



DEPLOYING JUNIPER DATA CENTERS WITH EVPN VXLAN



ANINDA CHATTERJEE

FREE SAMPLE CHAPTER |



The depth, detail, and thoroughness of this book easily surpasses any other VXLAN/EVPN book on the market. And it is the only book available that covers the topic from a Juniper Junos and an Apstra perspective. Whether you want a VXLAN/EVPN technical deep-dive, want to learn how to configure it on Junos, want to learn Apstra's Intent-Based Networking platform, or are studying for your JNCIE-DC lab, this book is essential for data center engineers and architects.

—*Jeff Doyle, Director of Solutions Architecture
Juniper Networks/Apstra*

Aninda has written the new definitive guide for learning, building and operating EVPN networks. This book should be on the shelves of any network engineer, from NOC technicians to senior architects.

—*Pete Lumbis, CCIE No. 28677, CCDE 2012:3*

Today's data centers require modern technologies that simplify operations and assure reliability at the tremendous scale demanded by AI training and digital applications. Juniper innovation is in the forefront with Apstra Intent-Based Networking automation for EVPN VXLAN multivendor networks. *Deploying Juniper Data Centers with EVPN VXLAN* is a comprehensive guide that includes all these technologies in one place to understand how they work together for robust, automated DC operations. Architects and operators responsible for the integrity of the data center will want this go-to book to advise step by step how to set up and run their network following Juniper recommended, best practice designs, tools, and workflow.

—*Mansour Karam, GVP
Juniper Networks*

Juniper's data center fabric solutions are world-renowned for their completeness and quality. This book begins right at the beginning, with basic data center fabric design, BGP in the data center, and VXLAN. After covering these topics, Aninda moves into an explanation of Apstra, one of the most complete multi-vendor intent-based data center fabric systems.

The many graphics and screen shots, combined with the detailed configuration and sample outputs, provide designers and operators alike with deeply researched and well-explained information about building and operating a data center fabric using Juniper hardware and software.

I even learned a few things about Apstra reading through this book—although I have built and operated networks using Apstra's technology.

I highly recommend this book for engineers looking for a good explanation of Juniper data center solutions.

—*Russ White*

Aninda is an outstanding engineer with an insatiable thirst for knowledge and discovery. His drive is endless and a wonderful opportunity for himself and many others to learn and explore subjects and technologies, as he is able to simplify them in a way that allows others to learn seamlessly. I have enjoyed Aninda's content for several years now. He has contributed [to] the community through webinars, articles, white papers, and blogs, which makes his book a logical step to consolidate his contributions and knowledge.

Aninda's work will always have my support and endorsement.

—*David Penalosa, Principal Engineer*

This page intentionally left blank

Deploying Juniper Data Centers with EVPN VXLAN

This page intentionally left blank

Deploying Juniper Data Centers with EVPN VXLAN

Aninda Chatterjee

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and the publisher was aware of a trademark claim, the designations have been printed with initial capital letters or in all capitals.

The author and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

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.

Visit us on the Web: informit.com/aw

Library of Congress Control Number: 2024934201

Copyright © 2024 Pearson Education, Inc.

Published by:

Addison-Wesley

Hoboken, New Jersey

All rights reserved. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. For information regarding permissions, request forms and the appropriate contacts within the Pearson Education Global Rights & Permissions Department, please visit www.pearsoned.com/permissions/.

Please contact us with concerns about any potential bias at <https://www.pearson.com/report-bias.html>.

ISBN-13: 978-0-13-822539-1

ISBN-10: 0-13-822539-7

\$PrintCode

GM K12, Early Career and Professional Learning: Soo Kang

Executive Editor: Brett Bartow

Development Editor: Ellie C. Bru

Managing Editor: Sandra Schroeder

Senior Project Editor: Tonya Simpson

Copy Editor: Bill McManus

Indexer: Erika Millen

Proofreader: Donna E. Mulder

Composer: codeMantra

Technical Reviewers: Jeff Doyle, Vivek Venugopal, and Ridha Hamidi

Editorial Assistant: Cindy Teeters

Cover Designer: Chuti Prasertsith

Figure Credits

Figures 3.11–3.15, 5.7, 5.8, 5.12, 5.13, 5.21, 5.22, 5.28, 5.30, 5.32, 5.33, 5.35, 5.47, 5.52–5.54, 6.7, 6.12, 6.13, 6.17, 7.8, 7.13, 7.14, 7.19, 7.22, 7.24, 7.26, 8.11, 8.13, 8.16, 8.21, 8.22, 8.26, 8.28, 9.5, 9.10, 10.4, 10.6, 10.8, 10.9, 10.12–10.14, 10.18, 10.19, 11.10, 11.17, 11.18, 11.21, 11.27–11.30, 11.34, 11.38–11.40, 12.15–12.18, 13.9, 13.40, 13.41: Wireshark Foundation

Figures 12.1–12.4, 12.6, 12.8, 12.10, 12.13, 12.14, 12.19–12.28, 12.30–12.40, 12.42–12.65, 12.67–12.75, 13.2–13.8, 13.12–13.26, 13.28–13.39, 13.42, 13.43, 13.45–13.59, 14.2–14.10, 15.3–15.7, 15.11–15.16, 15.18, 15.19, 15.24–15.33: Juniper Networks, Inc

Dedications

This book is dedicated to the family I was born into, and the family I married into.

Foreword

The titans of the networking industry stand tall not because they have proven themselves masters of theory. Nor is it because they have waxed poetic about all manners of enabling our connected world. Those whose heads and shoulders rise above achieve their place because they are practitioners.

And so, as we evaluate technical works for their transformative potential, we should come to know our authors by their hands-on skills more than their willingness to pontificate. Their experience is the bedrock on which truly great works are built.

But let's be honest. Our industry is one where most of the really important work is done behind closed doors, in places where peering eyes might never reach. So how do you assess skills when the work they deliver is hidden by design?

I have had the great pleasure of building multiple organizations over the years. I have led data center businesses at multiple large vendors, which has given me the opportunity to assemble all kinds of teams. Early in my career, I would seek out experience. But as I matured and became a better leader, I learned to hunt for potential.

In my not terribly humble opinion, the highest potential exists at the intersection of capability, drive, and humility. Capability is table stakes of course, and drive is an obvious prerequisite for progress. But humility might be the secret ingredient that brings everything together.

You see, it's easy to be humble when you are starting out because you lack the experience to know how good you are. All too often, there is an inverse relationship between experience and humility—indeed, many of us become louder as we develop a stronger command over our domains! But you cannot become a true master without true humility because it is the constant awareness of what you do not know that provides the impetus to continue learning.

Naturally, our industry's strongest spokespeople will then be brimming over with humility. When Aninda and I first crossed paths, he spoke fluently about technology and experience—the kinds of things you lead with during an interview, of course. But what I heard was different. As accomplished as Aninda is, I could see that he has a real learner's mind.

That learner's mind might make for some restless nights as Aninda never seems quite comfortable with where he is in his journey. But I can't help but think of the great Theodor Seuss Geisel book *Oh, the Places You'll Go!*, because oh, what a journey it will be.

This book represents a checkpoint of sorts in Aninda's journey so far. It's meant to be an approachable guide to data center networking, explaining how EVPN VXLAN data centers are architected and operated, but importantly, using the hands-on experience that Aninda has earned through the years to make it tangible.

And if you read this book with the same learner's mind with which it has been written, oh, the places you will go.

—Michael Bushong
VP, Data Center
Nokia

Acknowledgments

As the author, it is easy to say that I wrote this book, but that is hardly the complete truth. Technically, yes, I put these words on paper, but there were so many people who helped me get to the point in my journey where I felt confident and capable enough to do this.

There are many excellent engineers who helped keep this book technically accurate, provided support when I was lost, and validated what I wrote. This also includes individuals who probably have no idea how much I have learned from them by reading their books, learning from content created by them, or have supported me in my professional and personal growth. In no particular order, they are Ridha Hamidi, Vivek Venugopal, Soumyodeep Joarder, Anupam Singh, Selvakumar Sivaraj, Wen Lin, Mehdi Abdelouahab, JP Senior, Jeff Doyle, Russ White, Jeff Tantsura, Ivan Pepelnjak, Dinesh Dutt, Pete Lumbis, Richard Michael, Peter Paluch, David Peñaloza, Daniel Dib, Naveen Bansal, Manasi Jain, and Astha Goyal.

To Brett Bartow, Eleanor Bru, Tonya Simpson, Bill McManus, Donna E. Mulder, and everyone from Pearson who helped bring this book to life: Thank you for giving me the opportunity to write this and taking a chance on a nobody. Your support through the writing, production, and composition process has been nothing short of exceptional.

To my technical reviewers, Ridha Hamidi, Vivek Venugopal, and Jeff Doyle: Thank you for reading my manuscript with gentle hands. You made it better in every way, giving constructive but honest feedback. I'd have never imagined there would come a day when I would be collaborating with Jeff Doyle, whose books I learned my networking skills from. Professional dreams do come true.

To Souvik Ghosh and Reghu Rajendran: Back in late 2011, sitting in a meeting room in the offices of Cisco Systems, Bangalore, you both interviewed me and gave me the opportunity of a lifetime. My days in Cisco TAC were some of my best. I followed you into heavy-hitting escalation roles, working together on some of the most challenging technical escalations. Thank you for guiding and mentoring me.

To Dale Miller: You are, undoubtedly, the best mentor I could have asked for. You saw potential in me when I saw none. You pushed me to new heights, to try things out of my comfort zone, and taught me what true customer advocacy means. Cisco Live conferences, bringing up new TAC centers, and solving some of the hardest escalations—we've been through it all together. You are one of the brightest spots in my career and I am glad I can call you my friend. And to Matt Esau: Like Dale, you mentored me through tough times, and even now I can reach out to you for guidance and support. I am lucky to know you and to have worked with you.

To Pete Lumbis: I can't believe we haven't worked together yet, despite literally being one "yes" away from it. You are one of the most talented engineers I have the privilege of knowing and learning from. And with all that brain power, you continue to be humble and down to earth, and you constantly reach out with helping hands. Most importantly, you genuinely look out for your peers, and you nurture those just starting this journey. You read the entire manuscript for this book, even when you had no reason to, just to give feedback and show your support.

To my dearest friends, Vivek and Gino: It's funny how long our bond has lasted because I was quite certain I was intolerable on the TAC floor, with all my cursing. But I guess like minds do think alike. We've looked out for each other since 2012. It has truly been a blessing to have both of you by my side in this journey.

To Cathy Gadecki and Mike Bushong: I have been a network engineer for over 12 years now, spanning five different roles across several companies. Your leadership, unequivocally, is the best I have experienced. For me, it wasn't about technical growth—I know how to get that for myself. You both provided personal growth and helped me nurture skills I considered irrelevant. Mike, there's no leader like you, and I don't think there ever will be. There's a reason people follow you—sure, part of it is loyalty, but there's so much more to it. You genuinely care about people and you do everything in your control to make their lives better.

To my parents, Alope and Sujata Chatterjee, my brother, Arnab Chatterjee, and his wife, Radhika Arora: You have shaped me, as an individual, throughout my life. My interaction with the world is modeled after you and the values you taught me. Everything I have and I am stems from your kindness and love.

To my wife, Deepti: There is no measure of success without you. This last year has been grueling trying to balance work and writing this book. You were supportive every step of the way, giving me the time and space to write while managing your own work, taking care of our home, and being the best mother to our little girl. You make me a better person and a better father every day. I love you dearly and I am glad I get to walk this winding road of life with you by my side.

And to my little one, Raya: You're too young to read this, but maybe some day you will. You are the light of our lives. Now and forever.

About the Author

Aninda Chatterjee holds a Bachelor of Engineering degree in Information Science. His networking career started at AT&T, troubleshooting Layer 1 circuit issues, eventually transitioning to customer support at Cisco TAC, specializing in Layer 2. After his stint at Cisco TAC, he has held several roles across different organizations, with functions including escalation support for enterprise and data center engineering, designing, implementing, and troubleshooting enterprise and data center networks, and technical marketing for Cisco Software Defined Access (SDA).

In his current role as a senior technical marketing engineer at Juniper Networks, Aninda specializes in data center networks with EVPN VXLAN, while also focusing on the high demand of networking infrastructure for high-performance computing and AI/ML clusters.

Aninda actively writes on his personal blog, www.theasciiconstruct.com.

About the Technical Reviewers

Jeff Doyle is a director of solutions architecture at Juniper Networks. Specializing in IP routing protocols, complex BGP policy, SDN/NFV, data center fabrics, IBN, EVPN, MPLS, and IPv6, Jeff has designed or assisted in the design of large-scale IP and IPv6 service provider networks in 26 countries over 6 continents.

Jeff is the author of *CCIE Professional Development: Routing TCP/IP*, Volumes I and II; *OSPF and IS-IS: Choosing an IGP for Large-Scale Networks*; *Intent-Based Networking for Dummies*; was a co-author of *Network Programmability and Automation Fundamentals*; *Software Defined Networking: Anatomy of OpenFlow*; and is an editor and contributing author of *Juniper Networks Routers: The Complete Reference*. Jeff is currently writing *CCIE Professional Development: Switching TCP/IP*. He has also written for *Forbes*, has blogged for both *Network World* and *Network Computing*, and is co-host of the livestream show *Between 0x2 Nerds*. Jeff is one of the founders of the Rocky Mountain IPv6 Task Force, is an IPv6 Forum Fellow and a 2019 inductee into the IPv6 Internet Hall of Fame, and serves on the executive board of the Colorado chapter of the Internet Society (ISOC) and the advisory board of the Network Automaton Forum (NAF).

Vivek Venugopal has been in the computer network industry for more than 15 years. His experience spans multiple domains such as enterprise, data center, service provider networking, and network security. He has worked with a variety of networking giants such as Cisco Systems, Juniper Networks, and VMware in various capacities, and has founded a startup in the networking education space as well.

Ridha Hamidi, PhD, has decades-long experience in the telecommunications and Internet industries and has worked with both service providers and equipment vendors. He holds multiple industry-recognized certifications, such as JNCIE-SP, Emeritus. In his current role as a senior technical marketing engineer at Juniper Networks, Ridha has multiple responsibilities in projects involving data center technologies such as EVPN-VXLAN and, more recently, AI/ML Workloads.

