have flat topologies, enabling the nodes to be separated by a single hop.

The N-way multipathing has an additional benefit that a single node failure just leads to a reduction by 1/Nth of the fabric bandwidth.

- **High availability:** The enhancements to Layer 2 networks with the combination of Layer 3 capabilities enables the replacement of STP, which blocks all paths except a single path, enabling multiple paths between the endpoints. This enables the network administrator to incrementally add network bandwidth as the bandwidth needs increases.
- **Forwarding efficiency:** FabricPath enables the traffic to be forwarded across the shortest path to the destination, thus reducing latency in the Layer 2 network. This is more efficient when compared to Layer 2 forwarding based on the STP.
- **Small Layer 2 table size:** Conversational MAC learning in the FabricPath solution enables selective learning of the MAC addresses based on the active flows. This reduces the need for the large MAC tables.

FabricPath Architecture

Figure 2-1 shows the high-level addressing scheme employed by FabricPath.

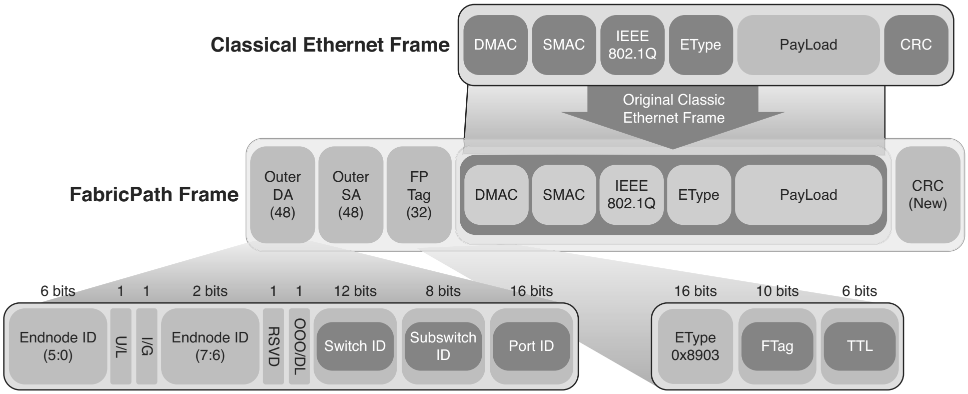


Figure 2-1 *FabricPath High-Level Architecture*

The following sections start with a brief description of the encapsulation employed by FabricPath[2] followed by a sample packet walk-through of a FabricPath network.

FabricPath Encapsulation

To forward the frames, FabricPath employs hierarchical MAC addresses that are locally assigned. FabricPath encapsulates the original Layer 2 frame with a new source and destination MAC address, a FabricPath tag, the original Layer 2 frame, and a new CRC (refer to Figure 2-1). To forward the frames in the FabricPath network, the outer source and destination MAC addresses contain a 12-bit unique identifier called a SwitchID. The SwitchID is the field used in the FabricPath core network to forward packets to the right destination switch. Chapter 4 describes each of these fields.

FabricPath Data Plane Operation

You can use Figure 2-2 as a reference to describe the high-level FabricPath data-path operation.[3]

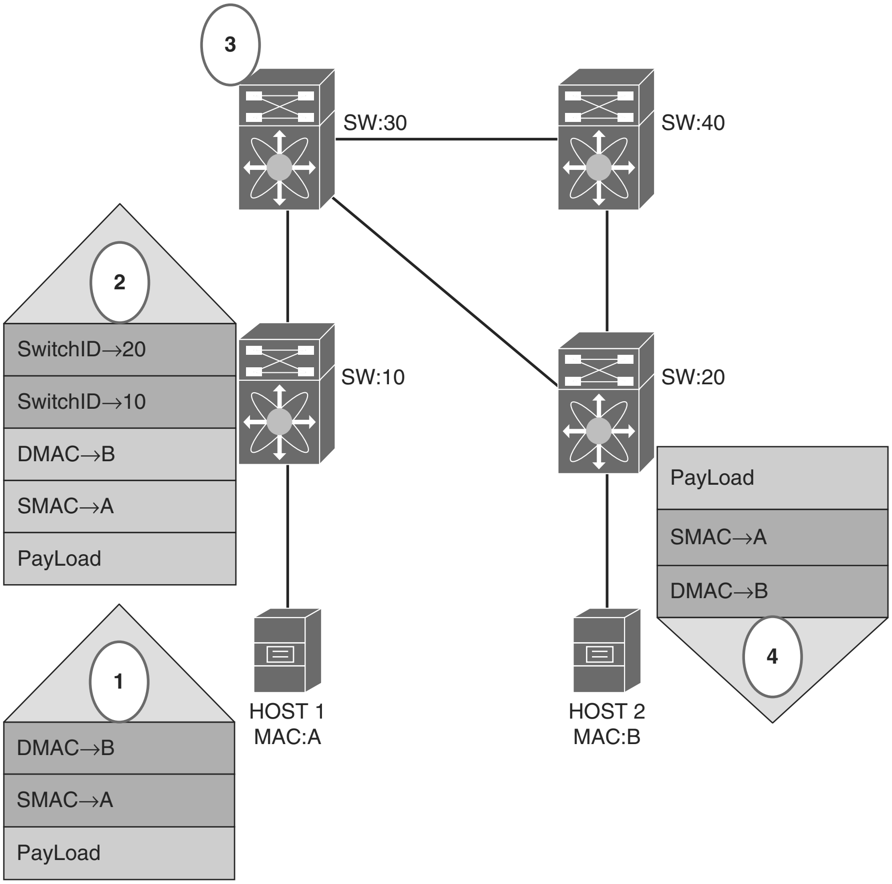


Figure 2-2 *FabricPath Data Path*

To describe the data path from Host 1 to Host 2, you can assume that all the control plane information has already been learned. Host 1 and Host 2 already know about each other's MAC addresses. The basic steps involve the encapsulation of the frame with a FabricPath header at the ingress switch, followed by switching the frame using the outer header in the FabricPath network and then finally decapsulation of the frame at the egress switch. The following steps provide more details on this operation.

1. Host 1 uses its MAC address A as a source MAC (SMAC) and sends a classical Ethernet frame, which is destined to Host 2 with a destination MAC (DMAC) address B. On receiving this frame, the ingress FabricPath switch does a standard layer lookup based on VLAN and DMAC.

2. The lookup result points to the destination SwitchID 20 as the egress switch for this frame. So the ingress switch encapsulates this frame in a FabricPath header and sends it out on an appropriate FabricPath core port. The source and destination switch IDs in the FabricPath header are set as 10 and 20, respectively.

3. The fabric path switch 30 forwards the frame based on the best path to the destination switch 20. Here there are two paths, but the best path is a directly connected link, and therefore the packet is forwarded over the directly connected interface to switch 20.

4. The destination switch 20 receives this frame. As the destination switch ID is itself, it removes the FabricPath header. On decapsulation of the frame, it uses the inner DMAC for a Layer 2 lookup and, based on the lookup result, forwards the frame toward Host 2.

TRILL

Transparent Inter-Connection of Lots of Links (TRILL) is a technology that addresses the same requirements as the FabricPath and has almost the same benefits as FabricPath. The requirements and benefits of FabricPath were given in the FabricPath section of this chapter. The chapter on TRILL discusses all the limitations of current Layer 2 networking in detail and how TRILL addresses them. TRILL, as of this writing, is an IETF standard. With the changes happening in the data center environments, the current STP has lots of disadvantages as outlined here:

- **Inefficient utilization of links:** To avoID loops in a Layer 2 network, the STP ensures that there's only one path from a source to a destination. To achieve this, many of the links in a switch are put in a blocked state so that data traffic doesn't flow through the links. With the rapID increase in server-to-server communication, referred to as east-west traffic, blocking many of the links can cause congestion in the links that are in an unblocked state. Shutting down or blocking the links in a switch reduces the value of a switch that has the capacity to host many ports capable of carrying high-bandwidth traffic. A Layer 3-like behavior is required, wherein all the links in a switch can be used and that provides a loop-free mechanism.
- **Long time to converge:** STP is not designed for topologies such as MSDC. The time taken for all the nodes in a network to go to a steady state is high. Traffic is disrupted until the steady state is reached. Whenever there is a change in the topology because of a link going up or down or when new nodes are added or removed, spanning tree recalculation results in traffic disruption. Clearly, a loop prevention mechanism is required that can scale well in an MSDC environment. Again, a Layer 3 behavior is required, wherein the routing protocol takes care of avoiding loops and can also scale to a large number of nodes.
- **Scaling the MAC table:** With the emergence of virtual machines, with each VM assigned a MAC address, the size of the Layer 2 table can grow by a big margin, especially at the core of the data center network that learns the MAC address of all

the VMs. The cost of the hardware may increase with the increase in the size of the hardware Layer 2 table. It's preferable to have a clear separation of the overlay network and the end host access network such that the core network can have a Layer 2 table whose size can be better quantified by the number of switches in the overlay network than trying to quantify the number of end host VMs in the entire network, which may not be a trivial task. If the size of the Layer 2 table at the core is less, it may result in some entries not being learned. This can result in a Layer 2 lookup miss, which can result in a flood in the network. Flooding can consume unnecessary network bandwidth and may consume the CPU resources of the server because the server may also receive the flood frames. Clearly, a tunneling protocol such as MAC-in-MAC is required so that all the core switches do not need to learn all the end host MAC addresses.