Contents at a Glance

	Introduction	xvii
Chapter 1	Introducing the Juniper Ecosystem	1
Chapter 2	Overview of Data Center Architecture	31
Chapter 3	BGP for the Data Center	43
Chapter 4	VXLAN as a Network Virtualization Overlay	69
Chapter 5	Bridged Overlay in an EVPN VXLAN Fabric	81
Chapter 6	MAC-VRFs	189
Chapter 7	Centrally Routed Bridging	225
Chapter 8	Edge-Routed Bridging	279
Chapter 9	Routed Overlay and Host-Routed Bridging	325
Chapter 10	DHCP in EVPN VXLAN Fabrics	353
Chapter 11	Data Center Interconnect	377
Chapter 12	Building Data Centers with Juniper Apstra, Part I—Apstra Foundation	455
Chapter 13	Building Data Centers with Juniper Apstra, Part II—Advanced Apstra Deployments	517
Chapter 14	Building Virtual Fabrics with vJunos, Containerlab, and Juniper Apstra	575
Chapter 15	Large-Scale Fabrics, Inter-VRF Routing, and Security Policies in Apstra	591
	Acronym Legend	631
Appendix A	Quick Reference Guide	635
	Index	647

Contents

	Introduction	xvii
Chapter 1	Introducing the Juniper Ecosystem	1
	Junos Architecture	1
	Building Layer 2 and Layer 3 Networks with Junos	3
	Introducing the Junos CLI	4
	Building a Network with Junos	11
	Miscellaneous Junos Features	25
	Rescue Configuration	25
	Junos Copy Utility	26
	Junos Groups	26
	Junos Insert Utility	28
	Summary	30
Chapter 2	Overview of Data Center Architecture	31
	History and Evolution of Data Centers	31
	Data Center Designs and Overlay Architectures	37
	3-Stage Clos Fabric	37
	5-Stage Fabric	39
	Collapsed Spine Design	40
	Summary	41
Chapter 3	BGP for the Data Center	43
	BGP Path Hunting and ASN Scheme for Data Centers	44
	Implementing BGP for the Underlay	49
	Auto-Discovered BGP Neighbors	59
	Summary	67
Chapter 4	VXLAN as a Network Virtualization Overlay	69
	Introducing VXLAN	70
	EVPN for Data Center VXLAN Fabrics	75
	Summary	79
Chapter 5	Bridged Overlay in an EVPN VXLAN Fabric	81
	Configuring and Validating a Bridged Overlay EVPN VXLAN Fabric	82
	Configuring the Underlay	83
	Configuring the Overlay	91
	Packet Flow in a Bridged Overlay Fabric	97
	Learning MAC Addresses and EVPN Type-2 Routes	101
	High-Level Software Architecture for MAC Address Learning	101
	Learning Local MAC Addresses	102
	Learning Remote MAC Addresses	112
	Proxy ARP and ARP Suppression	116
	Replication of BUM Traffic and EVPN Type-3 Routes	120
	EVPN Multihoming with ESI LAG and EVPN Type-1/Type-4 Routes	127
	Configuring ESI LAG and EVPN Multihoming	129

- MAC Address Synchronization Across ESI LAG VTEPs 132
- EVPN Type-4 Routes and the Need for a Designated Forwarder 139
- Aliasing, Fast Convergence, and Split Horizon with EVPN Type-1 Routes 147
- Core Isolation in an EVPN VXLAN Fabric 157
- Route Targets in an EVPN VXLAN Fabric 159
- MAC Mobility 169
- Loop Detection 173
 - Connectivity Fault Management 178
 - Loop Prevention Mechanism Using IETF Draft draft-snr-bess-evpn-loop-protect 181
- Bidirectional Forwarding Detection in an EVPN VXLAN Fabric 182
- Summary 188

Chapter 6 MAC-VRFs 189

- Introducing EVPN Service Types 189
- VLAN-Based MAC-VRFs 191
- Order of Operations with MAC-VRFs 200
- Shared Tunnels with MAC-VRFs 201
- VLAN-Aware MAC-VRFs 204
- Overlapping VLANs, VLAN Translation, and VLAN Normalization 208
 - VLAN Translation 210
 - VLAN Normalization 214
- Summary 223

Chapter 7 Centrally Routed Bridging 225

- Introducing Integrated Routing and Bridging and CRB Design 225
- Configuring a Centrally Routed Bridging EVPN VXLAN Fabric 228
- Validating and Understanding EVPN Route Exchange in a CRB Fabric 238
- Importance of “Sticky” MACs for Virtual Gateway and IRB Addresses 250
- Historical (and Present Day) Relevance of proxy-macip-advertisement 255
- Packet Walk for Hosts in Different Subnets 263
 - Control Plane Flow 264
 - Data Plane Flow 271
- Summary 277

Chapter 8 Edge-Routed Bridging 279

- Overview of Different Routing Models with Edge-Routed Bridging 279
- Asymmetric IRB 283
 - Configuring and Validating Asymmetric IRB 283
 - Control Plane and Data Plane with Asymmetric IRB 291
- Symmetric IRB 300
 - Configuring and Validating Symmetric IRB 300
 - Control Plane in a Symmetric IRB Design 304
 - Data Plane in a Symmetric IRB Design 313
 - Silent Hosts in a Symmetric IRB Design 319
- Summary 323

Chapter 9	Routed Overlay and Host-Routed Bridging	325
	Overview of a Routed Overlay Design	325
	Understanding EVPN Type-5 Routes and Their Use in Data Centers	326
	Configuring and Validating Routed Overlay	329
	Host-Routed Bridging	340
	Summary	352
Chapter 10	DHCP in EVPN VXLAN Fabrics	353
	A DHCP Refresher	353
	DHCP in a Bridged Overlay Fabric	355
	DHCP in an Edge-Routed Bridging Fabric	361
	DHCP Server in a Dedicated Services VRF	367
	Summary	375
Chapter 11	Data Center Interconnect	377
	Introduction to DCI	377
	Over-the-Top DCI	380
	Integrated Interconnect with IP Transit	394
	Stitching Two Bridged Overlay Data Centers via IP Transit	396
	Stitching EVPN Type-2 Symmetric IRB Routes	415
	Stitching EVPN Type-5 Routes	431
	Integrated Interconnect with MPLS Transit	436
	Control Plane Flow	442
	Data Plane Flow	448
	Summary	453
Chapter 12	Building Data Centers with Juniper Apstra, Part I—Apstra Foundation	455
	Introduction to Juniper Apstra	455
	Building Blocks of Apstra	457
	Onboarding Devices in Apstra	463
	Zero Touch Provisioning	464
	Manual Onboarding	475
	Creating Rack Types and Templates	481
	Creating Rack Types	482
	Creating Templates	487
	Deploying a Bridged Overlay 3-Stage Clos Fabric	489
	Lifecycle of a Device in Juniper Apstra	515
	Summary	516
Chapter 13	Building Data Centers with Juniper Apstra, Part II—Advanced Apstra Deployments	517
	Edge-Routed Bridging with Symmetric IRB	517
	Data Center Interconnect with Juniper Apstra	530
	Over-the-Top DCI	539
	<i>Adding an External Generic System</i>	539
	<i>Creating Connectivity Templates</i>	542
	<i>Configuring OTT DCI</i>	552

Integrated Interconnect 558

Interconnect Domain and MSB for Auto-derivation of Interconnect ESI 560

Creating Remote BGP EVPN Peers 561

Extending IP VRFs and Virtual Networks 562

Configuring DC2 for Integrated Interconnect 569

Validating Integrated Interconnect 571

Summary 574

Chapter 14 Building Virtual Fabrics with vJunos, Containerlab, and Juniper Apstra 575

Installing Containerlab and Building the vJunos-switch Image 575

Instantiating a Virtual Topology with vJunos-switch and Containerlab 579

Orchestrating a Virtual Fabric with Apstra 583

Summary 590

Chapter 15 Large-Scale Fabrics, Inter-VRF Routing, and Security Policies in Apstra 591

Deploying a 5-Stage Clos Fabric 591

Inter-VRF Routing in Apstra Deployments 601

Deploying Security Policies in Apstra 618

Summary 629

Acronym Legend 631

Appendix A Quick Reference Guide 635

Index 647

Introduction

My professional growth is built on the shoulders of tech and educational giants such as Jeff Doyle, Russ White, and Dinesh Dutt and their work. They have inspired generations, and just as their work inspired me, I hope this book inspires many others.

This book is a culmination of over a decade of technical learning and writing, working through customer escalations and designing, implementing, and troubleshooting small to large-scale enterprise and data center networks. And thus, this book is rooted in servant leadership and experiential learning. The goal of this book is not only to *show* but also to help you *learn* the finer details, the foundational knowledge that largely does not change as data center networks continue to evolve over time. More generally, the goal is to help you develop a mindset and a sound methodology behind building and troubleshooting data center networks.

To that end, each chapter is written with an unwavering focus on the “why.” My approach to learning new technologies has always been to understand the history behind how they evolved and what were the driving factors. In this book, I have adapted that approach to *teaching* you new technologies. Outside of focusing on the configuration that is necessary to build data centers with Junos, each chapter aims to unpack what happens behind the scenes to give you a deeper understanding of this infrastructure, while also providing historical context, wherever necessary.

By the end of this book, you will have gained expert-level knowledge about the following topics:

- The Junos CLI and how to navigate it
- The history and evolution of data centers, moving from three-tier designs to a Clos architecture, necessitated by the predominance of east-west traffic resulting from the rise of server virtualization and a shift to a microservices architecture
- The history and evolution of VXLAN, moving from a flood-and-learn model to coupling it with BGP EVPN for control plane dissemination of MAC addresses, while also providing Layer 3 reachability
- EVPN route types 1 through 5
- Building small to large-scale data centers using VXLAN with BGP EVPN and different overlay models, based on customer need, such as bridged overlay, edge-routed bridging, routed overlay, or host-routed bridging
- Connecting multiple data centers using different interconnect options such as over-the-top DCI or Integrated Interconnect with IP and MPLS transports
- Using Juniper Apstra to orchestrate data centers built using user intent with continuous validation of intent
- Using a network emulation tool such as Containerlab to build and deploy virtual lab infrastructure

While this book is not written with the intent of helping you to pass a specific certification exam, it does act as an excellent supplemental source for studying to obtain the JNCIA-DC, JNCIS-DC, JNCIP-DC, and JNCIE-DC certifications.

How This Book Is Organized

Although this book is intended to be read cover to cover, each chapter stands on its own and can be read individually, depending on your need. The first four chapters are introductory chapters, providing the proper historical context behind data center design and evolution, while also introducing the Junos CLI and how to navigate and use it. These chapters cover the following topics:

- **Chapter 1, “Introducing the Juniper Ecosystem”:** This chapter introduces the Juniper ecosystem with a focus on gaining familiarity with the Junos CLI by implementing common Layer 2 and Layer 3 features in a collapsed core design and using various `show` commands to validate user intent, including how to read and understand the MAC address table and various routing tables.
- **Chapter 2, “Overview of Data Center Architecture”:** This chapter dives into the history and evolution of data centers, focused on the driving factors that influenced and led to these changes, moving from a traditional three-tier architecture to a Clos design.

- **Chapter 3, “BGP for the Data Center”:** This chapter introduces how BGP is used for modern data centers built with a scale-out strategy using the Clos architecture.
- **Chapter 4, “VXLAN as a Network Virtualization Overlay”:** This chapter introduces VXLAN as a network overlay, elevating network services into a logical layer on top of the physical infrastructure. It also provides historical context on how VXLAN evolved from using a flood-and-learn mechanism to using BGP EVPN as a control plane to disseminate MAC address information.

Chapters 5 through 11 form the core of the book. These provide the basic building blocks of designing and operating small to large-scale data centers. Chapter 5, especially, is the main building block of this book, introducing, and diving deeper into, core VXLAN with BGP EVPN functionality; it is foundational to every chapter that comes after it. These seven chapters cover the following topics:

- **Chapter 5, “Bridged Overlay in an EVPN VXLAN Fabric”:** This chapter focuses on understanding, configuring, and validating a bridged overlay in an EVPN VXLAN fabric. It also provides a foundational understanding of how MAC addresses are learned in EVPN VXLAN fabrics and dives deeper into important aspects of such networks, such as how BUM traffic is replicated, EVPN multihoming, Route Targets, MAC mobility, loop detection, and Bidirectional Forwarding Detection.
- **Chapter 6, “MAC-VRFs”:** This chapter introduces MAC-VRFs, a construct that provides Layer 2 multitenancy in EVPN VXLAN fabrics. This chapter also explores different EVPN service types such as VLAN-Based and VLAN-Aware.
- **Chapter 7, “Centrally Routed Bridging”:** This chapter introduces the concept of integrated routed bridging and explores routing in EVPN VXLAN fabrics using a centrally routed bridging model.
- **Chapter 8, “Edge-Routed Bridging”:** This chapter builds on the previous chapter, introducing the edge-routed bridging design, while exploring the asymmetric and symmetric routing models.
- **Chapter 9, “Routed Overlay and Host-Routed Bridging”:** This chapter introduces the routed overlay and host-routed bridging designs, commonly used in infrastructures with cloud-native applications, with no requirement of Layer 2 overlays.
- **Chapter 10, “DHCP in EVPN VXLAN Fabrics”:** This chapter introduces the challenges with DHCP in such routed fabrics, diving deeper into DHCP functionality in both bridged overlay and edge-routed bridging designs, while also exploring EVPN VXLAN network designs with a dedicated services VRF where the DHCP server is located.
- **Chapter 11, “Data Center Interconnect”:** This chapter introduces how two or more data centers can be connected using the over-the-top DCI or Integrated Interconnect DCI options with IP or MPLS transports.

Chapters 12 through 15 introduce Juniper Apstra, an intent-based networking system, and dive deeper into how data centers can be deployed using Apstra. These chapters cover the following topics:

- **Chapter 12, “Building Data Centers with Juniper Apstra, Part I—Apstra Foundation”:** This chapter provides a first look at Juniper Apstra and introduces the building blocks used in designing data centers with Apstra, demonstrating how these building blocks are used to build and deploy a bridged overlay EVPN VXLAN fabric.
- **Chapter 13, “Building Data Centers with Juniper Apstra, Part II—Advanced Apstra Deployments”:** This chapter builds on the previous chapter, demonstrating how an edge-routed bridging design is built using Juniper Apstra. Various DCI options such as over-the-top DCI and Integrated Interconnect are also explored in detail in this chapter.
- **Chapter 14, “Building Virtual Fabrics with vJunos, Containerlab, and Juniper Apstra”:** This chapter introduces the need for virtual network infrastructure and how to build it using Containerlab, enabling organizations to build digital twins for network validation and pre-change and post-change testing, usually integrated in a CI/CD pipeline.
- **Chapter 15, “Large-Scale Fabrics, Inter-VRF Routing, and Security Policies in Apstra”:** The closing chapter of this book introduces and demonstrates how to build 5-stage Clos networks and the use of policies in Apstra to secure communication in EVPN VXLAN fabrics. This chapter also explores inter-VRF design options in Apstra.

BGP for the Data Center

As described in Chapter 2, “Overview of Data Center Architecture,” modern data centers are built with a *scale-out* strategy (rather than a *scale-up* strategy), with predominantly east-west traffic as opposed to the north-south traffic in the traditional three-tier architecture. This shift in strategy was prompted by many factors, including the rise of server virtualization, deployment of high-density server clusters (requiring inter-server communication), new technologies facilitating virtual machine migrations, a shift toward cloud-native applications and workloads, and, more recently, deployment of GPU clusters for artificial intelligence.

In line with this shift in strategy, data center topologies have evolved from a three-tier architecture to a 3-stage Clos architecture (and 5-stage Clos fabrics for large-scale data centers), with the need to eliminate protocols such as Spanning Tree, which made the infrastructure difficult (and more expensive) to operate and maintain due to its inherent nature of blocking redundant paths. Thus, a routing protocol was needed to convert the network natively into Layer 3, with ECMP for traffic forwarding across all available equal cost links. Operational expenditure (OPEX) considerations are equally important as well, since OPEX greatly exceeds capital expenditure (CAPEX) in most IT budgets—the goal should be using a simpler control plane, attempting to reduce control plane interaction as much as possible, and minimizing network downtime due to complex protocols.

In the past, BGP has been used primarily in service provider networks, to provide reachability between autonomous systems globally. BGP was (and still is) the protocol of the Internet, for inter-domain routing. BGP, being a path vector protocol, relies on routing based on policy (with the autonomous system number [ASN] usually acting as a tie-breaker), compared to interior gateway protocols such as Open Shortest Path First (OSPF) and Intermediate System-to-Intermediate System (IS-IS), which use path selection based on a shortest path first logic.

RFC 7938, “Use of BGP for Routing in Large-Scale Data Centers,” provides merit to using BGP with a routed design for modern data centers with a 3-stage or 5-stage Clos architecture. For VXLAN fabrics, external BGP (eBGP) can be used for both the underlay and the overlay. This chapter provides a design and implementation perspective of how BGP is adapted for the data center, specifically with eBGP for the underlay, offering the following features for large-scale deployments:

- It enables a simpler implementation, relying on TCP for underlying transport and to establish adjacency between BGP speakers.
- Although BGP is assumed to be slower to converge, with minimal design changes and well-known ASN schemes, such problems are nonexistent.
- Implementing eBGP for the underlay (for the IPv4 or IPv6 address family) and eBGP for the overlay (for the EVPN address family) using BGP groups in Junos provides a clear, vertical separation of the underlay and the overlay.
- Using BGP for both the underlay and overlay provides a simpler operational and maintenance experience. Additionally, eBGP is generally considered easier to deploy and troubleshoot, with internal BGP (iBGP) considered to be more complicated with its need for route reflectors (or confederations) and its best path selection.

- Implementing auto-discovery of BGP neighbors using link-local IPv6 addressing and leveraging RFC 8950 (which obsoletes RFC 5549) to transport IPv4 Network Layer Reachability Information (NLRI) over an IPv6 peering for the underlay enables plug-and-play behavior for any new leafs and spines.

BGP Path Hunting and ASN Scheme for Data Centers

Every BGP-speaking system requires an ASN to be assigned to exchange network reachability information with other BGP-speaking systems. An iBGP peering is defined as two BGP speakers with the same ASN peering to each other; an eBGP peering is defined as two BGP speakers with different ASNs peering to each other. For the Internet, publicly owned and assigned ASNs are used (allocated by the *Internet Assigned Numbers Authority*, or IANA), but this is dangerous for private data centers. One of the most common outages on the Internet is caused by ASN hijacking, in which an organization advertises routes from an ASN that is publicly owned by a different organization or service provider.

For this reason, IANA provides a list of 16-bit and 32-bit private ASNs that organizations can use. The 16-bit private ASNs range from 65412 to 65534, giving only 1023 available ASNs for use. To overcome this limitation, IANA offers 32-bit private ASNs for use as well, providing a much larger range, from 4200000000 to 4294967294. It is imperative that organizations building their own private data centers use ASNs from these private ranges for internal peering.