TRILL Requirements

Some of the design criteria and requirements of TRILL follow:

- **Control protocol:** TRILL uses Layer 2 IS-IS as its control protocol. The idea is to take the advantages of a Layer 3 routing protocol and at the same time maintain the simplicity of a Layer 2 network. Every node in a TRILL network is referred to as RBridge, aka Router-Bridge. Every RBridge is identified by its nickname. In other words, a nickname is the routable entity in a TRILL network, just like an IP address in an IP network. Unlike Layer 3, there are no separate protocols for unicast and multicast. The Layer 2-IS-IS protocol takes care of populating the routing table for unicast traffic, thereby ensuring multiple shortest equal cost paths (ECMPs) for all the RBridges and also creating trees for multicast traffic. Needless to say, Layer 2 IS-IS also ensures loop-free routing. But at the same time, TRILL inherits the TTL field from the Layer 3 world to ensure traffic due to intermittent loops eventually expires out.

- **Preserve plug-and-play features of classical Ethernet:** One of the main advantages of a Layer 2 network is its plug-and-play nature, and the administrator is relieved of heavy configuration unlike in a Layer 3 network. TRILL achieves this with its Dynamic Resource Allocation Protocol (DRAP), where every node derives its own nickname and the protocol ensures there's no duplicity. The configuration requirement of TRILL is minimal.

- **Layer 2 table scaling:** TRILL uses a MAC-in-MAC encapsulation, where the traffic from the host is encapsulated by the ingress RBridge. The core RBridges see only the outer MAC header, which has the MAC address of the source and destination RBridge. Consequently, the MAC table at the core RBridges will not be polluted with all the end host MAC addresses.

The following section starts with the TRILL frame format and then delves into the high-level data plane architecture:

TRILL Frame Format

To forward frames, TRILL uses a MAC-in-MAC encapsulation format, as shown in Figure 2-3. The ingress RBridge encapsulates the original Layer 2 frame with a new source and destination MAC, which are the MAC addresses of the source RBridge and the next-hop RBridge respectively; a TRILL Header, which has the Ingress and Egress nickname that identifies the source and destination RBridge, respectively; and the original Layer 2 frame with a new CRC. The incoming 802.1q or q-in-q tag needs to be preserved in the inner header. Chapter 5 covers all these fields in greater depth. Egress RBridge removes the headers added by the ingress RBridge and will forward based on the inner frame.

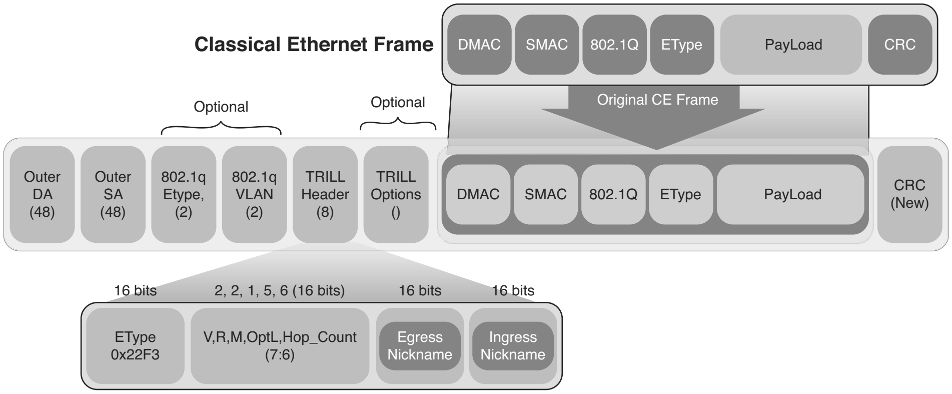


Figure 2-3 *TRILL Frame Format*

TRILL Data Plane Operation

To describe the high-level data path operation, use the network shown in Figure 2-4. By now you would have already figured out that the forwarding is similar to FabricPath.

To describe the data path from Host 1 to Host 2, assume that all the control plane information has already been learned. Host 1 and Host 2 already know about each other's MAC addresses. The basic steps involve the encapsulation of the frame with the TRILL header at the ingress RBridge, followed by switching using the TRILL header in the TRILL network and then finally decapsulation of the frame at the egress RBridge. The following steps provide more details on this operation.

1. Host 1 uses its MAC address of A as the source MAC (SMAC) and sends a classical Ethernet frame, which is destined to Host 2 with a destination MAC (DMAC) address of B. On receiving this frame, the ingress RBridge (Nickname 10) does a (VLAN, DMAC) lookup.

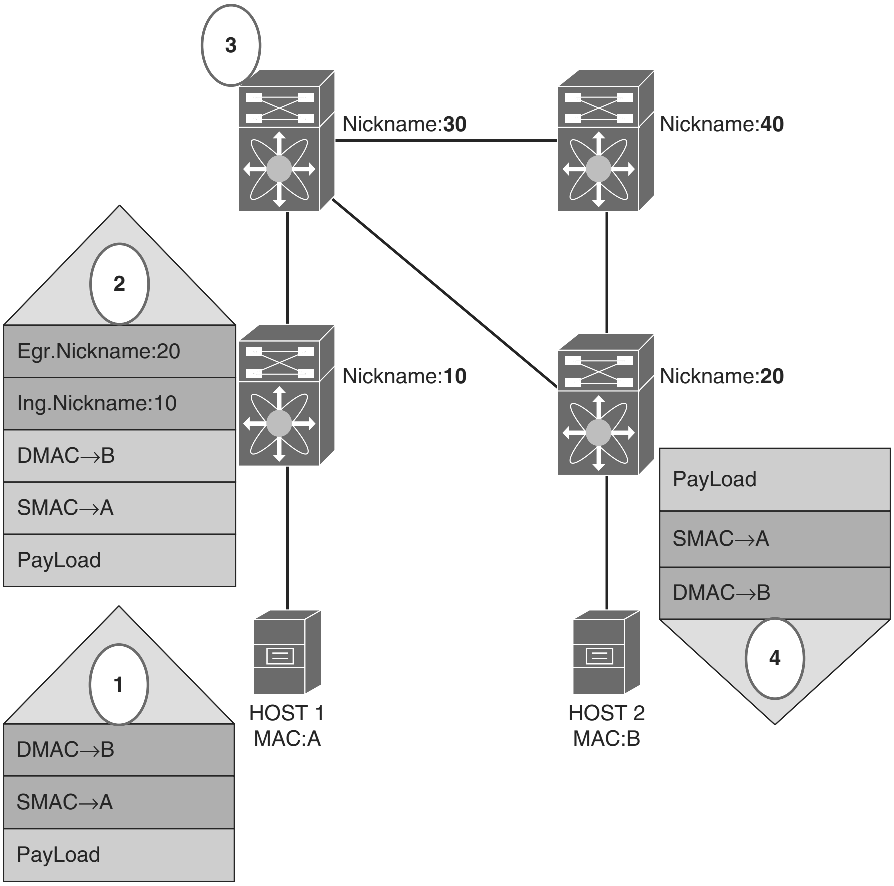


Figure 2-4 *TRILL Data Path*

2. The MAC lookup points to the destination (Nickname 20) as the egress RBridge for this Ethernet frame. So the ingress switch encapsulates this frame using the TRILL header for forwarding the frame to the TRILL core port. The source and destination nicknames are set as 10 and 20, respectively. The outer DMAC is the MAC address of the next-hop RBridge, and the outer SMAC is the MAC address of the source RBridge.

3. The core RBridge (Nickname 30 in this example) forwards the frame based on the best path to the destination RBridge Nickname 20. In this case there are two paths to reach the egress RBridge with Nickname 20, but the best path is a directly connected link; therefore, the packet is forwarded over the directly connected interface to the switch with Nickname 20. The TTL is decremented, and the outer SMAC and DMAC are rewritten with the MAC address of this RBridge and RBridge 20's MAC address. Just like regular IP routing, the TRILL header is not modified, but at each hop the outer DMAC and SMAC are rewritten along with a TTL decrement.