BGP is designed to route between autonomous systems, where the destination IP prefix is chosen based on the shortest number of AS hops (assuming no policy modification). These AS hops are tracked as part of a BGP attribute called AS_PATH.

In a densely interconnected topology such as a 3-stage Clos network, BGP can suffer from a problem known as *path hunting*. Path hunting occurs when BGP, on losing a route, *hunts* for reachability to the destination via all other available paths, not knowing whether the route still exists in the network or not.

Consider the 3-stage Clos network shown in Figure 3-1, with every node assigned a unique ASN from the 16-bit private ASN range.

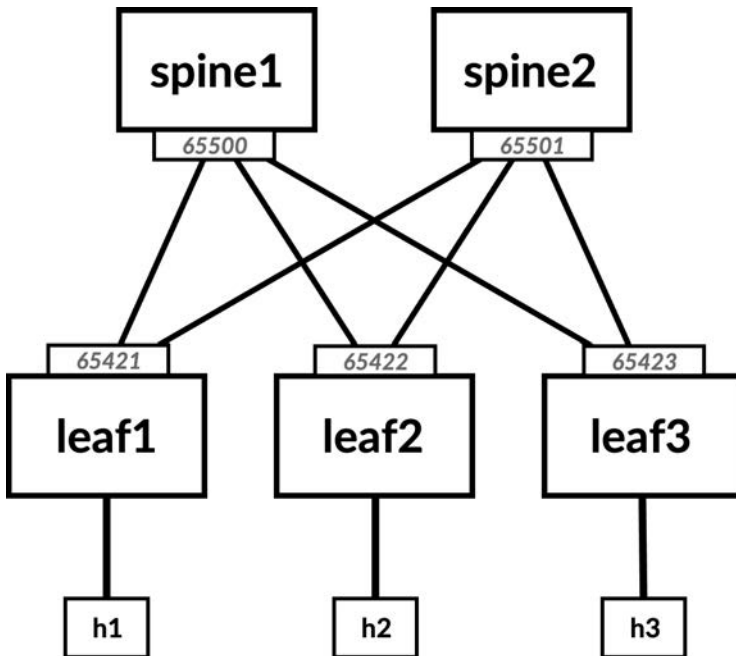


Figure 3-1 Three-stage Clos network with unique ASNs per fabric node

In this topology, leaf1 advertises a subnet x/y to spine1, as shown in Figure 3-2. This route is learned on spine1 with an AS_PATH attribute of [65421]. At the same time, the route is also advertised to spine2, and both spines advertise the route to leaf2 and leaf3.

BGP, by default, only advertises the best route to its neighbors. When leaf2 and leaf3 receive this route from both spine1 and spine2, they must elect one path as the best path. With no policy modification, the best path is chosen based on the shortest AS_PATH attribute, but in this case, the AS_PATH length is the same because the route received from spine1 will have an AS_PATH of [65500 65421] and the route received from spine2 will have an AS_PATH of [65501 65421]. Eventually, this

tie-breaker is broken by selecting the oldest path. Assuming the elected best path is via spine2 (since it is the oldest path), leaf2 and leaf3 advertise this route to their eBGP peer list, which, in this case, consists only of spine1 (the route cannot be advertised back to spine2 because it originally sent the route that was elected as the best route).

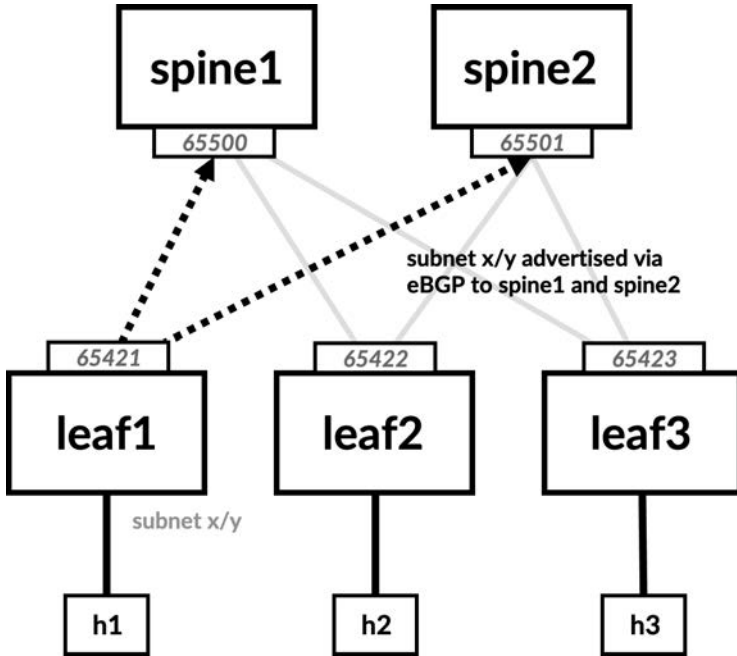


Figure 3-2 Subnet x/y advertised to spine1 and spine2 by leaf1

Thus, spine1 receives this route back from leaf2 and leaf3. At this point, spine1 has multiple paths available to reach subnet x/y advertised by leaf1; however, only the direct path (via leaf1) is selected as the best path, since it has the shortest AS_PATH length (again, assuming there are no policy modifications), as shown in Figure 3-3.

Route table on spine1

- x/y -> leaf1 via [65421] *
- > leaf2 via [65422 65501 65421]
- > leaf3 via [65423 65501 65421]

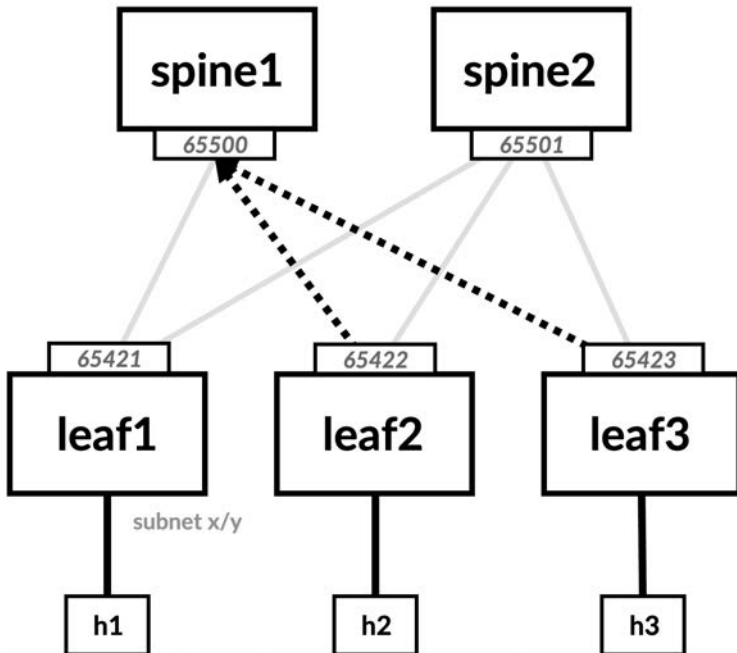


Figure 3-3 Routing table on spine1 showing all available paths for subnet x/y

When spine1 loses its best path to subnet x/y, which is via leaf1 (leaf1 goes down or withdraws the route), it hunts for an alternate best path from all available paths. At the same time, spine1 also sends a BGP withdraw to its neighbors, informing them of the lost route via leaf1 for subnet x/y. Eventually, once all withdraws have converged and the subnet has been fully purged from the network, spine1 has no available paths for it, and the route is removed from its routing table.

While this path-hunting behavior might appear to be a minor problem, it becomes increasingly problematic as the fabric size increases with more leaves, creating many alternate paths to hunt through. Thus, to avoid this problem, and to speed up BGP convergence, either of the following two methodologies can be followed, with the same end goal of ensuring that the spines do not learn alternate, suboptimal routes reflected from other leaves:

- Use an ASN scheme, leveraging eBGP's built-in loop-prevention mechanism of dropping updates that include its own ASN in its AS_PATH list. This is the default BGP behavior, and you do need to configure any additional policies for this.
- Use routing policies to prevent spines from accepting routes that were originally advertised by any other spine.

This ASN scheme is represented in Figure 3-4.

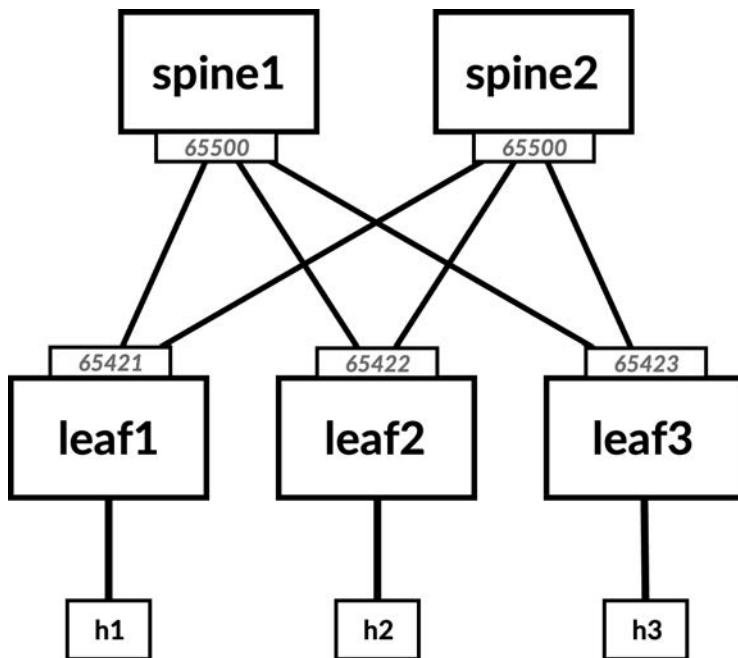


Figure 3-4 BGP ASN scheme for a 3-stage Clos fabric to avoid path hunting with same ASN on all spines

For a 5-stage Clos fabric, the ASN scheme mandates that all spines within a pod share the same ASN, but spines across pods have unique ASNs. Additionally, all leaves in each pod are assigned a unique ASN, while all superspines share the same ASN. This ASN scheme is represented in Figure 3-5.

Thus, for a 3-stage or 5-stage Clos fabric, with the ASN schemes shown in Figures 3-4 and 3-5, BGP path hunting is natively prevented.

The second methodology uses an ASN scheme in which all fabric nodes use a unique ASN, and routing policies are used to control how routes are advertised back to the spines to prevent BGP path hunting. In this case, as the spines advertise routes to the leaves, they are tagged with a BGP community using an export policy. On the leaves, an export policy is used to prevent the advertisement of routes with this BGP community from being sent back to the spines, thus preventing the existence of route state on the spines that can lead to path hunting. This is shown in Figure 3-6.

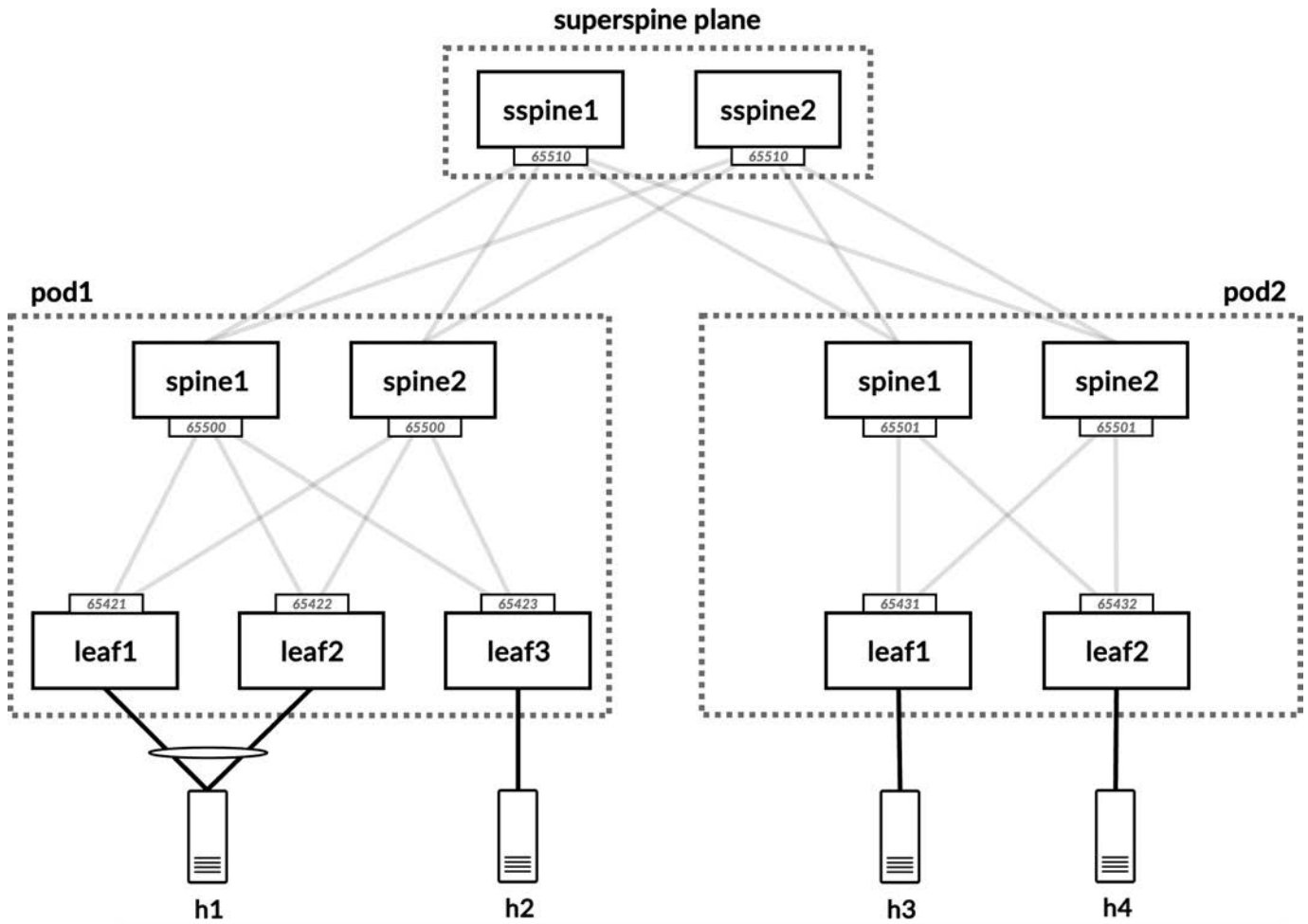


Figure 3-5 BGP ASN scheme for a 5-stage fabric to avoid path hunting

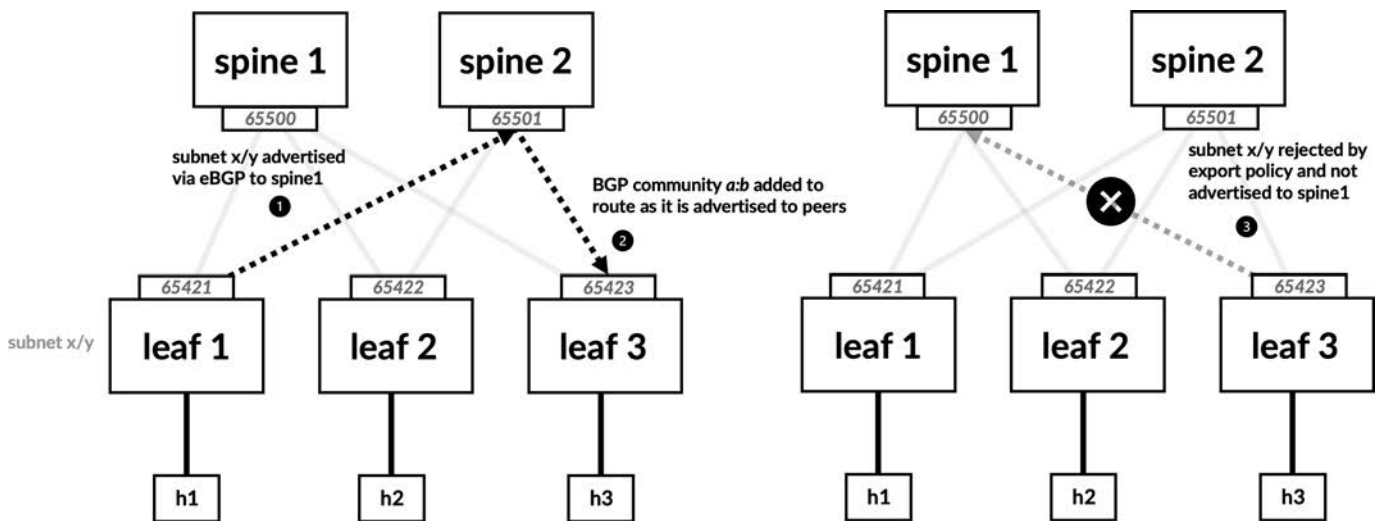


Figure 3-6 Routing policy logic to prevent path hunting

This implementation, while more complex and requiring additional operational overhead in the form of policy configuration, is necessary in certain designs where external devices are connected to the fabric for inter-VRF routing. Consider the topology shown in Figure 3-7, where the same ASN is used for both spines and a firewall is connected to leaf3 for inter-VRF routing.

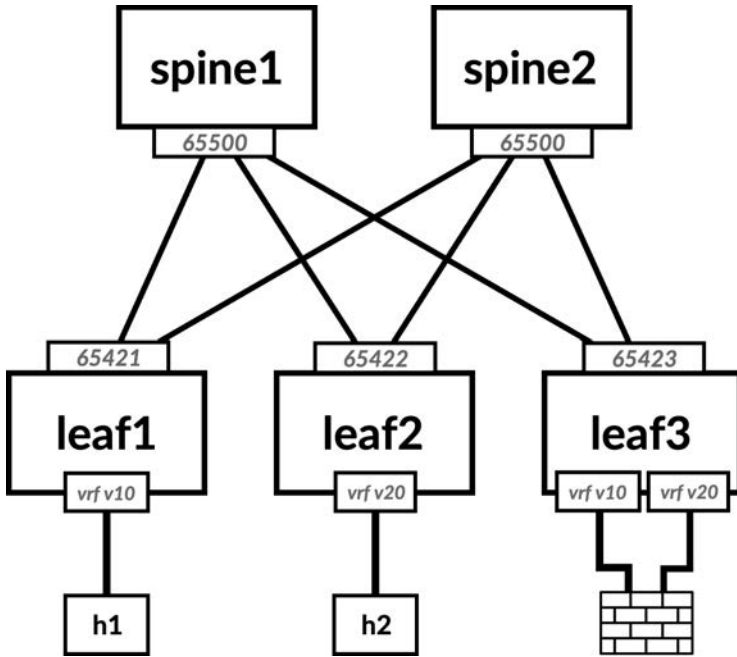


Figure 3-7 Firewall connected to fabric leaf for inter-VRF routing

In Figure 3-7, leaf1 is configured with an IP VRF *v10*, which includes an IPv4 subnet 172.16.10.0/24, and leaf2 is configured with an IP VRF *v20*, which includes an IPv4 subnet 172.16.20.0/24. The firewall has a BGP peering to leaf3 over both these IP VRFs to leak routes from one VRF to another.

The IPv4 subnet 172.16.10.0/24 is advertised by leaf1 toward leaf3, and eventually to the firewall, with an AS_PATH list of [65423 65500 65421], as shown in Figure 3-8.

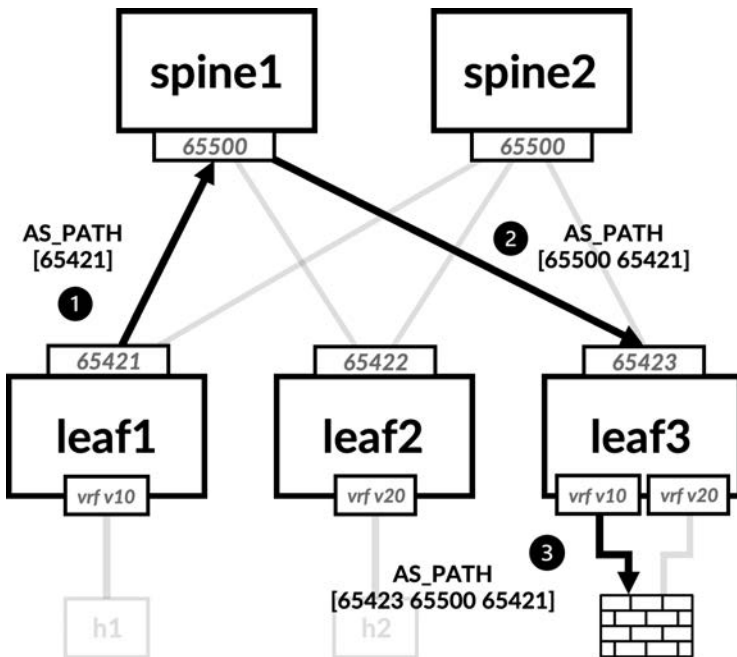


Figure 3-8 AS_PATH attribute as a prefix, originated by leaf1, is advertised toward firewall

The firewall “leaks” this route into IP VRF *v20* by advertising it to the VRF-specific BGP neighbor on leaf3. Thus, leaf3 receives this in IP VRF *v20* and advertises it to the rest of the fabric via the spines. However, when the spines receive this BGP update, they drop it because their local ASN is present in the AS_PATH list and BGP loop prevention rules indicate that such an update must be dropped. This is shown in Example 3-1, with BGP debugs on spine1.

Example 3-1 *Spines dropping BGP update due to AS loop prevention rules*

```

Jan 14 17:34:26.497233 BGP RECV 192.0.2.13+179 -> 192.0.2.101+61507
Jan 14 17:34:26.497273 BGP RECV message type 2 (Update) length 128
Jan 14 17:34:26.497369 BGP RECV Update PDU length 128
Jan 14 17:34:26.497452 BGP RECV flags 0x40 code Origin(1): IGP
Jan 14 17:34:26.497517 BGP RECV flags 0x40 code ASPath(2) length 22: 65423 65510 65423 65500 65421
Jan 14 17:34:26.497550 BGP RECV flags 0xc0 code Extended Communities(16): 2:502:502 encapsulation:vxlan(0x8) router-
mac:2c:6b:f5:75:70:f0
Jan 14 17:34:26.497561 BGP RECV flags 0x90 code MP_reach(14): AFI/SAFI 25/70
Jan 14 17:34:26.497577 BGP RECV nhop 192.0.2.13 len 4
Jan 14 17:34:26.497650 BGP RECV 5:192.0.2.14:502::0::172.16.10.0::24/248 (label field value 0x2906 [label 656, VNID
10502]) (esi 00:00:00:00:00:00:00:00)
Jan 14 17:34:26.497661 End-of-Attributes
Jan 14 17:34:26.497910 As loop detected. Rejecting update

```

snip

Figure 3-9 shows a visual representation of the same behavior.

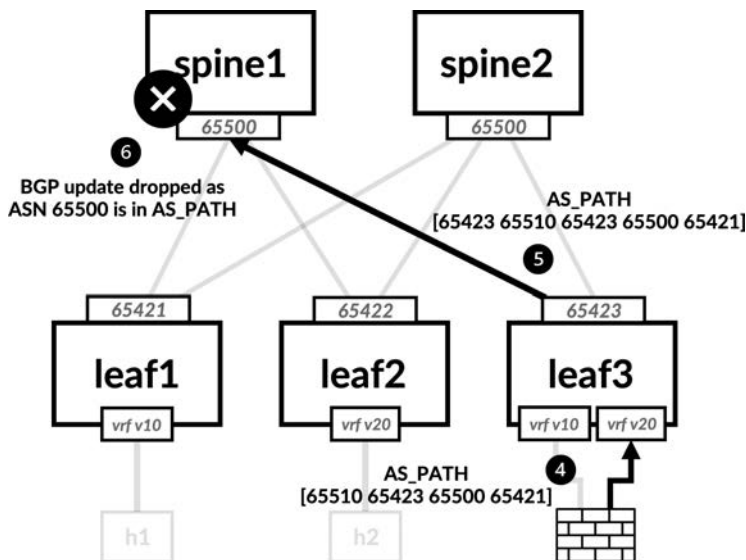


Figure 3-9 BGP update dropped on spine1 due to local ASN 65500 in AS_PATH

These problems can be circumvented by allowing the same ASN to be present in the AS_PATH attribute using several configuration options in Junos or by using an ASN scheme where each spine is assigned a unique ASN. Intent-based networking systems such as Juniper Apstra take away the complexity of implementing such an ASN scheme by automating and orchestrating the configuration of necessary policies to prevent path hunting (since that is the prevailing problem when each spine is assigned a unique ASN), with no requirement of operator intervention, while also facilitating designs as shown in Figure 3-7.

Implementing BGP for the Underlay

This section provides implementation specifics for building an eBGP underlay for an IP fabric or a VXLAN fabric using network devices running Junos. A unique ASN per fabric node design is used to demonstrate how spines can have suboptimal paths that can lead to path hunting, since the implementation of using the same ASNs on all spines in a 3-stage Clos network is straightforward and requires no demonstration. Then, routing policies are implemented to prevent path hunting. The implementation is based on the topology shown earlier in Figure 3-1.

In this network, for the underlay, each fabric-facing interface is configured as a point-to-point Layer 3 interface, as shown in Example 3-2 from the perspective of leaf1.

Example 3-2 *Point-to-point Layer 3 interface configuration on leaf1 for fabric-facing interfaces*

```
admin@leaf1# show interfaces ge-0/0/0
description "To spine1";
mtu 9100;
unit 0 {
  family inet {
    address 198.51.100.0/31;
  }
}
```

```
admin@leaf1# show interfaces ge-0/0/1
description "To spine2";
mtu 9100;
unit 0 {
  family inet {
    address 198.51.100.2/31;
  }
}
```

The goal of the underlay is to advertise the loopbacks of the *VXLAN Tunnel Endpoints (VTEPs)*, since these loopbacks are used to build end-to-end VXLAN tunnels. Thus, on each VTEP, which are the fabric leaves in this case, a loopback interface is configured, as shown on leaf1 in Example 3-3.

Example 3-3 *Loopback interface on leaf1*

```
admin@leaf1# show interfaces lo0
unit 0 {
  family inet {
    address 192.0.2.11/32;
  }
}
```

The underlay eBGP peering is between these point-to-point interfaces. Since a leaf's loopback address is sent toward other leaves via multiple spines, each leaf is expected to install multiple, equal cost paths to every other leaf's loopback address. In Junos, to enable ECMP routing, both the protocol (software) and the hardware need to be explicitly enabled to support it. In the case of BGP, this is enabled using the **multipath** knob (with the **multiple-as** configuration option if the routes received have the same AS_PATH length but different ASNs in the list). A subset of the eBGP configuration, for the underlay, is shown from the perspective of both spines and leaf1 in Example 3-4.

Example 3-4 *BGP configuration on spine1, spine2, and leaf1*

```
admin@spine1# show protocols bgp
group underlay {
  type external;
  family inet {
    unicast;
  }
  neighbor 198.51.100.0 {
    peer-as 65421;
  }
}
```

```

neighbor 198.51.100.4 {
    peer-as 65422;
}
neighbor 198.51.100.8 {
    peer-as 65423;
}
}

```

admin@spine2# **show protocols bgp**

```

group underlay {
    type external;
    family inet {
        unicast;
    }
    neighbor 198.51.100.2 {
        peer-as 65421;
    }
    neighbor 198.51.100.6 {
        peer-as 65422;
    }
    neighbor 198.51.100.10 {
        peer-as 65423;
    }
}

```

admin@leaf1# **show protocols bgp**

```

group underlay {
    type external;
    family inet {
        unicast;
    }
    export allow-loopback;
    multipath {
        multiple-as;
    }
    neighbor 198.51.100.1 {
        peer-as 65500;
    }
    neighbor 198.51.100.3 {
        peer-as 65501;
    }
}

```

Every leaf is advertising its loopback address via an export policy attached to the BGP group for the underlay, as shown in Example 3-4. The configuration of this policy is shown in Example 3-5, which enables the advertisement of direct routes in the 192.0.2.0/24 range to its eBGP peers.

Example 3-5 Policy to advertise loopbacks shown on leaf1

```

admin@leaf1# show policy-options policy-statement allow-loopback
term loopback {
    from {
        protocol direct;
        route-filter 192.0.2.0/24 orlonger;
    }
    then accept;
}
term discard {
    then reject;
}

```

Note It is important to note that each routing protocol is associated with a default routing policy in Junos. For BGP, active BGP routes are readvertised to BGP speakers without the need of an export policy, while following protocol-specific rules, such as those for iBGP neighbors, which is why there is no need for an explicit export policy on the spines to advertise received routes from a leaf to all other leaves.

With the other leaves configured in the same way, the spines can successfully form an eBGP peering with each leaf, as shown in Example 3-6.

Example 3-6 eBGP peering on spine1 and spine2 with all leaves

```
admin@spine1> show bgp summary
```

```
Threading mode: BGP I/O
```

```
Default eBGP mode: advertise - accept, receive - accept
```

```
Groups: 1 Peers: 3 Down peers: 0
```

Table	Tot Paths	Act Paths	Suppressed	History	Damp	State	Pending
inet.0	3	3	0	0	0	0	0
Peer	AS	InPkt	OutPkt	OutQ	Flaps	Last Up/Dwn	State #Active/Received/Accepted/Damped...
198.51.100.0	65421	191	189	0	0	1:24:41	Establ
inet.0: 1/1/1/0							
198.51.100.4	65422	184	182	0	0	1:21:12	Establ
inet.0: 1/1/1/0							
198.51.100.8	65423	180	179	0	0	1:19:35	Establ
inet.0: 1/1/1/0							

```
admin@spine2> show bgp summary
```

```
Threading mode: BGP I/O
```

```
Default eBGP mode: advertise - accept, receive - accept
```

```
Groups: 1 Peers: 3 Down peers: 0
```

Table	Tot Paths	Act Paths	Suppressed	History	Damp	State	Pending
inet.0	3	3	0	0	0	0	0
Peer	AS	InPkt	OutPkt	OutQ	Flaps	Last Up/Dwn	State #Active/Received/Accepted/Damped...

```

198.51.100.2      65421      194      191      0      0      1:25:52 Establ
  inet.0: 1/1/1/0
198.51.100.6      65422      183      181      0      0      1:20:57 Establ
  inet.0: 1/1/1/0
198.51.100.10     65423      180      179      0      0      1:19:21 Establ
  inet.0: 1/1/1/0

```

With the policy configured as shown in Example 3-5, and the BGP peering between the leafs and the spines in an *Established* state, the loopback address of each leaf should be learned on every other leaf in the fabric.

Consider leaf1 now, to understand how equal cost paths for another leaf's loopback address are installed. For the loopback address of leaf2, advertised by both spine1 and spine2 to leaf1, two routes are received on leaf1. Since BGP is configured with **multipath**, both routes are installed as equal cost routes in software, as shown in Example 3-7.

Example 3-7 Equal cost routes to leaf2's loopback on leaf1

```
admin@leaf1> show route table inet.0 192.0.2.12
```

```

inet.0: 7 destinations, 9 routes (7 active, 0 holddown, 0 hidden)
Limit/Threshold: 1048576/1048576 destinations
+ = Active Route, - = Last Active, * = Both

192.0.2.12/32      *[BGP/170] 02:10:44, localpref 100, from 198.51.100.1
                   AS path: 65500 65422 I, validation-state: unverified
                   to 198.51.100.1 via ge-0/0/0.0
> to 198.51.100.3 via ge-0/0/1.0
[BGP/170] 02:10:44, localpref 100
                   AS path: 65501 65422 I, validation-state: unverified
> to 198.51.100.3 via ge-0/0/1.0

```

A validation-state of *unverified*, as shown in Example 3-7, implies that the BGP route validation feature has not been configured (this is a feature to validate the origin and the path of a BGP route, to ensure that it is legitimate), and the route has been accepted but it was not validated.

These equal cost routes must also be installed in hardware. This is achieved by configuring the Packet Forwarding Engine (PFE) to install equal cost routes, and in turn, program the hardware, by applying an export policy under the **routing-options** hierarchy, as shown in Example 3-8. The policy itself simply enables per-flow load balancing. This example also demonstrates how the forwarding table, on the Routing Engine, can be viewed for a specific destination IP prefix, using the **show route forwarding-table destination [ip-address] table [table-name]** operational mode command.