4. The destination RBridge 20 receives this frame. Because the incoming frame is destined to this RBridge, it removes the outer MAC and the TRILL header. It then forwards the frame to Host 2 based on the inner (DMAC and VLAN) lookup.

Locator ID/Separator Protocol

Locator ID/Separator Protocol (LISP) as the name suggests separates the location and the identifier of the network hosts, thus making it possible for virtual machines to move across subnet boundaries while retaining their IP address. LISP is composed of a network architecture and a set of protocols that enable new semantics for IP addressing by creating two namespaces:

- **Endpoint Identifiers (EIDs):** EIDs are assigned to end hosts.
- **Routing Locators (RLOCs):** RLOCs are assigned to routers that make up the global routing system.

The creation of these separate namespaces provides several advantages, including the following:

- Topologically aggregated RLOCs enable improved routing system scalability.
- IP portability.
- Easier IPv6 transition.
- IP mobility, the host EIDs can move without changing the IP address of the host or virtual machine; only the RLOC changes on a host move.

LISP integrates well into the current network infrastructure and requires no changes to the end host stack. It fosters a simple, incremental, network-based implementation with most of the deployment at the network edge devices.

LISP Frame Format

Figure 2-5 shows the various fields in the LISP header.

A LISP frame's outer encapsulation is a UDP frame where the destination and source IP addresses are the addresses of the Ingress Tunnel Router (ITR) and Egress Tunnel Router (ETR), respectively. For Layer 3 LISP, the destination UDP port number is 4341. The LISP header has the Locator reachability bits and the nonce fields.

LISP Routing

As a host transmits a packet, if the destination of the packet is in another LISP domain, it reaches the LISP ITR. The ITR maps the destination endpoint ID (EID) to an RLOC by looking up the destination in a map server. As shown in Figure 2-6, using this information the ITR encapsulates the packet with an outer header. The destination RLOC is ETR behind which the destination host exists.

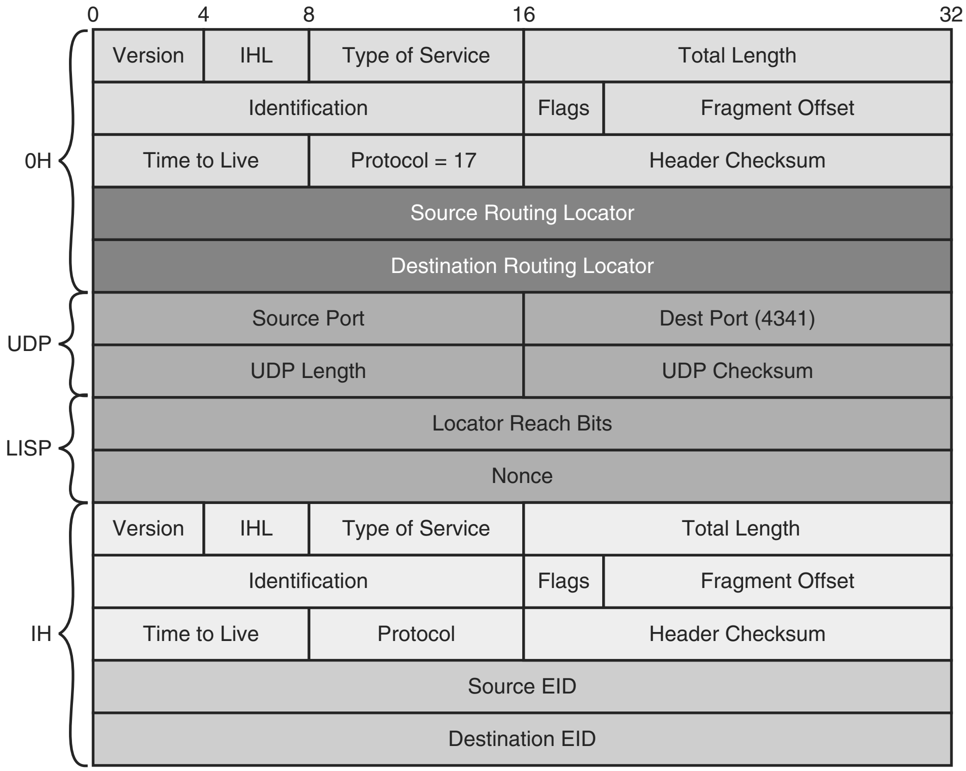


Figure 2-5 *LISP Frame Format*

When the destination ETR is known, the ITR encapsulates the packet, setting the destination address to the RLOC of the destination ETR returned by the mapping infrastructure. Refer to Figure 2-6 to see the flow of traffic in a LISP-enabled network.

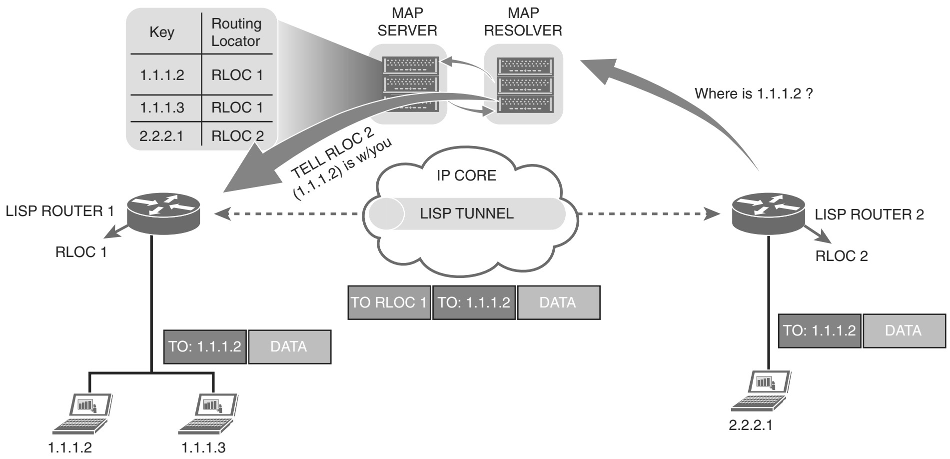


Figure 2-6 *LISP Routing*

In addition to LISP routing, the location and EID separation provides flexible and unmatched mobility for IP endpoints without any subnet boundary limitation allowing IP endpoints, regardless of their IP addresses, to be deployed anywhere. These EIDs can freely move within and across data center racks and across geographical and organizational boundaries. The LISP Mobility solution has the following characteristics:

- Optimal shortest path routing.
- Both IPv4 and IPv6 addresses are supported.
- Support for load balancing and multihoming.
- Provides a solution that is transparent to both EIDs and the core network.

By allowing IP endpoints to change location while maintaining their assigned IP address, the LISP mobility solution enables the IP endpoints to move between different subnets, while guaranteeing optimal routing to the IP endpoint.

VXLAN

Cloud service providers, specifically Infrastructure as a Service (IaaS) providers, require a network segmentation solution that supports a large number of network segments. The advent of server virtualization has increased the demand on the physical network infrastructure. As the number of VMs attached to the network increases, there is an increased demand in the number of MAC address table entries in switches. In addition, the VMs may be grouped according to their VLAN with the current limitation of number of VLANs being 4096. Server virtualization especially in service provider data center environments has exposed the limitation of a limited number of VLANs. This limitation has introduced challenges for the IP address management.

In addition, VM mobility requires a Layer 2 extension from the old physical host to the new host where the VM is moving. As the data center architects strive to remove this limitation of native Layer 2 extension, they are looking for solutions that don't bind them to physical infrastructure.

The network segmentation, server virtualization, and Layer 2 VM Mobility require an overlay that can carry Layer 2 (MAC) frames. VXLAN is a Layer 2 overlay mechanism that addresses these needs. VXLAN stands for the Virtual eXtensible Local Area Network and provides a way to implement a large number of virtual Layer 2 networks on top of today's networking and virtualization infrastructure.

VXLAN encapsulates a MAC frame within a User Datagram Protocol packet (MAC-in-UDP). A 24-bit virtual segment identifier in the form of a VXLAN ID (VNI) is part of the VXLAN header that enables the VLANs to scale up to 16 million. In addition, the UDP encapsulation enables each VLAN to span across a Layer 3 routed network.

In its simplest form, for broadcast, multicast, and unknown unicast traffic, VXLAN employs IP multicast. After a virtual machine joins a VXLAN segment, the physical host on which the VM resides joins the multicast group associated with that segment.