Example 3-8 Equal cost routes in PFE of leaf1 with a policy for load-balancing per flow

```
admin@leaf1# show routing-options forwarding-table
```

```
export ecmp;
```

```
admin@leaf1# show policy-options policy-statement ecmp
```

```

then {
  load-balance per-flow;
}

```

```
admin@leaf1> show route forwarding-table destination 192.0.2.12/32 table default
```

```
Routing table: default.inet
```

```
Internet:
```

```

Destination      Type RtRef Next hop          Type Index  NhRef Netif

```

```

192.0.2.12/32    user    0                ulst 1048574    3
                198.51.100.1  ucst    583    4 ge-0/0/0.0
                198.51.100.3  ucst    582    4 ge-0/0/1.0

```

While the control plane and the route installation in both software and hardware are as expected on the leaves, the spines paint a different picture. If the loopback address of the leaf, advertised by spine1 to other leaves, is chosen as the best route, spine2 will receive and store all suboptimal paths in its routing table. Again, considering leaf1's loopback address as an example here, spine2 has three paths for this route, as shown in Example 3-9.

Example 3-9 *Multiple paths for leaf1's loopback address on spine2*

```
admin@spine2> show route table inet.0 192.0.2.11/32
```

```
inet.0: 10 destinations, 16 routes (10 active, 0 holddown, 0 hidden)
```

```
Limit/Threshold: 1048576/1048576 destinations
```

```
+ = Active Route, - = Last Active, * = Both
```

```

192.0.2.11/32    *[BGP/170] 15:05:38, localpref 100
                 AS path: 65421 I, validation-state: unverified
                 > to 198.51.100.2 via ge-0/0/0.0
                 [BGP/170] 00:02:39, localpref 100
                 AS path: 65422 65500 65421 I, validation-state: unverified
                 > to 198.51.100.6 via ge-0/0/1.0
                 [BGP/170] 00:01:02, localpref 100
                 AS path: 65423 65500 65421 I, validation-state: unverified
                 > to 198.51.100.10 via ge-0/0/2.0

```

This includes the direct path via leaf1, an indirect path via leaf2, and another indirect path via leaf3. Thus, in this case, if spine2 loses the direct path via leaf1, it will start path hunting through the other suboptimal paths, until the network fully converges with all withdraws processed on all fabric nodes. This problem can be addressed by applying an export policy on the spines that adds a BGP community to all advertised routes, and then using this community on the leaves to match and reject such routes from being advertised back to the spines.

In Junos, a routing policy controls the import of routes into the routing table and the export of routes from the routing table, to be advertised to neighbors. In general, a routing policy consists of terms, which include match conditions and associated actions. The routing policy on the spines is shown in Example 3-10 and includes the following two policy terms:

- **all-bgp:** Matches all BGP learned routes, accepts them, and adds a community value from the community name spine-to-leaf.
- **loopback:** Matches all direct routes in the IPv4 subnet 192.0.2.0/24. The **orlonger** configuration option matches any IPv4 address that is equal to or longer than the defined prefix length.

Example 3-10 *Policy to add a BGP community on the spines as they advertise routes to leafs*

```
admin@spine2# show policy-options policy-statement spine-to-leaf
```

```

term all-bgp {
  from protocol bgp;
  then {
    community add spine-to-leaf;
    accept;
  }
}

```

```

term loopback {
  from {
    protocol direct;
    route-filter 192.0.2.0/24 orlonger;
  }
  then {
    community add spine-to-leaf;
    accept;
  }
}

```

```

admin@spine2# show policy-options community spine-to-leaf
members 0:15;

```

Once the policy in Example 3-10 is applied as an export policy on the spines for the underlay BGP group, the leafs receive all BGP routes attached with a BGP community of value 0:15. This can be confirmed on leaf2, taking leaf1's loopback address into consideration, as shown in Example 3-11.

Example 3-11 *Leaf1's loopback address received with a BGP community of 0:15 on leaf2*

```

admin@leaf2> show route table inet.0 192.0.2.11/32 extensive

inet.0: 9 destinations, 12 routes (9 active, 0 holddown, 0 hidden)
Limit/Threshold: 1048576/1048576 destinations
192.0.2.11/32 (2 entries, 1 announced)
TSI:
KRT in-kernel 192.0.2.11/32 -> {list:198.51.100.5, 198.51.100.7}
Page 0 idx 0, (group underlay type External) Type 1 val 0x85194a0 (adv_entry)
  Advertised metrics:
    Nexthop: 198.51.100.5
    AS path: [65422] 65500 65421 I
    Communities: 0:15
  Advertise: 00000002
Path 192.0.2.11
from 198.51.100.5
Vector len 4. Val: 0
  *BGP Preference: 170/-101
    Next hop type: Router, Next hop index: 0
    Address: 0x7a46fac
    Next-hop reference count: 3, Next-hop session id: 0
    Kernel Table Id: 0
    Source: 198.51.100.5
    Next hop: 198.51.100.5 via ge-0/0/0.0
    Session Id: 0
    Next hop: 198.51.100.7 via ge-0/0/1.0, selected
    Session Id: 0
    State: <Active Ext>
    Local AS: 65422 Peer AS: 65500
    Age: 3:35

```



```

Validation State: unverified
Task: BGP_65500.198.51.100.5
Announcement bits (3): 0-KRT 1-BGP_Multi_Path 2-BGP_RT_Background
AS path: 65500 65421 I
Communities: 0:15
Accepted Multipath
Localpref: 100
Router ID: 192.0.2.101
Thread: junos-main
BGP Preference: 170/-101
Next hop type: Router, Next hop index: 577
Address: 0x77c63f4
Next-hop reference count: 5, Next-hop session id: 321
Kernel Table Id: 0
Source: 198.51.100.7
Next hop: 198.51.100.7 via ge-0/0/1.0, selected
Session Id: 321
State: <Ext>
Inactive reason: Active preferred
Local AS: 65422 Peer AS: 65501
Age: 5:30
Validation State: unverified
Task: BGP_65501.198.51.100.7
AS path: 65501 65421 I
Communities: 0:15
Accepted MultipathContrib
Localpref: 100
Router ID: 192.0.2.102
Thread: junos-main

```

On the leafs, it is now a simple matter of rejecting any route that has this community to stop it from being readvertised back to the spines. A new policy is created for this, and it is applied using an *and* operation to the existing policy that advertises the loopback address, as shown in Example 3-12 from the perspective of leaf1.

Example 3-12 *Policy on leaf1 to reject BGP routes with a community of 0:15*

```

admin@leaf1# show policy-options policy-statement leaf-to-spine
term reject-to-spine {
  from {
    protocol bgp;
    community spine-to-leaf;
  }
  then reject;
}
term accept-all {
  then accept;
}

admin@leaf1# show policy-options community spine-to-leaf
members 0:15;

```

```

admin@leaf1# show protocols bgp
group underlay {
  type external;
  family inet {
    unicast;
  }
  export ( leaf-to-spine && allow-loopback );
  multipath {
    multiple-as;
  }
  neighbor 198.51.100.1 {
    peer-as 65500;
  }
  neighbor 198.51.100.3 {
    peer-as 65501;
  }
}

```

With this policy applied on all the leaves, the spines will not learn any suboptimal paths to each of the leaf loopbacks. This is confirmed in Example 3-13, with each spine learning every leaf's loopback address via the direct path to the respective leaf.

Example 3-13 *Route to each leaf's loopback address on spine1 and spine2*

```

admin@spine1> show route table inet.0 192.0.2.11/32

inet.0: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
Limit/Threshold: 1048576/1048576 destinations
+ = Active Route, - = Last Active, * = Both

192.0.2.11/32      *[BGP/170] 15:45:36, localpref 100
                  AS path: 65421 I, validation-state: unverified
                  > to 198.51.100.0 via ge-0/0/0.0

admin@spine1> show route table inet.0 192.0.2.12/32

inet.0: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
Limit/Threshold: 1048576/1048576 destinations
+ = Active Route, - = Last Active, * = Both

192.0.2.12/32      *[BGP/170] 15:42:09, localpref 100
                  AS path: 65422 I, validation-state: unverified
                  > to 198.51.100.4 via ge-0/0/1.0

admin@spine1> show route table inet.0 192.0.2.13/32

inet.0: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
Limit/Threshold: 1048576/1048576 destinations
+ = Active Route, - = Last Active, * = Both

```

```
192.0.2.13/32    *[BGP/170] 15:40:35, localpref 100
                AS path: 65423 I, validation-state: unverified
                > to 198.51.100.8 via ge-0/0/2.0
```

```
admin@spine2> show route table inet.0 192.0.2.11/32
```

```
inet.0: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
Limit/Threshold: 1048576/1048576 destinations
+ = Active Route, - = Last Active, * = Both
```

```
192.0.2.11/32    *[BGP/170] 15:47:10, localpref 100
                AS path: 65421 I, validation-state: unverified
                > to 198.51.100.2 via ge-0/0/0.0
```

```
admin@spine2> show route table inet.0 192.0.2.12/32
```

```
inet.0: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
Limit/Threshold: 1048576/1048576 destinations
+ = Active Route, - = Last Active, * = Both
```

```
192.0.2.12/32    *[BGP/170] 15:42:18, localpref 100
                AS path: 65422 I, validation-state: unverified
                > to 198.51.100.6 via ge-0/0/1.0
```

```
admin@spine2> show route table inet.0 192.0.2.13/32
```

```
inet.0: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
Limit/Threshold: 1048576/1048576 destinations
+ = Active Route, - = Last Active, * = Both
```

```
192.0.2.13/32    *[BGP/170] 15:40:45, localpref 100
                AS path: 65423 I, validation-state: unverified
                > to 198.51.100.10 via ge-0/0/2.0
```

Junos also offers the operator a direct way to test the policy, which can be used to confirm that a leaf's locally owned loopback address is being advertised to the spines, and other loopback addresses learned via BGP are rejected. This uses the **test policy** operational mode command, as shown in Example 3-14, where only leaf1's loopback address (192.0.2.11/32) is accepted by the policy, while leaf2's and leaf3's loopback addresses, 192.0.2.12/32 and 192.0.2.13/32 respectively, are rejected by the policy.

Example 3-14 *Policy rejecting leaf2's and leaf3's loopback addresses from being advertised to the spines on leaf1*

```
admin@leaf1> test policy leaf-to-spine 192.0.2.11/32
```

```
inet.0: 9 destinations, 11 routes (9 active, 0 holddown, 0 hidden)
Limit/Threshold: 1048576/1048576 destinations
+ = Active Route, - = Last Active, * = Both
```

```
192.0.2.11/32    *[Direct/0] 1d 04:38:27
                > via lo0.0
```

```
Policy leaf-to-spine: 1 prefix accepted, 0 prefix rejected
```

```
admin@leaf1> test policy leaf-to-spine 192.0.2.12/32
```

```
Policy leaf-to-spine: 0 prefix accepted, 1 prefix rejected
```

```
admin@leaf1> test policy leaf-to-spine 192.0.2.13/32
```

```
Policy leaf-to-spine: 0 prefix accepted, 1 prefix rejected
```

With this configuration in place, the fabric underlay is successfully built, with each leaf's loopback address reachable from every other leaf, as shown in Example 3-15, while also preventing any path-hunting issues on the spines by using appropriate routing policies.

Example 3-15 Loopback reachability from leaf1

```
admin@leaf1> ping 192.0.2.12 source 192.0.2.11
PING 192.0.2.12 (192.0.2.12): 56 data bytes
64 bytes from 192.0.2.12: icmp_seq=0 ttl=63 time=3.018 ms
64 bytes from 192.0.2.12: icmp_seq=1 ttl=63 time=2.697 ms
64 bytes from 192.0.2.12: icmp_seq=2 ttl=63 time=4.773 ms
64 bytes from 192.0.2.12: icmp_seq=3 ttl=63 time=3.470 ms
^C
--- 192.0.2.12 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 2.697/3.490/4.773/0.790 ms

admin@leaf1> ping 192.0.2.13 source 192.0.2.11
PING 192.0.2.13 (192.0.2.13): 56 data bytes
64 bytes from 192.0.2.13: icmp_seq=0 ttl=63 time=2.979 ms
64 bytes from 192.0.2.13: icmp_seq=1 ttl=63 time=2.814 ms
64 bytes from 192.0.2.13: icmp_seq=2 ttl=63 time=2.672 ms
64 bytes from 192.0.2.13: icmp_seq=3 ttl=63 time=2.379 ms
^C
--- 192.0.2.13 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 2.379/2.711/2.979/0.220 ms
```

Auto-Discovered BGP Neighbors

The previous section demonstrated how to build an eBGP-based fabric underlay using point-to-point Layer 3 interfaces. This requires extensive IP management and operational maintenance as the fabric grows. An alternate, more efficient approach is to use a BGP feature called *BGP auto-discovery* (also referred to as *BGP unnumbered*), which uses link-local IPv6 addressing to automatically peer with its discovered neighbor by leveraging IPv6 Neighbor Discovery (ND). This is very beneficial for several reasons:

- It eliminates the need for IP address management of the underlay and enables plug-and-play insertion of new fabric nodes.
- It allows for easier automation of the underlay of the fabric since every fabric interface is configured the same way, with no IP addressing required. BGP, unlike IGP, is designed to peer with untrusted neighbors, and thus the default need to

specify a peer address, assign an ASN, and configure authentication for BGP peering. In a data center, which is largely a trusted environment, BGP is utilized more like an IGP, which makes automating it much easier, reducing any configuration complexity.

This section provides an implementation example of how to configure and deploy BGP auto-discovery, using packet captures for a deeper understanding of the same. The topology shown in Figure 3-10 is used to demonstrate this feature.

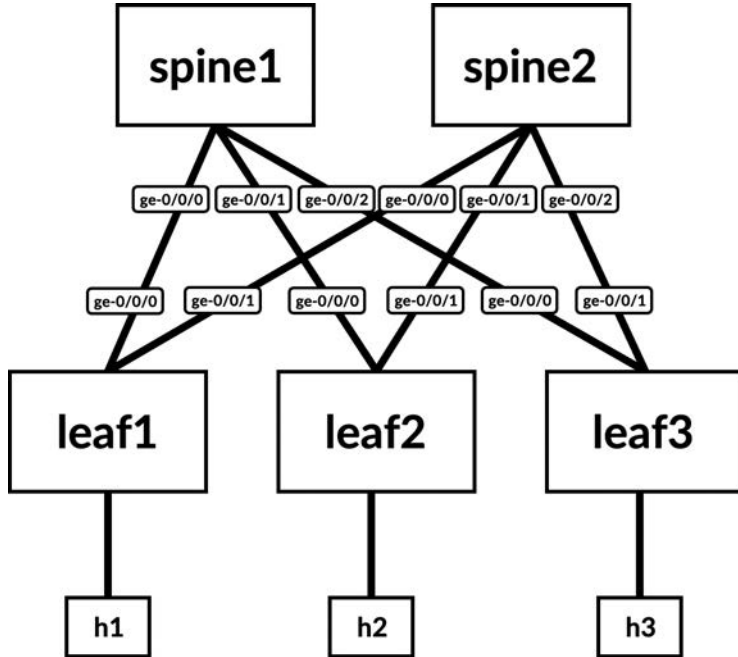


Figure 3-10 *Topology to implement BGP auto-discovered neighbors*

BGP auto-discovery relies on IPv6 Neighbor Discovery Protocol (NDP), which uses ICMPv6 messages to announce its link-local IPv6 address to its directly attached neighbors and learn the neighbors' link-local IPv6 addresses from inbound ICMPv6 messages, replacing the traditional IPv4 ARP process. More specifically, this is achieved using an ICMPv6 message type called *Router Advertisement (RA)*, which has an opcode of 134.

To enable BGP auto-discovery, the following steps must be done:

- Enable IPv6 on the fabric-facing point-to-point interfaces. The IPv4 family must be enabled as well if IPv4 traffic is expected on the interface. Even though the peering between neighbors uses IPv6, the interface can carry traffic for any address family. No IPv6 or IPv4 address is required to be configured on these interfaces.
- Enable **protocol router-advertisements** on the fabric-facing interfaces (the default RA interval is 15 seconds).
- Configure BGP to automatically discover peers using IPv6 ND by enabling the underlay group for the IPv6 unicast address family and using the **dynamic-neighbor** hierarchy to define neighbor discovery using IPv6 ND for the fabric-facing interfaces.
- Configure BGP for the IPv4 unicast address family, with the **extended-next-hop** configuration option. This allows IPv4 routes to be advertised via BGP with an IPv6 next-hop using a new BGP capability defined in RFC 8950 (which obsoletes RFC 5549) called the Extended Next Hop Encoding capability. This capability is exchanged in the BGP OPEN message.

The configuration of spine1 is shown in Example 3-16 as a reference. For the spines, since each leaf is in a different ASN, the **peer-as-list** configuration option is used to specify a list of allowed peer ASNs to which a BGP peering can be established. It is important that this peer ASN list be carefully curated, since a peering request from any other ASN (outside of this list) will be rejected.