VXLAN uses a multicast tree to send broadcast/multicast/unknown-unicast packets to all the servers in the same multicast group. When the learning is complete, the unicast packets are encapsulated and sent directly to the destination physical host. On each virtualized host, there resides an entity called the Virtual Tunnel End-Point (VTEP). This entity is responsible for suitably encapsulating and decapsulating the VXLAN header as it is sent to or received from the upstream network.

VXLAN Frame Format

Figure 2-7 shows the VXLAN frame format.[4] Each of the components of the frame is also described:

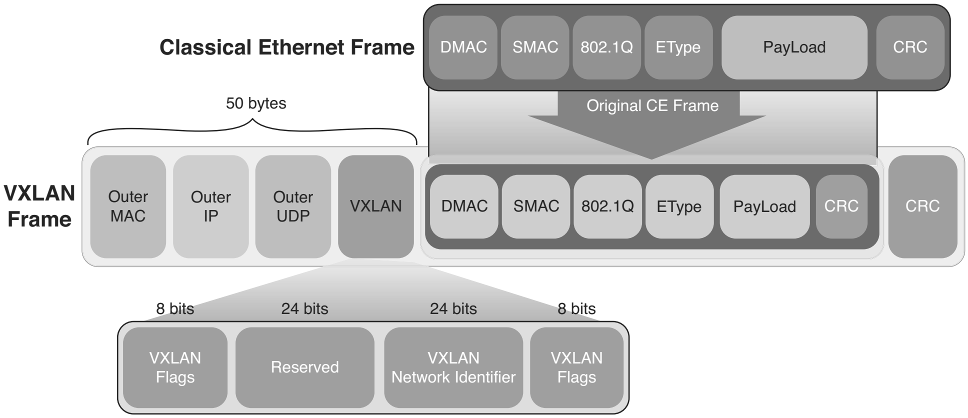


Figure 2-7 *VXLAN Frame Format*

- Outer Ethernet header
- Outer IP header
- Outer UDP header
- VXLAN header
- Inner Ethernet header

The different fields in the VXLAN header,[5] of size 8 bytes, include 8 bits of VXLAN flags, 24 bits of VXLAN identifier (VNI), and reserved flags.

- **VXLAN flags:** Reserved bits set to 0 except bit 3, the I bit, which is set to 1 to indicate a valID VNI
- **VNI:** 24-bit field that is the VXLAN network identifier
- **Reserved:** A set of fields, 24 bits and 8 bits, that are reserved and set to zero

VXLAN Data Path Operation

Figure 2-8 shows a sample VXLAN packet flow;[6] now consider a packet being sent by a virtual machine on one of its vNICs. As the virtual switch (that is, vSwitch) receives the packet from the vNIC, it knows the VXLAN Network ID (VNI) for this packet. The vSwitch performs two lookups on the packet:

- The vSwitch uses the ingress interface (vNIC) to determine which VNI the packet belongs to.
- vSwitch does a (VNI and DMAC) lookup.
- If the lookup is a HIT and the packet is destined to a remote VM, the packet is suitably encapsulated by the source VTEP with a VXLAN header with the Destination IP (DIP) set to the physical host on which the destination VM resides.
- If the lookup is a MISS, the packet is VXLAN encapsulated, but the DIP is set to the multicast group associated with the corresponding VNI.
- The vSwitch then does a second lookup, this time on the encapsulated packet, and dispatches the packet toward the IP core that delivers the packet to the DIP in the overlay header.
- VXLAN header decapsulation is performed at the destination VTEPs where the inner SMAC is learned against the source VTEP's IP in the overlay header and the packet is switched as per the (VNI, inner_DMAC) lookup.

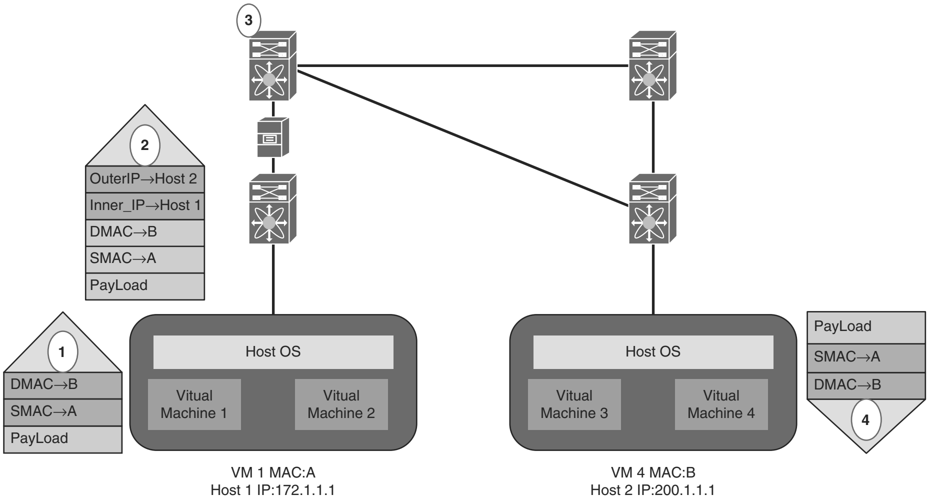


Figure 2-8 *VXLAN Data Path*

The following list describes a sample packet flow in a VXLAN network (refer to Figure 2-8).

1. VM1 (MAC=A) tries to communicate with VM4 (MAC=B). Assume that VM1's and VM4's MAC addresses are known on Host 1 and Host 2 whose VTEP IP addresses are 172.1.1.1 and 200.1.1.1, respectively. VM1 sends out a frame with SMAC=A, and DMAC=B. vSwitch on Host 1 performs a lookup based on (VNI, B).

2. The lookup result yields destination Host 2 as the egress endpoint for this frame. Hence, the ingress vSwitch encapsulates this frame with a VXLAN header for forwarding the frame through the core network. The outer source and destination IP addresses are set as 172.1.1.1 and 172.1.1.2, respectively. The outer DMAC is the MAC address of the next-hop router, and the outer SMAC is the MAC address of the source Host 1.

3. The intermediate routers or switches (for example, 3) do a routing lookup on the outer header and forwards the frame based on the best path to the destination 200.1.1.1. The TTL is decremented, and the outer SMAC and DMAC are rewritten as per regular Layer 3 routing semantics.

4. The destination Host 2 receives this frame. Because the destination IP address points to itself, it decapsulates the packet by removing the outer headers. The packet is forwarded to VM4 based on a Layer 2 lookup on the inner frame, which in this example is (VNI, B).

NVGRE

Network Virtualization using Generic Routing Encapsulation (NVGRE),[7] like VXLAN, is an encapsulation of a Layer 2 Ethernet Frame in IP, which enables the creation of virtualized Layer 2 segments. These virtualized Layer 2 segments, because of the external IP header, can span across Layer 3 networks. NVGRE is based on Generic Routing Encapsulation (GRE), which is a tunneling protocol developed by Cisco. For detail on GRE, refer to www.cisco.com.[8]

As NVGRE creates a connection between two or more Layer 3 networks, it makes them appear as Layer 2 accessible. The Layer 2 accessibility enables VM migrations across Layer 3 networks and inter-VM communication. During these transactions, the VMs operate as if they were attached to the same VLAN (Layer 2 segment).

NVGRE's use of the GRE header enables it to be backward compatible with many stacks because GRE has been there for many years in the switching arena where hardware support is needed for tunnels. Because of this current support of a GRE header in many vendor switches, supporting NVGRE on these platforms is likely to be much simpler than other overlay encapsulations.

All-in-all like VXLAN, NVGRE provides a Layer 2 overlay over an IP network.

NVGRE Frame Format

Figure 2-9 shows the NVGRE frame format. Each of the components of the frame is also described:

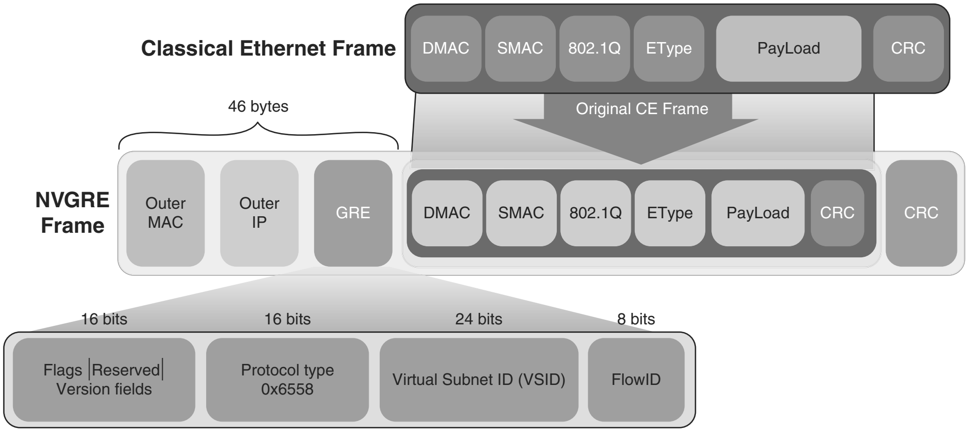


Figure 2-9 *NVGRE Frame Format*

- Outer Ethernet header
- Outer IP header
- NVGRE header
- Inner Ethernet header

NVGRE Data Path Operation

Figure 2-10 shows a sample NVGRE packet flow. At a high level, the NVGRE packet flow is almost identical to that employed by VXLAN except for the different encapsulation header. NVGRE, being a Layer 2 overlay, considers a packet sent by the VM out of one of its vNICs. As the vSwitch receives the packet from the vNIC, it knows the 24-bit Virtual Subnet ID, aka Tenant Network ID (TNI), for this packet. Basically, the vSwitch does two lookups on the packet:

- The vSwitch uses the ingress interface (vNIC) to determine which TNI the packet belongs to.
- vSwitch uses Destination MAC (DMAC) in the packet to determine which NVGRE tunnel the packet should be sent on.
- If DMAC is known, the packet is sent over a point-to-point GRE tunnel.
- If the DMAC is unknown, the packet is sent over a multipoint GRE tunnel with the destination IP being a multicast address associated with the TNI that the packet ingresses on.

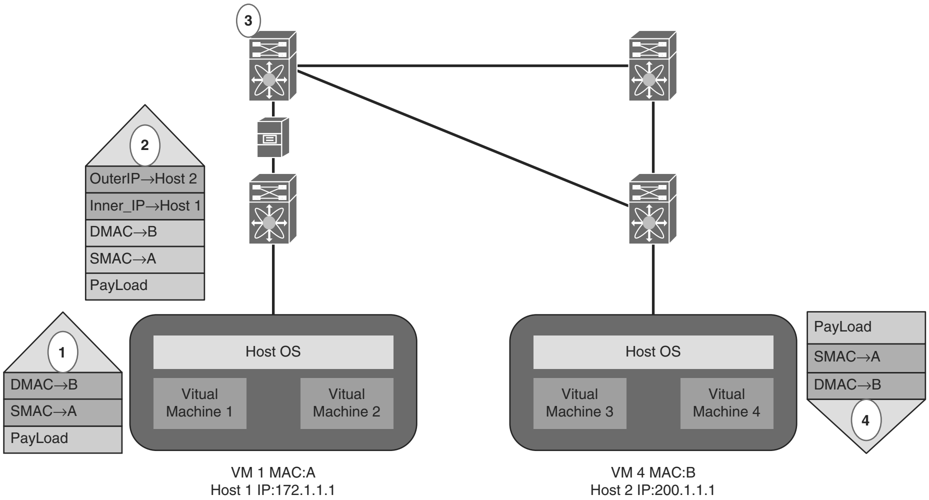


Figure 2-10 *NVGRE Data Path*

Refer to Figure 2-10 to see a high-level flow for a packet traversing the NVGRE network. NVGRE, like VXLAN, is an IP encapsulation, so the data path operation is similar to the VXLAN. The only difference is the GRE header is carried inside the outer IP frame.

1. VM 1 uses its MAC address of A as source MAC (SMAC) and sends a classical Ethernet frame, which is destined to VM 4 with a destination MAC (DMAC) address of B. On receiving this frame, the vSwitch on Host 1 does a Layer 2 lookup based on (VLAN and DMAC).

2. Now consider a case where the destination VM's address is known resulting in a hit in the Layer 2 table. The lookup points to the destination Host 2 as the egress endpoint for this Ethernet frame. The ingress vSwitch encapsulates this frame using the GRE header for forwarding the frame through the core network. The outer source and destination IP addresses are set as 172.1.1.1 and 172.1.1.2, respectively. The outer DMAC is the MAC address of the next-hop router, and the outer SMAC is the MAC address of the source Host 1.

3. The core router or switch (Router/Switch 3 in this example) forwards the frame based on the best path to the destination IP address of Host 2. In this case, there are two paths, but the best path is a single hop away; therefore, the frame is forwarded based on the outer IP address. The TTL is decremented and the outer SMAC and DMAC are rewritten as per regular routing semantics.

4. The destination Host (2) receives this frame. Because the destination IP address in the packet points to itself, Host 2 decapsulates the packet thereby stripping off the outer MAC and the GRE header. It then forwards the frame to VM 4 based on the inner DMAC, VLAN lookup.

Overlay Transport Virtualization

Overlay Transport Virtualization (OTV), also called Over-The-Top virtualization, is a Cisco-proprietary innovation in the Data Center Interconnect (DCI) space enabling Layer 2 extension across data center sites. It was introduced to address the drawbacks of other DCI technologies such as Virtual Private LAN Service (VPLS), Ethernet over MPLS (EoMPLS), Layer 2 over GRE (L2oGRE), and so on.

In OTV, the spanning tree domains remain site-local, and an overlay protocol is employed to share site-local unicast and multicast information with other sites that are all considered part of the same overlay network. OTV employs a MAC in IP encapsulation. One or more edge devices per site that interface with the provider core network are configured with OTV configuration. Each such device has two types of interfaces:

- **Internal interfaces:** It serves as a regular switch or bridge for packets entering and leaving these interfaces. In other words, it does regular SMAC learning based on incoming traffic and DMAC lookup for forwarding the traffic toward the appropriate destination.
- **Overlay interface:** This is a logical interface that faces the provider or core network. It has an IP address in the provider or core address space. All the MAC addresses within a site are advertised to remote sites against this IP address by the overlay control plane.

In OTV, there is no data plane learning. All unicast and multicast learning between sites is facilitated via the overlay control plane that runs on top of the provider/core network. The provider/core network may be Layer 2 or Layer 3. In its most common form, Layer 2 IS-IS is the control protocol of choice for the overlay. All edge devices belonging to the same VPN join the same provider multicast group address thereby allowing peering with each other. The multicast group is employed both for exchange of information in the control plane and sending multicast or broadcast frames in the data plane. A set of multicast group addresses in the provider or core network is made available for OTV usage.

As mentioned earlier, for scalability reasons, spanning tree Bridge Protocol Data Units (BPDUs) are never sent over the overlay interface. Unknown unicast lookups at the edge device are never flooded over the overlay interface but are instead dropped. OTV relies on the concept that hosts or nodes are not silent, and after they speak they will be discovered at a site locally. Then this information will be shared with the remote sites, thereby reducing the probability of unknown unicast lookups at remote sites for existing hosts. Internal Group Management Protocol (IGMP)/Multicast Listener Discovery (MLD) snooping on the internal edge interfaces enables learning about multicast sources, receivers, and group information that, in turn, triggers appropriate joins or leaves on the overlay interface.

To prevent loops and duplicate packets, OTV introduces the concept of an authoritative edge device (AED). A site may have multiple OTV edge devices, and they can either be statically or dynamically configured as AEDs at the granularity of a VLAN or potentially a (VLAN, MAC) combination. An AED is the chosen edge device that is responsible for encapsulating and decapsulating packets to and from the remote sites over the overlay interface for the chosen VLAN or (VLAN, MAC) pair. OTV also supports active-active multihoming to leverage multiple equal-cost paths between edge devices across the provider or core network. Finally, because the functionality of OTV is only on the edge boxes, no changes are required to any core or customer boxes.

OTV Frame Format

OTV employs a MAC in IP frame format, as shown in Figure 2-11 that adds a 42-byte header to each frame transported across the overlay. The source and destination IP addresses are set to that of the source overlay interface and remote overlay interface behind which the destination MAC is located, respectively. The Don't Fragment (DF) flag in the IP header is set to prevent fragmentation of the OTV packet. The OTV header contains an overlay ID and an instance of the ID. The overlay ID is for control plane packets belonging to a particular overlay. The instance ID field provides the option for using a logical table ID for lookup at the destination edge device. For completeness, OTV may also employ an optional UDP encapsulation where the UDP destination port is set to a well-known IANA reserved value of 8472 [13].