Example 3-16 BGP auto-discovery configuration on spine1

```

admin@spine1# show interfaces
ge-0/0/0 {
    unit 0 {
        family inet;
        family inet6;
    }
}
ge-0/0/1 {
    unit 0 {
        family inet;
        family inet6;
    }
}
ge-0/0/2 {
    unit 0 {
        family inet;
        family inet6;
    }
}

admin@spine1# show protocols router-advertisement
interface ge-0/0/0.0;
interface ge-0/0/1.0;
interface ge-0/0/2.0;

admin@spine1# show protocols bgp
group auto-underlay {
    family inet {
        unicast {
            extended-nextthop;
        }
    }
    family inet6 {
        unicast;
    }
    dynamic-neighbor underlay {
        peer-auto-discovery {
            family inet6 {
                ipv6-nd;
            }
            interface ge-0/0/0.0;
            interface ge-0/0/1.0;
            interface ge-0/0/2.0;
        }
    }
    peer-as-list leafs;
}

```

Once the respective fabric interfaces are enabled with IPv6 RA, the fabric nodes discover each other's link-local IPv6 addresses. For example, leaf1 has discovered spine1's and spine2's link-local IPv6 addresses (as well as the corresponding MAC addresses) over its directly attached interfaces, as shown in Example 3-17, using the `show ipv6 neighbors` operational mode command.

Example 3-17 *IPv6 neighbors discovered using RA on leaf1*

```
admin@leaf1> show ipv6 neighbors
IPv6 Address                Linklayer Address  State    Exp  Rtr  Secure Interface
fe80::e00:b3ff:fe09:1001    0c:00:b3:09:10:01 reachable  9    yes no    ge-0/0/1.0
fe80::e00:ffff:fee3:3201    0c:00:ff:e3:32:01 reachable 14    yes no    ge-0/0/0.0
Total entries: 2
```

This process of sending Router Advertisements can be seen in the packet capture shown in Figure 3-11, from the perspective of the link between leaf1 and spine1.

No.	Time	Source	Destination	Protocol	Length	Info
4	2023-1...	fe80::e00:ffff:fee3:3201	ff02::1	ICMPv6	78	Router Advertisement from 0c:00:ff:e3:32:01
6	2023-1...	fe80::e00:ecff:fe11:c601	ff02::1	ICMPv6	78	Router Advertisement from 0c:00:ec:11:c6:01
10	2023-1...	fe80::e00:ecff:fe11:c601	fe80::e00:ffff:fee3:3201	BGP	167	OPEN Message
11	2023-1...	fe80::e00:ffff:fee3:3201	fe80::e00:ecff:fe11:c601	BGP	167	OPEN Message
13	2023-1...	fe80::e00:ffff:fee3:3201	fe80::e00:ecff:fe11:c601	BGP	105	KEEPALIVE Message
14	2023-1...	fe80::e00:ecff:fe11:c601	fe80::e00:ffff:fee3:3201	BGP	105	KEEPALIVE Message
16	2023-1...	fe80::e00:ffff:fee3:3201	fe80::e00:ecff:fe11:c601	BGP	146	UPDATE Message, UPDATE Message
17	2023-1...	fe80::e00:ecff:fe11:c601	fe80::e00:ffff:fee3:3201	BGP	146	UPDATE Message, UPDATE Message

> Frame 4: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface		0000	33 33 00 00
> Ethernet II, Src: 0c:00:ff:e3:32:01 (0c:00:ff:e3:32:01), Dst: IPv6mcast_01 (33:33:00:00:00:01)		0010	00 00 00 18
> Internet Protocol Version 6, Src: fe80::e00:ffff:fee3:3201, Dst: ff02::1		0020	ff ff fe e3
> Internet Control Message Protocol v6		0030	00 00 00 00
Type: Router Advertisement (134)		0040	00 00 00 00
Code: 0			
Checksum: 0xb754 [correct]			
[Checksum Status: Good]			
Cur hop limit: 64			
> Flags: 0x00, Prf (Default Router Preference): Medium			
Router lifetime (s): 1800			
Reachable time (ms): 0			
Retrans timer (ms): 0			
> ICMPv6 Option (Source link-layer address : 0c:00:ff:e3:32:01)			
Type: Source link-layer address (1)			
Length: 1 (8 bytes)			
Link-layer address: 0c:00:ff:e3:32:01 (0c:00:ff:e3:32:01)			

Figure 3-11 *Packet capture of ICMPv6 Router Advertisement*

Packet #4, highlighted in Figure 3-11, is an ICMPv6 Router Advertisement sent by spine1, while packet #5 is an ICMPv6 Router Advertisement sent by leaf1. Such packets are sent using the link-local IPv6 address as the source, destined to the well-known IPv6 multicast group of FF02::1. The link-local IPv6 address of leaf1's interface can be confirmed as shown in Example 3-18.

Example 3-18 *IPv6 link-local address assigned to ge-0/0/0.0 on leaf1*

```
admin@leaf1> show interfaces ge-0/0/0.0
Logical interface ge-0/0/0.0 (Index 349) (SNMP ifIndex 540)
  Flags: Up SNMP-Traps 0x4004000 Encapsulation: ENET2
  Input packets : 847
  Output packets: 857
  Protocol inet, MTU: 1500
  Max nh cache: 100000, New hold nh limit: 100000, Curr nh cnt: 0, Curr new hold cnt: 0,
  NH drop cnt: 0
  Flags: Sendbcst-pkt-to-re, Is-Primary, 0x0
```

```

Protocol inet6, MTU: 1500
Max nh cache: 100000, New hold nh limit: 100000, Curr nh cnt: 1, Curr new hold cnt: 0,
NH drop cnt: 0
  Flags: Is-Primary, 0x0
  Addresses, Flags: Is-Preferred 0x800
    Destination: fe80::/64, Local: fe80::e00:ecff:fe11:c601
Protocol multiservice, MTU: Unlimited
  Flags: Is-Primary, 0x0

```

With the link-local IPv6 addresses discovered for a given link, a TCP session can be initiated to establish BGP peering between the fabric nodes. The entire communication is IPv6 only, including the initial TCP three-way handshake and all the BGP messages exchanged between the prospective neighbors, such as the BGP OPEN and the BGP UPDATE messages shown in Figure 3-11.

The entire handshake, as well as the instantiation of the BGP session, is shown in Figure 3-12 as a reference.

No.	Time	Source	Destination	Protocol	Length	Info
4	2023-1...	fe80::e00:ffff:fee3:3201	ff02::1	ICMP...	78	Router Advertisement from 0c:00:ff:e3:32:01
6	2023-1...	fe80::e00:ecff:fe11:c601	ff02::1	ICMP...	78	Router Advertisement from 0c:00:ec:11:c6:01
7	2023-1...	fe80::e00:ecff:fe11:c601	fe80::e00:ffff:fee3:3201	TCP	98	62412 → 179 [SYN] Seq=0 Win=16384 Len=0 MSS=1440 W
8	2023-1...	fe80::e00:ffff:fee3:3201	fe80::e00:ecff:fe11:c601	TCP	98	179 → 62412 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0
9	2023-1...	fe80::e00:ecff:fe11:c601	fe80::e00:ffff:fee3:3201	TCP	86	62412 → 179 [ACK] Seq=1 Ack=1 Win=17136 Len=0 TSva
10	2023-1...	fe80::e00:ecff:fe11:c601	fe80::e00:ffff:fee3:3201	BGP	167	OPEN Message
11	2023-1...	fe80::e00:ffff:fee3:3201	fe80::e00:ecff:fe11:c601	BGP	167	OPEN Message
12	2023-1...	fe80::e00:ecff:fe11:c601	fe80::e00:ffff:fee3:3201	TCP	86	62412 → 179 [ACK] Seq=82 Ack=82 Win=17136 Len=0 TS
13	2023-1...	fe80::e00:ffff:fee3:3201	fe80::e00:ecff:fe11:c601	BGP	105	KEEPALIVE Message
14	2023-1...	fe80::e00:ecff:fe11:c601	fe80::e00:ffff:fee3:3201	BGP	105	KEEPALIVE Message

> Frame 7: 98 bytes on wire (784 bits), 98 bytes captured (784 bits)		0000	0c 00 ff e3 32 01
> Ethernet II, Src: 0c:00:ec:11:c6:01 (0c:00:ec:11:c6:01), Dst: 0c:00:ff:e3:32:01 (0c:00:ff:e3:32:01)		0010	e4 a3 00 2c 06 01
> Internet Protocol Version 6, Src: fe80::e00:ecff:fe11:c601, Dst: fe80::e00:ffff:fee3:3201		0020	ec ff fe 11 c6 01
0110 = Version: 6		0030	ff ff fe e3 32 01
> 1100 0000 = Traffic Class: 0xc0 (DSCP: CS6, ECN: Not-ECT)		0040	00 00 b0 02 40 00
.... 0000 1110 0100 1010 0011 = Flow Label: 0x0e4a3		0050	03 00 01 01 08 0a
Payload Length: 44		0060	00 00
Next Header: TCP (6)			
Hop Limit: 1			
Source Address: fe80::e00:ecff:fe11:c601			
Destination Address: fe80::e00:ffff:fee3:3201			
[Source SLAAC MAC: 0c:00:ec:11:c6:01 (0c:00:ec:11:c6:01)]			
[Destination SLAAC MAC: 0c:00:ff:e3:32:01 (0c:00:ff:e3:32:01)]			
> Transmission Control Protocol, Src Port: 62412, Dst Port: 179, Seq: 0, Len: 0			
Source Port: 62412			
Destination Port: 179			
[Stream index: 0]			
[Conversation completeness: Incomplete, DATA (15)]			
[TCP Segment Len: 0]			
Sequence Number: 0 (relative sequence number)			
Sequence Number (raw): 1058496507			
[Next Sequence Number: 1 (relative sequence number)]			
Acknowledgment Number: 0			
Acknowledgment number (raw): 0			
1011 = Header Length: 44 bytes (11)			
> Flags: 0x002 (SYN)			

Figure 3-12 Packet capture of TCP three-way handshake using IPv6 link-local addresses

In the BGP OPEN message exchanged between spine1 and leaf1, the extended next-hop capability is advertised, confirming that both devices support IPv4 NLRI encoded with an IPv6 next-hop address, as shown in Figure 3-13.

Once all leaves and spines are configured in the same way, an eBGP peering is established between the fabric nodes, as shown in Example 3-19 from the perspective of spine1 and spine2.

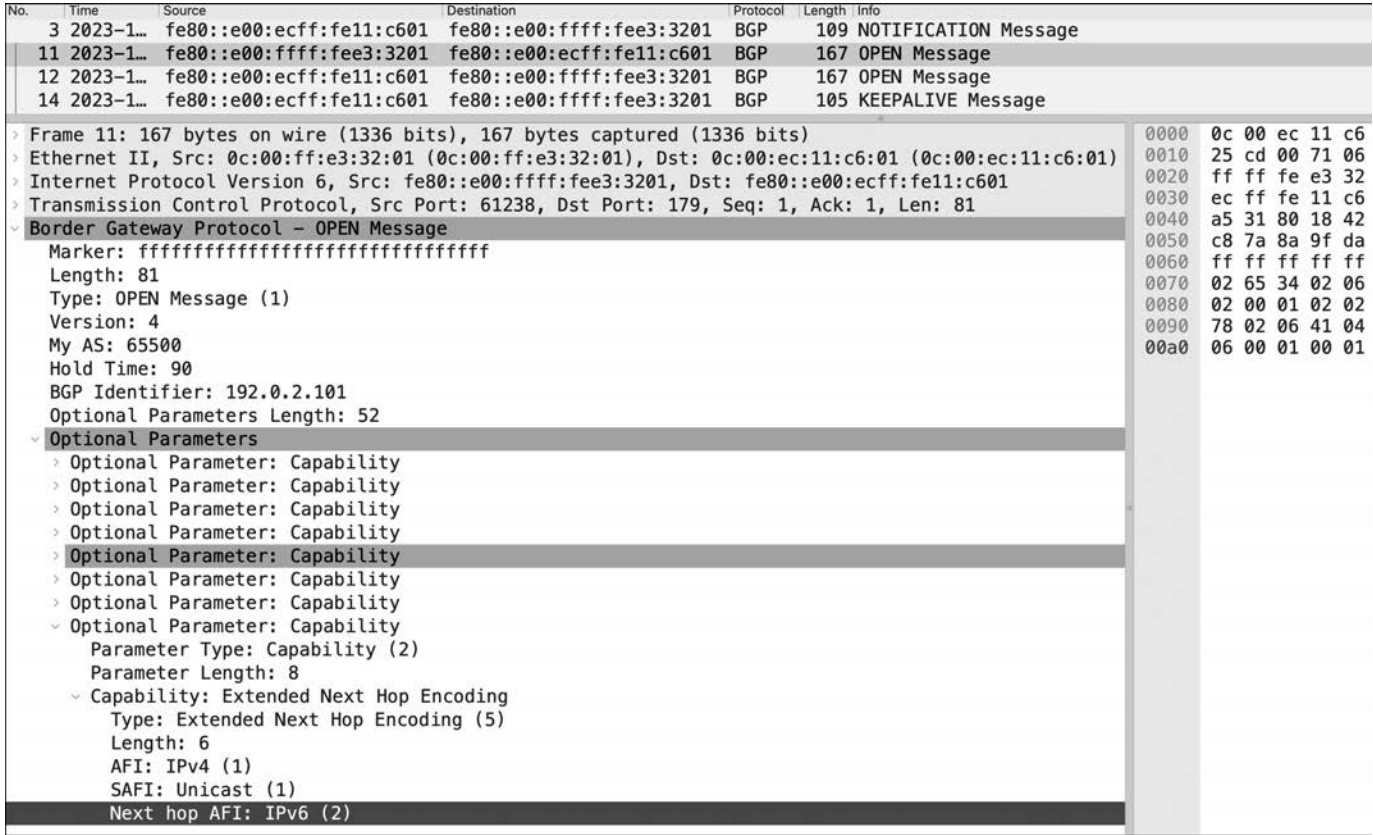


Figure 3-13 Packet capture of BGP OPEN message from spine1 advertised with extended next-hop capability

Example 3-19 Summary of BGP peers on spine1 and spine2

```

admin@spine1> show bgp summary

Threading mode: BGP I/O
Default eBGP mode: advertise - accept, receive - accept
Groups: 1 Peers: 3 Down peers: 0
Auto-discovered peers: 3
Table          Tot Paths  Act Paths Suppressed   History Damp State   Pending
inet.0
                0          0          0          0          0          0
inet6.0
                0          0          0          0          0          0
Peer           AS         InPkt   OutPkt   OutQ   Flaps Last Up/Dwn  State|#Active/Received/Accepted/Damped...
fe80::e00:36ff:fe96:af01%ge-0/0/1.0 65422     207     205     0     1:31:38 Establ
  inet.0: 0/0/0/0
  inet6.0: 0/0/0/0
fe80::e00:bdff:fed8:c901%ge-0/0/2.0 65423     206     204     0     1:31:00 Establ
  inet.0: 0/0/0/0
  inet6.0: 0/0/0/0
fe80::e00:ecff:fe11:c601%ge-0/0/0.0 65421     275     273     0     2:02:23 Establ
  inet.0: 0/0/0/0
  inet6.0: 0/0/0/0
    
```

```
admin@spine2> show bgp summary
```

```
Threading mode: BGP I/O
```

```
Default eBGP mode: advertise - accept, receive - accept
```

```
Groups: 1 Peers: 3 Down peers: 0
```

```
Auto-discovered peers: 3
```

Table	Tot Paths	Act Paths	Suppressed	History	Damp	State	Pending
inet.0	0	0	0	0	0	0	0
inet6.0	0	0	0	0	0	0	0

Peer	AS	InPkt	OutPkt	OutQ	Flaps	Last Up/Dwn	State	#Active/Received/Accepted/Damped...
fe80::e00:11ff:fe86:9602%	ge-0/0/1.0	65422	207	206	0	0	1:31:54	Establ
inet.0: 0/0/0/0								
inet6.0: 0/0/0/0								
fe80::e00:7dff:fe45:5902%	ge-0/0/0.0	65421	211	209	0	0	1:33:18	Establ
inet.0: 0/0/0/0								
inet6.0: 0/0/0/0								
fe80::e00:95ff:feec:8502%	ge-0/0/2.0	65423	206	205	0	0	1:31:16	Establ
inet.0: 0/0/0/0								
inet6.0: 0/0/0/0								

The last piece of the puzzle is how IPv4 routes are advertised over this IPv6 BGP peering. Since the BGP group is configured to use an extended next-hop for the IPv4 address family, IPv4 routes can be advertised with an IPv6 next-hop address, as shown in Figure 3-14. In this packet capture, leaf1's loopback address, 192.0.2.11/32, is advertised with an IPv6 next-hop address that matches leaf1's respective link-local IPv6 address.

No.	Time	Source	Destination	Protocol	Length	Info
17	2023-1...	fe80::e00:ffff:fee3:3201	fe80::e00:ecff:fe11:c601	BGP	146	UPDATE Message, UPDATE Message
19	2023-1...	fe80::e00:ecff:fe11:c601	fe80::e00:ffff:fee3:3201	BGP	228	UPDATE Message, UPDATE Message, UPDATE Message
> Frame 19: 228 bytes on wire (1824 bits), 228 bytes captured (1824 bits)						
> Ethernet II, Src: 0c:00:ec:11:c6:01 (0c:00:ec:11:c6:01), Dst: 0c:00:ff:e3:32:01 (0c:00:ff:e3:32:01)						
> Internet Protocol Version 6, Src: fe80::e00:ecff:fe11:c601, Dst: fe80::e00:ffff:fee3:3201						
> Transmission Control Protocol, Src Port: 179, Dst Port: 61238, Seq: 101, Ack: 161, Len: 142						
> Border Gateway Protocol - UPDATE Message						
Marker: ffffffff						
Length: 82						
Type: UPDATE Message (2)						
Withdrawn Routes Length: 0						
Total Path Attribute Length: 59						
> Path attributes						
> Path Attribute - ORIGIN: IGP						
> Path Attribute - AS_PATH: 65421						
> Path Attribute - MP_REACH_NLRI						
> Flags: 0x90, Optional, Extended-Length, Non-transitive, Complete						
Type Code: MP_REACH_NLRI (14)						
Length: 42						
Address family identifier (AFI): IPv4 (1)						
Subsequent address family identifier (SAFI): Unicast (1)						
> Next hop: IPv6=fe80::e00:ecff:fe11:c601 Link-local=fe80::e00:ecff:fe11:c601						
IPv6 Address: fe80::e00:ecff:fe11:c601						
Link-local Address: fe80::e00:ecff:fe11:c601						
Number of Subnetwork points of attachment (SNPA): 0						
> Network Layer Reachability Information (NLRI)						
> 192.0.2.11/32						
MP Reach NLRI prefix length: 32						
MP Reach NLRI IPv4 prefix: 192.0.2.11						
> Border Gateway Protocol - UPDATE Message						
> Border Gateway Protocol - UPDATE Message						

Figure 3-14 Packet capture of leaf1's IPv4 loopback address advertised with an IPv6 next-hop

Taking leaf1 as an example again, all remote leaf loopback addresses are now learned with IPv6 next-hop addresses, as shown in Example 3-20, which also confirms loopback to loopback reachability between the leaves.

Example 3-20 *IPv4 loopback addresses learned with an IPv6 next-hop*

```
admin@leaf1> show route table inet.0

inet.0: 3 destinations, 5 routes (3 active, 0 holddown, 0 hidden)
Limit/Threshold: 1048576/1048576 destinations
+ = Active Route, - = Last Active, * = Both

192.0.2.11/32      *[Direct/0] 1d 11:41:30
                  > via lo0.0
192.0.2.12/32      *[BGP/170] 00:00:27, localpref 100
                  AS path: 65500 65422 I, validation-state: unverified
                  > to fe80::e00:ffff:fee3:3201 via ge-0/0/0.0
                  [BGP/170] 00:00:27, localpref 100
                  AS path: 65500 65422 I, validation-state: unverified
                  > to fe80::e00:b3ff:fe09:1001 via ge-0/0/1.0
192.0.2.13/32      *[BGP/170] 00:00:07, localpref 100
                  AS path: 65500 65423 I, validation-state: unverified
                  > to fe80::e00:b3ff:fe09:1001 via ge-0/0/1.0
                  [BGP/170] 00:00:07, localpref 100
                  AS path: 65500 65423 I, validation-state: unverified
                  > to fe80::e00:ffff:fee3:3201 via ge-0/0/0.0

admin@leaf1> ping 192.0.2.12 source 192.0.2.11
PING 192.0.2.12 (192.0.2.12): 56 data bytes
64 bytes from 192.0.2.12: icmp_seq=0 ttl=63 time=3.290 ms
64 bytes from 192.0.2.12: icmp_seq=1 ttl=63 time=2.319 ms
64 bytes from 192.0.2.12: icmp_seq=2 ttl=63 time=2.914 ms
64 bytes from 192.0.2.12: icmp_seq=3 ttl=63 time=2.259 ms
^C
--- 192.0.2.12 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 2.259/2.696/3.290/0.428 ms

admin@leaf1> ping 192.0.2.13 source 192.0.2.11
PING 192.0.2.13 (192.0.2.13): 56 data bytes
64 bytes from 192.0.2.13: icmp_seq=0 ttl=63 time=2.849 ms
64 bytes from 192.0.2.13: icmp_seq=1 ttl=63 time=2.453 ms
64 bytes from 192.0.2.13: icmp_seq=2 ttl=63 time=2.734 ms
64 bytes from 192.0.2.13: icmp_seq=3 ttl=63 time=2.936 ms
^C
--- 192.0.2.13 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 2.453/2.743/2.936/0.182 ms
```

From the perspective of the data plane, there is no change—the underlay is purely hop-by-hop routing, with a resolution of the Layer 2 address (MAC address) required for every hop. This is already resolved using the IPv6 Router Advertisement

messages exchanged between the leafs and the spines, as shown in Example 3-17. Thus, the packet is still an IPv4 packet as shown in Figure 3-15, which is a packet capture of leaf1’s reachability to leaf2’s loopback address using the ping tool, while sourcing its own loopback address.

No.	Time	Source	Destination	Protocol	Length	Info
1	2023-1...	192.0.2.11	192.0.2.12	ICMP	98	Echo (ping) request id=0x4d46, seq=63/16128, ttl=64 (reply in 2)
2	2023-1...	192.0.2.12	192.0.2.11	ICMP	98	Echo (ping) reply id=0x4d46, seq=63/16128, ttl=63 (request in 1)
3	2023-1...	192.0.2.11	192.0.2.12	ICMP	98	Echo (ping) request id=0x4d46, seq=64/16384, ttl=64 (reply in 4)
4	2023-1...	192.0.2.12	192.0.2.11	ICMP	98	Echo (ping) reply id=0x4d46, seq=64/16384, ttl=63 (request in 3)

<ul style="list-style-type: none"> Frame 1: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) Ethernet II, Src: 0c:00:ec:11:c6:01 (0c:00:ec:11:c6:01), Dst: 0c:00:ff:e3:32:01 (0c:00:ff:e3:32:01) <ul style="list-style-type: none"> Destination: 0c:00:ff:e3:32:01 (0c:00:ff:e3:32:01) Source: 0c:00:ec:11:c6:01 (0c:00:ec:11:c6:01) Type: IPv4 (0x0800) Internet Protocol Version 4, Src: 192.0.2.11, Dst: 192.0.2.12 <ul style="list-style-type: none"> 0100 = Version: 4 ... 0101 = Header Length: 20 bytes (5) Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT) Total Length: 84 Identification: 0xa1be (41406) 000. = Flags: 0x0 ...0 0000 0000 0000 = Fragment Offset: 0 Time to Live: 64 Protocol: ICMP (1) Header Checksum: 0x54d3 [validation disabled] [Header checksum status: Unverified] Source Address: 192.0.2.11 Destination Address: 192.0.2.12 Internet Control Message Protocol 	<pre> 0000 0c 00 ff e3 32 0010 00 54 a1 be 00 0020 02 0c 08 00 23 0030 8e 59 08 09 0a 0040 16 17 18 19 1a 0050 26 27 28 29 2a 0060 36 37 </pre>
--	---

Figure 3-15 Packet capture of leaf1’s reachability test to leaf2’s loopback, using the ping tool

Summary

This chapter introduced how BGP can be adapted for a data center, with the benefits it brings, especially for larger-scale data centers. Problems such as BGP path hunting can easily be avoided by using ASN schemes for 3-stage and 5-stage Clos fabrics, or as an alternative, by using routing policies to ensure that sub-optimal paths, which can lead to path hunting, do not exist in the network.

Using eBGP as the underlay and the overlay provides a consolidated and simpler operational and maintenance experience, while continuing to provide vertical separation between the underlay and the overlay by leveraging Junos BGP groups. However, IP addressing for the underlay is operationally challenging and can get complex, very quickly, as the network grows.

With the BGP auto-discovery feature, which uses IPv6 Neighbor Discovery behind the scenes, underlay IP addressing complexity can be eliminated. This also provides an underlay framework that enables easier plug-and-play of fabric nodes, and the capability to automate the underlay without tracking any IP addressing schemes, since all fabric-facing interfaces are configured the same way.

This page intentionally left blank

Index

Symbols

? option, set command 4

3-stage Clos fabric configuration 34–39, 228–238

- BGP overlay 232–233
- BGP underlay 230–231
- EVPN 238
- IRB interfaces 236–238
- loopback routes 231–232
- physical connectivity 228–229
- point-to-point interfaces between leafs and spines 229–230
- reference topology 228–229
- underlay 230–231
- VXLAN 234–235

5-stage Clos fabric deployment 39–40, 591–600

- BGP EVPN Type-2 MAC+IP route 598
- general workflow for 591
- host address 599–600
- host communication 597
- loopback addresses of leafs 595–596
- Pod-based Templates 593–594
- Rack-based Templates 591–593
- spines 594–595
- Virtual Networks 595

802.3ad (IEEE) 129

A

accept-data configuration option 19

access layer, data center architecture 31

Ack packets (DHCP)

- definition of 354
- DHCP server in dedicated services VRF 372–374

active-server-group configuration option 528

Address Family Indicator (AFI) 75

Address Resolution Protocol. *See* ARP (Address Resolution Protocol)

addresses, IP (Internet Protocol) 359, 372

- Gateway 328
- IPv4 66, 612–617
- IPv6 60–66
- loopback 73
- Relay agent 361

addresses, MAC (Media Access Control) 74. *See also* MAC address learning

- DHCP in bridged overlay fabric 358–359
- duplicate 176
- MAC address table 21
- routed overlay 338–339
- sticky 250–255
- symmetric IRB (integrated routing and bridging) 316–318
- synchronization across ESI LAG VTEPs 132–139

advertise ipv4 unicast configuration option 343

advertisement, proxy-macip-advertisement 255–263

advertise-peer-as configuration option 614

AE (Aggregated Ethernet) interfaces 131

- configuring 15–16, 94
- validating 17

AFI (Address Family Indicator) 75

Aggregated Ethernet (AE) interfaces 131

- configuring 15–16, 94
- validating 17

aliasing 133, 144–151**All-Active mode, Ethernet Segments 129****all-bgp policy, eBGP (external BGP) underlay 54****Amazon Web Services (AWS) 380****anycast distributed gateway 361****apply-groups option, show interfaces command 26****Apstra 468. *See also* Apstra deployment inter-VRF routing; Apstra deployment security policies; Apstra edge-routed bridging with symmetric IRB deployment; Apstra Integrated Interconnect DCI deployment; Apstra Over-the-Top (OTT) DCI deployment**

- blueprints 461–463

- bridged overlay 3-stage Clos fabric, deploying 489–515

- characteristics of 456

- device lifecycle in 515–516

- device profiles 458

- Freeform 456

- interface maps 459

- internal endpoints 620–621

- key features of 456

- logical devices 457

- manual onboarding 463, 475–480

- off-box Device Agents 463, 475–480

- on-box Device Agents 463

- orchestrating virtual fabric with 583–590

- overview of 455–456

- rack types 460, 481–487

- templates 460, 487–489

- user interface 457

- ZTP (Zero Touch Provisioning) 463, 464–474

Apstra deployment inter-VRF routing 601–618

- export policy attached to BGP peering 611–612

- fabric-wide validation policy 603

- host address 607–608

- host communication 602–603, 608, 617–618

- import policy attached to BGP peering 611

- IPv4 route as EVPN Type-5 route 612–617

- reference topology 601–602

- route advertisement 614–615

- Route Target policies 604–607

- strategies for 601

- VRF tenant configuration 609–611

Apstra deployment security policies 618–629

- conflict resolution 620, 623–625

- firewall filters 622–625, 627–628

- host communication 625–626, 628–629

- internal endpoints 620–621

- policy search 620

- reference topology 618–619

- source and destination application points 619–620

Apstra edge-routed bridging with symmetric IRB deployment 517–531

- configuring 522–525

- distributed anycast IP address 520–521

- host communication 529–530

- Layer 3 VRF routing-instance 522

- Rack Types 517–518

- Virtual Networks 518–519

Apstra Integrated Interconnect DCI deployment 530–538

- configuring 558–560

- DC2 configuration 569–571

- host communication 537–538

- host-facing interface 534–537

- interconnect domain 560

- IP VRFs, extending 562–569

- Rack Types 532

- reference topology 530–532

- remote BGP EVPN peers 561–562

- Routing Zone 532

- validating 571–574

- Virtual Networks 533

- Virtual Networks, extending 562–569

Apstra Over-the-Top (OTT) DCI deployment 530–538

- configuring 552–556

- Connectivity Template 542–551

- design of 539

- external generic system 539–542

- host communication 537–538

- host-facing interface 534–537

- Rack Types 532

- reference topology 530–532

- Routing Zone 532

- Virtual Networks 533

ARP (Address Resolution Protocol) 20

- GARP (Gratuitous ARP) 264, 354

- proxy 116–120

- requests 74
- suppression 116–120
- AS_PATH attribute 44**
- ASN (autonomous system number) schemes 487–489**
 - BGP (Border Gateway Protocol) 44–49
 - Over-the-Top (OTT) data center interconnect 381
- asymmetric IRB (integrated routing and bridging) 636.**
 - See also* routed overlay
 - BGP overlay configuration 285–287
 - BGP underlay configuration 283–285
 - bridge-route-bridge model 279–283
 - control plane 291–300
 - data plane 291–300
 - definition of 279
 - host communication 290–291
 - IRB interfaces configuration 289–290
 - MAC-VRF configuration 287–289, 636
 - overview of 279–283
 - packet walk 636
 - reference topology 283
- authoritative edge devices 70**
- auto-derived route targets in EVPN VXLAN fabric 164–169**
- auto-discovered BGP neighbors 59–66**
- autonomous system number (ASN) 487–489**
- AWS (Amazon Web Services) 379**
- Azure 379**

B

- B bit, VXLAN header 73**
- BGP (Border Gateway Protocol) 22.** *See also* eBGP (external BGP) underlay
 - 5-stage Clos fabric deployment 598
 - ASN (autonomous system number) schemes 44–49
 - auto-discovery 59–66
 - BGP EVPN peering 233–234, 346, 587–588
 - BGP over L3 Connectivity Template 608–610
 - Bidirectional Forwarding Detection and 182–188
 - bridged overlay 3-stage Clos fabric deployment 503–504
 - configuration for Integrated Interconnect with MPLS Transit 437–438
 - host-routed bridging 343–346
 - iBGP (internal BGP) 43, 441–442
 - internal 43
 - overlay configuration 91–93, 232–233, 285–287
 - Over-the-Top (OTT) data center interconnect 381–382, 387–388
 - overview of 43–44
 - path hunting 44–49
 - underlay configuration 43–44, 49–59, 84–88, 230–231, 283–285
- bgp.evpn.0 table 23**
- bias, local 154–156**
- Bidirectional Forwarding Detection 182–188**
 - BGP overlay session 182–183
 - Centralized BFD 183–184
 - configuring 183
 - definition of 182
 - hardware-assisted inline BFD 183–184
 - impact of interface going down on 187–188
 - sessions 184–186
- binds key 581**
- Blueprints**
 - creating 489–496
 - definition of 461–463
- Border Gateway Protocol.** *See* BGP (Border Gateway Protocol)
- bridge domains 219–220**
- bridged overlay 3-stage Clos fabric, deploying 489–515**
 - BGP (Border Gateway Protocol) 503–504
 - Blueprints, creating 489–496
 - cabling map 498–502
 - committing changes 497
 - Connectivity Template 512–514
 - ESI LAG configuration 512–513
 - fabric nodes, mapping 499–502
 - host communication 514–515
 - MAC-VRF and IP VRF routing instances 511–512
 - resource assignment 493–497
 - routing policy 505
 - routing table 507
 - Routing Zones 508–511
 - VNs (virtual networks) 508
- bridged overlay data centers, stitching 396–415**
 - configuration on dc1-gw1 and dc2-gw1 397–398
 - DC2 leaf installing host h1's address 409
 - DCI route 406–407
 - EVPN Type-1 routes 399–404
 - EVPN Type-2 routes 404–406