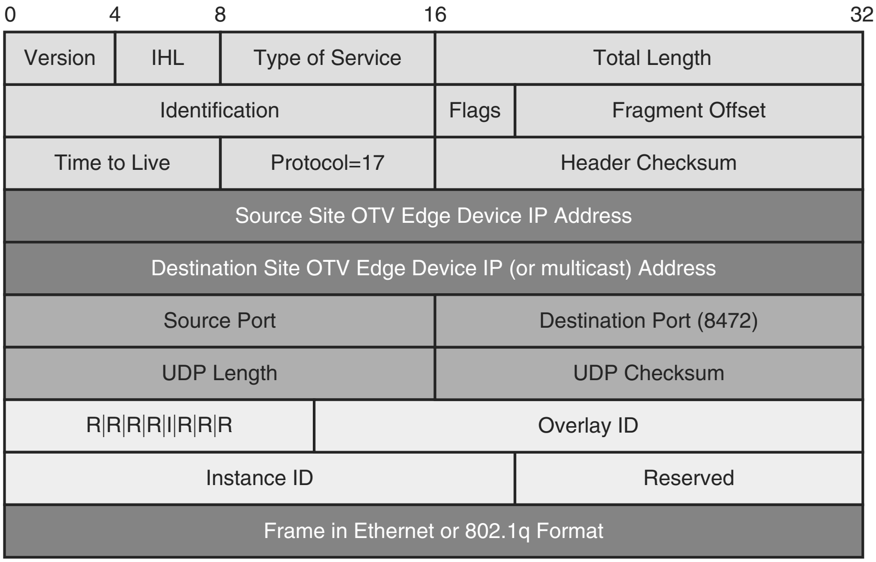


Figure 2-11 *OTV Frame Format*

OTV Operation

In its simplest form, an edge device appears as an IP host with respect to the provider or core network. It learns about (VLAN and uMAC) and (VLAN, mMAC, and mIP) bindings on the internal interfaces (where uMAC is a unicast MAC address, mMAC is a multicast MAC address, and mIP is the multicast group IP address) and distributes it to the remote edge devices via the overlay control plane. The remote devices learn about these bindings against the IP address of the advertising AED. Suitable extensions have been introduced in the Layer 2 IS-IS protocol to enable this information to be carried. including introduction of appropriate TLVs and sub-TLVs. For more information on this, refer to reference [12] at the end of this chapter.

Figure 2-12 shows that MAC1, MAC2, and MAC3 represent hosts in data centers 1, 2, and 3, respectively. Based on the exchange of learned information on the overlay control plane, an edge device on data center 1 has its MAC table appropriately populated. This enables dynamic encapsulation of packets destined from data center 1 to MAC2 and MAC3.

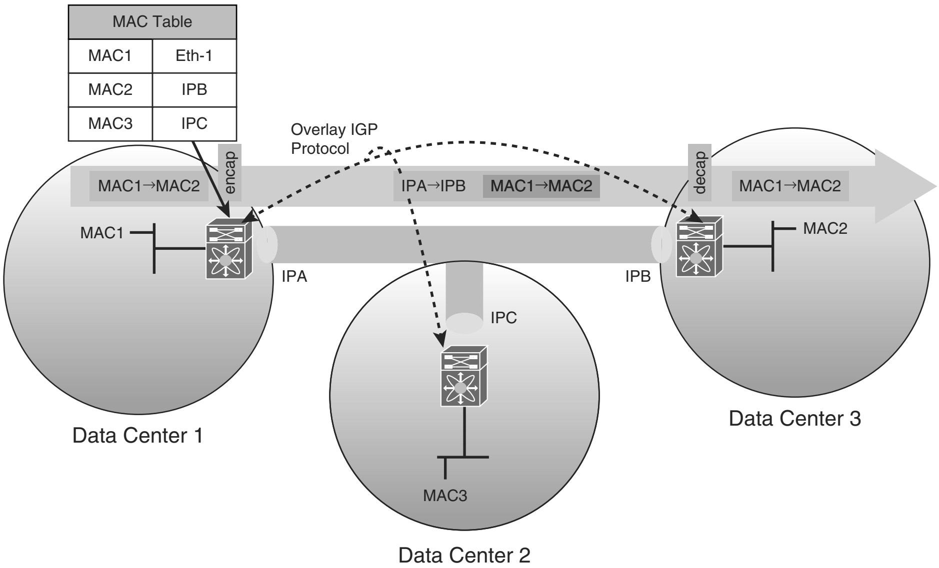


Figure 2-12 *OTV Illustration*

OTV enables quick detection of host mobility across data centers. This is facilitated by the control plane advertisement of a metric of 0 for the moved host. All remote sites update their MAC table on reception of an advertisement with metric 0. In addition, the old site on reception of such an advertisement can detect a host move event and withdraw its original advertisement.

Provider Backbone Bridges (PBB)

802.1ah[9] is an IEEE standard developed for addressing the scalability concerns in carrier Ethernet. It is a MAC-IN-MAC encapsulation scheme that addresses the limitation of 4KVLANs and MAC address table explosion at the metro-Ethernet provider. Figure 2-13 shows the historical evolution of Ethernet tagging.

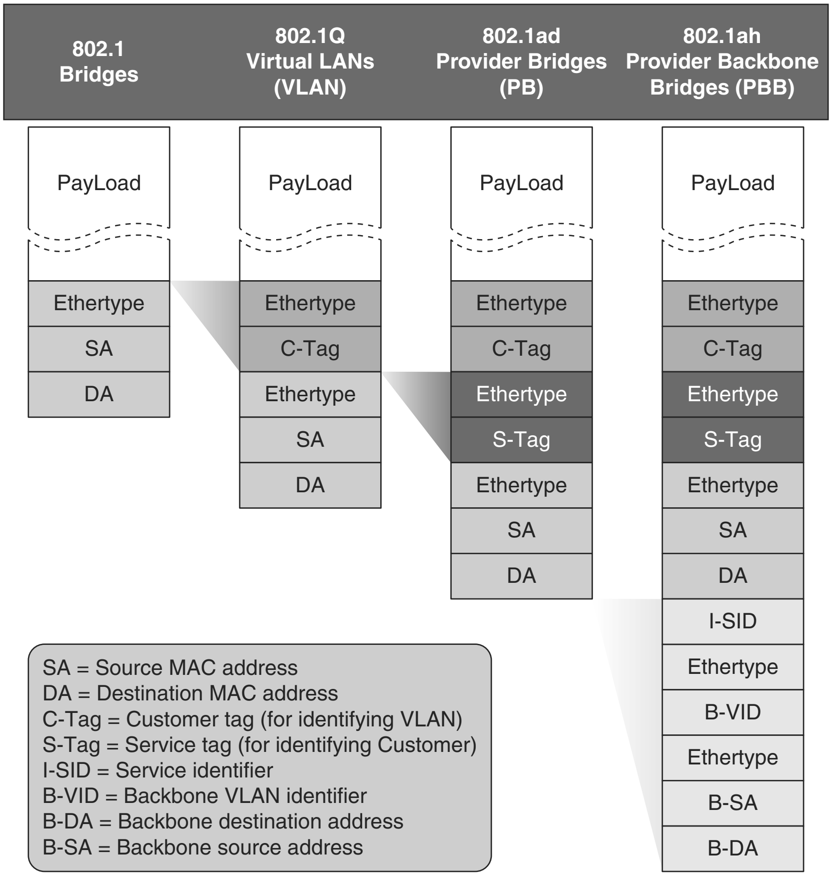


Figure 2-13 *802.1 Frame Formats*

802.1Q defines the customer frames to be differentiated using a 2-byte TAG. The Tag consisted of a 12-bit VLAN field, which enables roughly 4 K services to be provided. The 802.1ad (also called QinQ) enables the customer and provider tags to be separated. The idea is to separate the customer and provider space. The frame consists of a customer tag (C-TAG) along with a service tag used at the service provider core (called S-TAG). The 4 K limitation of service instances was to an extent addressed by 802.1ad, but the MAC table explosion still remained at the core. The 802.1ah defines bridge protocols for the inter-connection of provider bridged networks (PBN). An 802.1ah frame is shown in Figure 2-14.

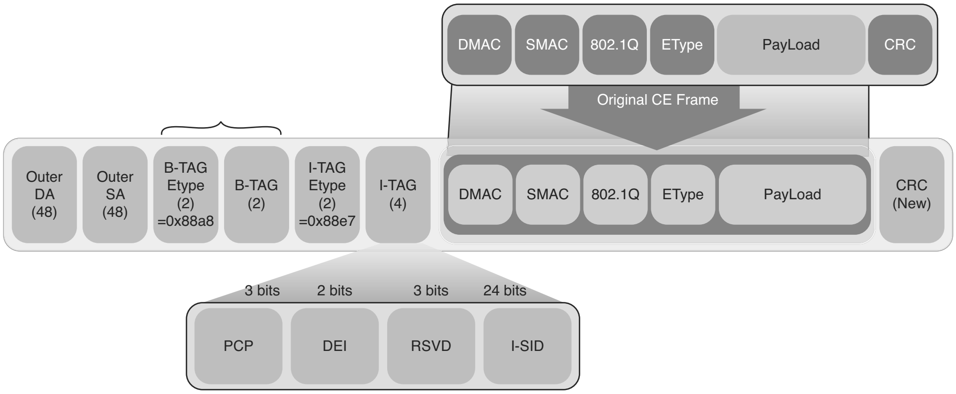


Figure 2-14 *802.1ah Frame Formats*

Figure 2-14 shows an 802.1ah header, which consists of the following:

Outer DA: This is the Destination Backbone Edge Bridge's MAC Address.

Outer SA: This is the Source Backbone Edge Bridge's MAC Address.

BTAG: This field prefixed with the Ether Type represents the Backbone VLAN.

ITAG: This field prefixed with the Ether Type represents the service identifier.

The header fields will become clear after you go through a packet flow. Consider the topology shown in Figure 2-15:

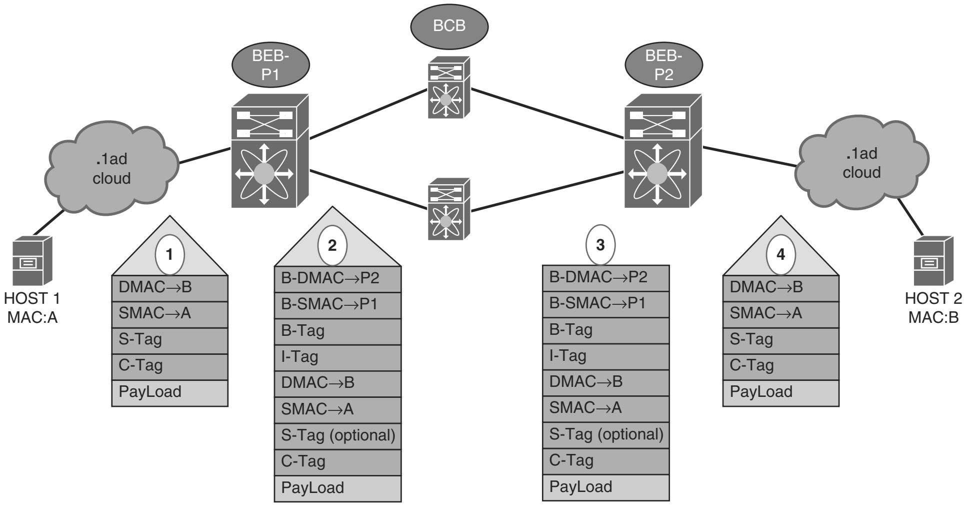


Figure 2-15 *IEEE 802.1ah Data Path*

Figure 2-15 shows a customer frame, after it arrives at the Backbone Edge Bridge (BEB), is encapsulated with an 802.1ah header. The customer frame is associated with a service instance. The S-VLAN or C-VLAN or a combination of both can be used to derive a service instance. The 802.1ah enables a provider to support up to 2^24 service instances (16 million). The Backbone VLAN is used at the provider core to form different bridge domains. The outer B-SA and B-DA are the source and destination MAC addresses of the BEBs. The Backbone Core Bridge (BCB) does regular Layer 2 bridging and doesn't need to be 802.1ah-aware. At the egress, a BEB learns the inner MAC address and associates it with the BEB's MAC address present at the outer header. In this way, the provider core needs to learn only all the BEB's MAC addresses. It's true that this doesn't solve the BEB's MAC address table explosion problem. The egress BEB then removes the outer 802.1ah header, derives the service instance of the frame based on the I-SID, and then forwards the frame to the destination host.

This example assumes that the MAC addresses are already learned. Of course, the source BEB initially does not know the B-DA for the destination host's MAC address (MAC:B). Therefore, the destination host's MAC address is not present in the Layer 2 table; the frame is flooded to all the BEB's in the network. There are optimizations that require the frame to be sent to only the BEBs that are part of the Backbone VLAN, or a special multicast address is used that allow the frames to be sent to the BEBs that have hosts in the service instance.

Shortest Path Bridging

Shortest Path Bridging (SPB) is an IEEE standard (802.1aq). It tries to solve the same problems as TRILL and FabricPath do, namely inefficient utilization of links and scalability concerns due to Spanning Tree, MAC table explosion in the MSDC environment, but still retaining the simplicity of a Layer 2 network.

SPB used Layer 2-IS-IS with some specific extensions[10] as the control protocol. Every SPB switch computes a number of shortest path trees with every other SPB switch as the root. SPB comes in two flavors, SPBM and SPBV. SPBM uses an IEEE 802.1ah format, whereas SPBV uses an IEEE 802.1ad format.

Shortest Path Bridging MAC

The overlay header and the data path for Shortest Path Bridging MAC (SPBM) are similar to that of 802.1ah, described in the previous section. Now consider an example for unicast traffic. A sample topology is shown in Figure 2-16. Layer 2-IS-IS distributes the Backbone MAC (BMAC) of all the SPB nodes. Every node has a link state database of all the nodes in the SPB network identified uniquely by its BMAC, and the shortest path is computed for every node.

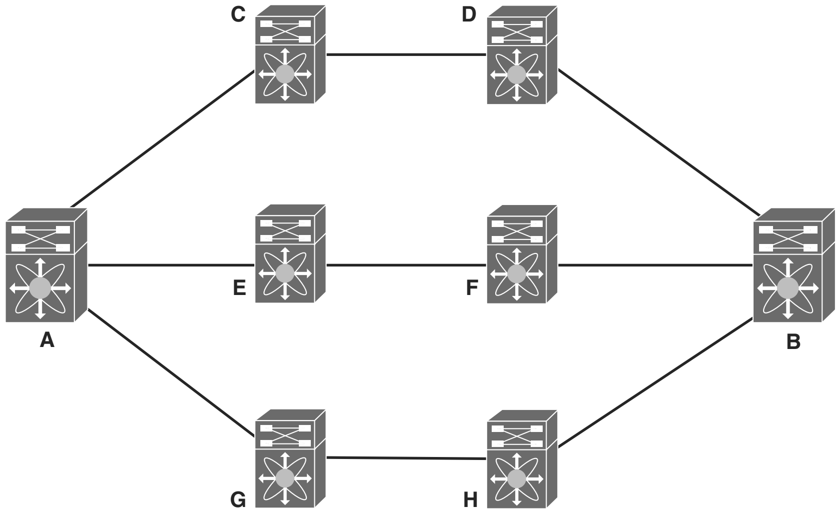


Figure 2-16 *IEEE 802.1aq Data Path*

In the figure, from node A to B, there are three different equal cost shortest paths available. Each path is assigned a different B-VID. B-VID is the backbone VLAN ID, which is a subfield in the B-TAG field of the 802.1ah frame shown in Figure 2-14. The path that has the lowest node ID is chosen among the different equal cost paths (Low Path ID). Alternately, the path with the highest node ID can also be chosen among the different equal cost paths. All the nodes in the SPB network use the same tie-breaking mechanism. The exact tie-breaking mechanism used is advertised in the IS-IS TLV to ensure all nodes use the same tie-breaking mechanism to guarantee symmetric lookup results for traffic forwarding. Node ID, here, is the MAC-Address of the bridge. So, assuming Low Path ID is used, the frames from A to B traverse the path A - C - D - B.

The reverse path can also flow through the same set of nodes. When a native frame arrives at the ingress node of the SPB network, a look up is done in the Layer 2 table to find the egress node of the SPB network behind which the host is located. In the example, say, the look up result indicates B as the egress node. Node A then encapsulates the frame into an 802.1ah header. The B-VID information for the corresponding path is exchanged through the Layer 2 IS-IS. The B-MAC is the MAC address of egress node, which is B. The appropriate I-SID is used based on the classification of the frame. Then, another lookup is performed in the forwarding table to find the outgoing interface for B-MAC - 'B'. This is where SPBM differs from 802.1ah. The802.1ah operates like a regular Layer 2 switch in terms of learning and forwarding of the BMAC, BVID in the 802.1ah header. In SPB-M, the topology information of all nodes, identified by its B-MAC, is distributed through IS-IS and a shortest path is computed for every BMAC. BMAC is the routable entity in SPB-M, and the result of the lookup for BMAC, BVID is the interface to reach the next hop obtained through shortest path computation. Switch A, after a lookup in the forwarding table, chooses the interface connecting to Switch C as the outgoing interface. Switch C does a lookup based on the outer 802.1ah header (B-MAC and

B-VID and optionally ISID, as will be explained subsequently) and forwards the frame to node D. Node D forwards the frame to the egress node B. Node B decapsulates the frame; it learns the source MAC address of the sending host (inner source MAC address) against the ingress SPB node, which in this example is A. Then, the original frame is forwarded using the traditional way based on the Layer 2/Layer 3 header.

The key points to note here are that the data traffic between two SPB nodes is symmetrical and no learning on B-MAC happens at the SPB network.

There are further optimizations proposed for load balancing the traffic. One mechanism picks the path among the different set of ECMP paths based on the I-SID. As can be recalled from the preceding section on 802.1ah, I-SID is the service instance to which the frame belongs. So, traffic belonging to different services takes different paths, thereby achieving load balancing. To illustrate this, a simple mechanism consists of employing a modulo-operation of I-SID with the total number of equal cost paths to yield the path to be taken.

The example described in this section is for unicast traffic, where in the destination host MAC was already present in the Layer 2 table of the ingress node. Therefore, the look up for the inner DMAC is a miss; a special multicast address is used as the destination B-MAC. The multicast address is derived based on the I-SID, to which the frame belongs to and the source node ID from where the traffic is rooted, which is node A in this example. So, there is per-source, per-service multicast forwarding and the multicast address uses a special format to achieve this. The low-order 24 bits represent the service ID and the upper 22 bits represent a network-wide unique identifier. The I/G bit is set and the U/L bit is also set to mark it as a nonstandard OUI address. To achieve this forwarding behavior, Layer 2 IS-IS carries information about which nodes are members of a given service. Because all nodes compute the tree rooted at every SPB node, they populate the forwarding table with the different multicast addresses along with the outgoing interfaces.

Shortest Path Bridging VID

Shortest Path Bridging VID (SPBV) is wanted when there is no need to separate the customer and core address spaces. The control plane operation in SPBV is similar to that of SPBM. A SPBV frame is single (one 802.1q tag) or double (that is, q-in-q) tagged. There's no overlay header added for SPBV unlike the SPBM case. SPBV limits the number of VLANs in the network.

The 802.1q[11] tag is overloaded to carry both the VLAN ID and the Tree ID. Each node computes a shortest path tree to all other nodes. Consequently, SPBV is suitable for deployments in which Number_of_VLAN's * Number_of_Nodes < 4K. A VLAN or 802.1q tag translation happens at the ingress. The translated tag is called the SPVID, Now consider a simple example with SPBV for a bidirectional traffic flow. A sample topology is shown in Figure 2-17 where the Node ID of each node is listed in parenthesis. Layer 2 IS-IS distributes the SPVID of all the SPB nodes. Every node has a link state database of all the nodes in the SPB network, and the shortest path tree is computed for every node.

For example, Node E would have computed a number of shortest path trees with nodes A to H as the root.

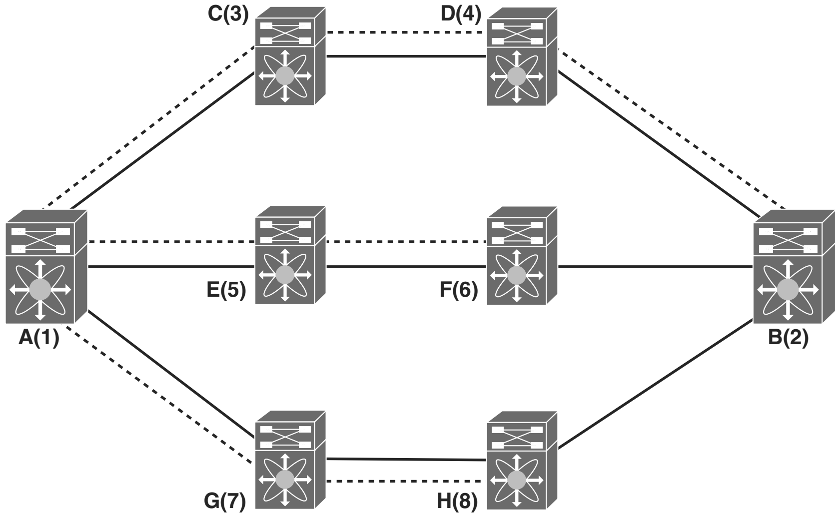


Figure 2-17 *SPBV Data Path*

SPVID is calculated as Base VID + Node ID. For example, if there's a VLAN 100, there will be 8 different node IDs, as shown in Figure 2-17, starting from 101 to 108, where 1 to 8 are the node IDs. Forwarding in SPBV works as follows: Broadcast, multicast, and unknown unicast frames are forwarded out of the Shortest Path Tree (SPT) with the ingress as the root. IS-IS already would have propagated the tree information along with its appropriate SPVID, which also identifies the tree. Known unicast frames are forwarded based on the lookup result. The key for the lookup and the learning mechanism employed is HW-specific, and that dictates how this forwarding behavior is achieved.

Now walk through a conceptual packet flow. When a frame arrives at ingress node A, an 802.1q tag translation or encapsulation is done on the frame based on whether the original frame arrived with a 802.1q tag or was untagged. The tag translation is done based on the VLAN to which the outgoing frame is associated with and the node ID of the ingress. For example, if the frame arrives at A and the outgoing frame is associated with VLAN 100, the SPVID in the frame will have the value of 101. Initially, when the frame arrives at A, a lookup on destination B will be a miss in its forwarding table. Lookup is done based on VLAN and the MAC address of B. Because it's a miss, the frame is forwarded along the SPT corresponding to SPVID 101, which are interfaces connecting to C, E, and G. The outgoing interfaces are a part of the shortest path tree computed by A. The tree with node A as the root is shown in dotted lines in Figure 2-17.

When the frame arrives at C, E, and G, each derives the interfaces corresponding to the tree (SPVID of 101) because the lookup is a miss. Each node can also do a reverse path check to ensure that the frame arrived at a valID interface and is a part of the computed shortest path to the node ID, carried in the SPVID. The frame is forwarded to destination

B and follows the path shown in Figure 2-17 (A -> C -> D -> B). Nodes F and H terminate the frame and don't forward the frame to B because the tree for SPVID 101 does not include the links F - B and H - B. Node B learns the <MAC address of A, VLAN (value of 100)> against the incoming interface. An important thing to note is learning does not happen on the incoming SPVID (value of 101). The reason will become clear when you trace the packet flow for the reverse traffic.

When reverse traffic arrives at B, an 802.1q tag translation is done, as explained previously, and the SPVID will have the value of 102 because the node ID of B is 2. A lookup in the forwarding table for destination MAC address of" and VLAN ID of 100 points to the interface connecting to D. The reverse traffic follows the path B -> D -> C -> A. If learning had happened on just SPVID 101 instead, a lookup for the reverse traffic would have been a miss, which is not wanted. The basic idea is that learning happens by ignoring the node ID.

To summarize, shared VLAN learning is employed for unicast traffic and node ID-specific entries are installed in the forwarding table for broadcast and multicast traffic.

Summary

In this chapter, you learned about the various Layer 2 and Layer 3 overlay technologies including Cisco Fabric path, TRILL, LISP, VXLAN, NVGRE, OTV, PBB, and Shortest Path Bridging. Each of these technologies tries to address various limitations of today's networks including mobility across networks which are Layer 3 apart. Layer 2 LISP, VXLAN, NVGRE, PBB, and Shortest Path Bridging also address the current limitation of 4094 Virtual LANs by providing 16 million logical Layer-2 segments in the network.

References

IS-IS for Layer 2 Systems:

1. http://www.ietf.org/rfc/rfc6165.txt

FabricPath:

2. http://www.cisco.com/en/US/docs/switches/datacenter/nexus5000/sw/fabricpath/513_n1_1/fp_n5k_switching.html#wp1790893
3. http://www.cisco.com/en/US/prod/collateral/switches/ps9441/ps9402/white_paper_c11-687554.html

VXLAN:

4. http://tools.ietf.org/html/draft-mahalingam-dutt-dcops-vxlan-02
5. http://www.techmahindra.com/Documents/WhitePaper/VXLAN2011.pdf
6. http://www.borgcube.com/blogs/2011/11/vxlan-primer-part-1/

NVGRE:

7. http://tools.ietf.org/html/draft-sridharan-virtualization-nvgre-01

GRE:

8. http://www.cisco.com/en/US/tech/tk827/tk369/tk287/tsd_technology_support_sub-protocol_home.html

802.1ah:

9. http://www.ieee802.org/1/pages/802.1ah.html

SPB:

10. http://tools.ietf.org/html/rfc6329
11. http://www.ieee802.org/1/pages/802.1aq.html
12. http://www.cisco.com/en/US/docs/solutions/Enterprise/Data_Center/DCI/whitepaper/DCI3_OTV_Intro_WP.pdf

OTV:

13. http://tools.ietf.org/html/draft-hasmit-otv-04

Index

Numerics

A

D

E

F

G

H

I

J-K-L

M

N

O

P

Q-R

S

T

U

V

W-X-Y-Z