- interconnection options 396
- MAC lookup 410–415
- remote DC GW (dc2-gw1) 407–409
- bridged overlay EVPN VXLAN fabric**
 - aliasing with EVPN Type-1 routes 144–151
 - Bidirectional Forwarding Detection in 182–188
 - core isolation in 157–159
 - DHCP (Dynamic Host Configuration Protocol) in 355–361
 - EVPN Type-4 routes and need for designated forwarder 139–146
 - fast convergence with EVPN Type-1 routes 151–153
 - local MAC address learning 102–111
 - loop detection 173–182
 - MAC address synchronization across ESI LAG VTEPs 132–139
 - MAC mobility 169–173
 - overlay configuration 91–97
 - overview of 81–82
 - packet flow in 97–101
 - proxy ARP and ARP suppression 116–120
 - reference topology 82
 - remote MAC address learning 112–116
 - replication of BUM traffic and EVPN Type-3 routes 120–127
 - route targets in 159–169
 - software architecture for MAC address learning 101–102
 - split horizon with EVPN Type-1 routes 153–156
 - underlay configuration 83–90
 - VPN multihoming 127–132
- bridge-route-bridge model 279–283. *See also* asymmetric IRB (integrated routing and bridging)**
- bridge-route-route-bridge model 279–283, 415. *See also* symmetric IRB (integrated routing and bridging)**
- BUM traffic bit, VXLAN header 73**
- BUM traffic, replication of 120–127. *See also* EVPN Type-4 routes; EVPN Type-5 routes**
 - configuring 121–122
 - confirming 126–127
 - EVPN Type-3 route originated by leaf1 122–123
 - flood list on leaf1 124–127
 - reference topology 120–121
 - remote leafs discovered on leaf1 123

C

- cabling maps 498–502**
- capital expenditure (CAPEX) 43**
- CCMs (continuity check messages) 178**
- CE (customer edge) 127–128**
- Centralized BFD 183–184**
- changes**
 - committing 6–7
 - comparing revisions 8
- chassis aggregated-devices ethernet device-count command 15**
- chassis aggregated-devices ethernet device-count hierarchy 131**
- Cisco Identity Services Engine (ISE) 72**
- Cisco Software-Defined Access 72**
- clear evpn duplicate-mac-suppression mac-address command 173**
- CLI (command-line interface) 1, 3, 4–11. *See also* individual commands**
- client IP address 359**
- Clos, Charles 34, 37**
- Clos fabric. *See also* 3-stage Clos fabric configuration; 5-stage Clos fabric deployment; CRB (centrally routed bridging)**
 - BGP path hunting and ASN schemes 44–49
 - folded 36
- Clos network, history and evolution of 34–37**
- collapsed spine design 40–41**
- commit and quit command 7**
- commit check command 7, 208–209**
- commit command 6–7**
- commit comment command 7**
- commit confirmed command 6**
- committing changes 6–7, 497**
- compare rollback command 8**
- /config path 7**
- configuration files 7–8**
- configuration mode, CLI (command-line interface) 4**
- configure command 4**
- conflict resolution 620, 623–625**
- connectivity fault management 178–181**
- Connectivity Template**
 - Apstra Over-the-Top (OTT) DCI deployment 542–551

bridged overlay 3-stage Clos fabric deployment
512–514

Containerlab

description of 575
installing 575–579
instantiating virtual topology with 579–582

containerlab inspect -t command 581–583

containers

host-routed bridging 351
vJunos-switch 579

/containers_data/dhcp/dhcpd.conf 466

continuity check messages (CCMs) 178

control plane

asymmetric IRB (integrated routing and bridging)
291–300
CRB (centrally routed bridging) 264–271
Integrated Interconnect with MPLS Transit 442–448
symmetric IRB (integrated routing and bridging)
304–308

copy command 26

core isolation in EVPN VXLAN fabric 157–159

core layer, data center architecture 31

CRB (centrally routed bridging) 643. *See also* CRB
EVPN VXLAN fabric configuration

control plane flow 264–271
data plane flow 271–276
design of 225–228
edge-routed bridging compared to 644–645
EVPN route exchange, validating 238–250
EVPN VXLAN fabric, configuring 228–238
overview of 225–228
packet walks for hosts in different subnets 263–264
proxy-macip-advertisement 255–263
sticky MAC addresses 250–255

CRB EVPN VXLAN fabric configuration 228–238

BGP overlay 232–233
BGP underlay 230–231
EVPN 238
IRB interfaces 236–238
loopback routes 231–232
physical connectivity 228–229
point-to-point interfaces between leafs and spines
229–230
reference topology 228–229

underlay 230–231

VXLAN 234–235

customer edge (CE) 127–128

D

daemons, L2ALD (Layer 2 Address Learning Daemon)
102, 115–118, 209

data center architecture. *See also* BGP (Border Gateway
Protocol)

3-stage fabric 34–39

5-stage fabric 39–40

access layer 31

collapsed spine design 40–41

core layer 31

distribution layer 31

FabricPath 37

history and evolution of 31–37

limitations of 32

MC-LAG (Multi-chassis Link Aggregation) 32–33

scale-out strategy 33

VXLAN as network virtualization overlay 69–79

data center interconnect. *See* DCI (data center
interconnect)

data plane

asymmetric IRB (integrated routing and bridging)
291–300

CRB (centrally routed bridging) 271–276

Integrated Interconnect with MPLS Transit 448–452

symmetric IRB (integrated routing and bridging)
313–319

database sharding 33

**DC2 configuration, Integrated Interconnect DCI
deployment** 569–571

DCI (data center interconnect) 643. *See also* Apstra
Integrated Interconnect DCI deployment; Apstra
Over-the-Top (OTT) DCI deployment

bridged overlay data centers stitched via IP Transit
406–407

bridged overlay data centers, stitching via IP Transit
396–415

definition of 377

EVPN Type-2 symmetric IRB routes, stitching
415–431

EVPN Type-5 routes, stitching 431–436

Integrated Interconnect with IP Transit 394–395,
641–643

- Integrated Interconnect with MPLS Transit 436–452, 641–643
 - Over-the-Top (OTT) 379, 380–394, 643
 - overview of 377–380
 - DDoS (distributed denial-of-service) protection policers 176–177
 - dedicated services VRF, DHCP in 367–374
 - EVPN Type-5 routes 371
 - import policy to import routes 369
 - many-to-one model 367
 - VRF configuration 368–370
 - default gateway 235
 - default routes, configuring 20
 - default-gateway do-not-advertise command 289
 - default-gateway no-gateway-community configuration option 262
 - Deploy mode 500
 - designated forwarder 139–146
 - elected 142–143
 - ES-Import community in 139–146
 - EVPN Type-4 route received and imported on leaf3 141–142
 - forwarding state of interface ae1 for BUM traffic 143–146
 - implicit policy for filtering EVPN Type-4 routes 141
 - Device Agents
 - manual creation of 463, 475–480
 - off-box 463, 475–480
 - on-box 463, 475–480
 - device lifecycle, in Apstra 515–516
 - device profiles 458, 583
 - DHCP (Dynamic Host Configuration Protocol) 640–641
 - in bridged overlay fabric 355–361
 - in dedicated services VRF 367–374
 - DORA process 353–355
 - in edge-routed bridging fabric 361–366
 - overview of 353–355
 - relay agents 354
 - Transaction IDs 354
 - ZTP (Zero Touch Provisioning) 466–467
 - digital twins
 - Containerlab installation 575–579
 - description of 575
 - instantiating with Containerlab 579–582
 - instantiating with vJunos-switch image 579–582
 - learn by breaking with 575
 - orchestrating with Apstra 583–590
 - overview of 575
 - vJunos-switch image build 575–579
 - Discover packets (DHCP)
 - definition of 354
 - DHCP in bridged overlay fabric 356–359
 - DHCP in edge-routed bridging fabric 364
 - display inheritance option, show interfaces command 26–27
 - display set option, show system login command 10
 - distributed anycast IP address 520–521
 - distributed denial-of-service (DDoS) protection policers 176–177
 - distribution layer, data center architecture 31
 - Docker
 - containers 465
 - host-routed bridging 340–343
 - vJunos-switch container, building 579
 - docker attach c1 command 350
 - docker images command 579
 - docker ps -a command 465
 - DORA process 353–355
 - Doyle, Jeff 455
 - draft-snr-bess-evpn-loop-protect IETF draft, loop protection with 181–182
 - Drain mode 500
 - duplicate MAC addresses 176
 - Dynamic Host Configuration Protocol. *See* DHCP (Dynamic Host Configuration Protocol)
 - dynamic-neighbor hierarchy 60
- ## E
-
- EAD (Ethernet Auto-Discovery) route 75–79, 147–151, 645
 - eBGP (external BGP) underlay 38, 43–44, 49–59
 - BGP configuration 50–51
 - eBGP peering 52, 383–384
 - equal cost paths for another leaf’s loopback address 53
 - equal cost routes in PFE 53–54
 - loopback interface 50
 - loopback reachability from leaf1 59
 - multiple paths of leaf1’s loopback address 54
 - point-to-point Layer 3 interface 49–50

- policy to advertise loopbacks 52
- routing policy 54–59
- ECMP (equal-cost multipath) 25, 37, 89**
- edge-routed bridging. *See* ERB (edge-routed bridging) with asymmetric IRB; ERB (edge-routed bridging) with symmetric IRB**
- edit command 5**
- edit system services command 5**
- equal-cost multipath (ECMP) 25**
- ERB (edge-routed bridging) with asymmetric IRB 636, 644. *See also* Apstra edge-routed bridging with symmetric IRB deployment; routed overlay**
 - BGP overlay configuration 285–287
 - BGP underlay configuration 283–285
 - bridge-route-bridge model 279–283
 - control plane 291–300
 - data plane 291–300
 - definition of 279
 - DHCP (Dynamic Host Configuration Protocol) in 361–366
 - host communication 290–291
 - IRB interfaces configuration 289–290
 - MAC-VRF configuration 287–289, 636
 - overview of 279–283
 - packet walk 636
 - reference topology 283
- ERB (edge-routed bridging) with symmetric IRB 637, 644**
 - bridge-route-route-bridge model 279–283
 - control plane 304–308
 - data plane 313–319
 - definition of 279
 - host communication 303–304
 - IP VRF configuration 302–303
 - MAC-VRF configuration 301–303, 637
 - overview of 279–283
 - reference topology 300
 - silent hosts 319–322
- Erwin, Edwin 34**
- ES (Ethernet Segment) 75–79, 129, 147–151, 638**
- ESI (Ethernet Segment Identifier) 129, 241, 638**
 - ESI Label 147
 - ESI LAG 94–96, 127–132, 512–513, 638
 - synchronization of MAC addresses across 132–139
- ES-Import community 139–146**
- Ethernet Auto-Discovery (EAD) route 75–79, 147–151, 645**
- Ethernet headers 73**
- Ethernet Segment (ES) 75–79, 129, 147–151, 638**
- Ethernet Segment Identifier. *See* ESI (Ethernet Segment Identifier)**
- Ethernet Segment Identifier (ESI) 129, 241**
- Ethernet VPN. *See* EVPN (Ethernet VPN)**
- ethernet-switching family 12**
- ether-options 802.3ad [*ae-number*] configuration option 131**
- evolution of data centers 31–37**
- EVPN (Ethernet VPN) 638. *See also* multihoming**
- EVPN (Ethernet VPN) route exchange 73–75, 238–250. *See also* DHCP (Dynamic Host Configuration Protocol)**
 - advantages of 75
 - ARP cache 110–111
 - ESI resolution to remote VTEPs 246
 - Ethernet Segment Identifier (ESI) 241
 - EVPN database 238–239, 243, 434
 - history and evolution of 75
 - policy to export EVPN routes 246
 - policy to import EVPN routes 245
 - Route Distinguishers 77–79
 - Route Targets 77
 - Route Types 75–79
 - service types 189–191
 - troubleshooting 246–250
 - virtual gateway address 239–241
- EVPN Instance (EVI) route, Ethernet Auto-Discovery route per 147–151**
- EVPN Type-1 routes 144–156, 243–245, 645–646**
 - aliasing with 144–151
 - bridged overlay data centers stitched via IP Transit 399–404
 - EVPN route exchange, validating 243–245
 - fast convergence with 151–153
 - split horizon with 153–156
 - VPN multihoming with 127–129
- EVPN Type-2 routes 239–241, 404–406, 415–431, 646**
- EVPN Type-3 routes 120–127, 205–207, 388–391, 645–646**
- EVPN Type-4 routes 646**
 - designated forwarder 139–146
 - need for designated forwarder with 139–146

VPN multihoming with 127–129

EVPN Type-5 routes 646

host-routed bridging 347–349

inter-VRF routing 612–617

routed overlay 326–328

stitching 431–436

EVPN VXLAN fabric, bridged overlay in. *See* bridged overlay EVPN VXLAN fabric

exec key 581

export policy

EVPN route exchange 246

inter-VRF routing 611–612

route targets in EVPN VXLAN fabric 160, 167

routed overlay 331–333

extended-nextthop configuration option 60

extensive keyword 238–240, 259

external BGP. *See* eBGP (external BGP) underlay

external generic system, Apstra OTT DCI deployment 539–542

F

fabric nodes, mapping 499–502

FabricPath 37

family inet option 11

family inet6 option 11

fast convergence with EVPN Type-1 routes 151–153

firewall filters 28–29, 622–625, 627–628

Flexible PIC Concentrator (FPC) 12, 102

flood list 124–127, 357–358

flood-and-learn mechanism 73–75

folded fabric 36

forwarding, Bidirectional Forwarding Detection 182–188

forwarding table 24–25

forwarding-options configuration hierarchy 528

FPC (Flexible PIC Concentrator) 12, 102

Free Range Routing (FRR) 91

Freeform, Apstra 456

FRR (Free Range Routing) 91

G

GARP (Gratuitous ARP) 264, 354

Gateway IP address 328

Generic Network Virtualization Encapsulation (GENEVE) 70

Generic Protocol Extension (GPE) for VXLAN (VXLAN-GPE) 72–73

Generic Routing Encapsulation (GRE) 38

GENEVE (Generic Network Virtualization Encapsulation) 70

giaddr 361

golden configuration 500

Google Cloud 379

Gratuitous ARP (GARP) 264, 354

GRE (Generic Routing Encapsulation) 38

Group Policy ID, VXLAN header 72

group-based policy extension 72

groups 26–27

H

Hadoop 69

HAL (hardware abstraction layer) 106

hardware abstraction layer (HAL) 106

hardware-assisted inline BFD 183–184

hashes, ECMP 37

headers

Ethernet 73

IP 73

UDP 73

VXLAN 71–73

history of data centers 31–37

host communication

5-stage Clos fabric deployment 597

Apstra edge-routed bridging with symmetric IRB deployment 529–530

Apstra Integrated Interconnect DCI deployment 537–538

Apstra Over-the-Top (OTT) DCI deployment 537–538

asymmetric IRB (integrated routing and bridging) 290–291

bridged overlay 3-stage Clos fabric deployment 514–515

deployment of security policies in Apstra 625–626, 628–629

inter-VRF routing 602–603, 608, 617–618

symmetric IRB (integrated routing and bridging) 303–304, 319–322

VLAN-based MAC-VRFs 198–199

host route installation 305–306

host-facing interface

Apstra Integrated Interconnect DCI deployment
534–537

Apstra Over-the-Top (OTT) DCI deployment
534–537

host-routed bridging 282, 340–351

BGP configuration on host h1 343–344

BGP configuration on leaf1 344–346

BGP peering 346

definition of 340

Docker bridge on host h1 340–343

EVPN Type-5 route for bridge address 347–348

host h1's subnet as Type-5 route 348–349

interface connection between host h1 and container
c1 350–351

IP VRF route table 349–350

reachability of container c1 to other hosts in the
fabric 351

reference topology 340

HSRP (Hot Standby Router Protocol) 31

I

IANA (Internet Assigned Numbers Authority) 44

iBGP (internal BGP) 43, 441–442

ICMP requests 271–276

Identity Services Engine (ISE) 72

IEEE 802.3ad 129

I-ESI (Integrated ESI) 394, 643

IETF, draft-snr-bess-evpn-loop-protect 181–182

image key 581

images, vJunos-switch

building 575–579

instantiating virtual topology with 579–582

IMET (Inclusive Multicast Ethernet Tag) route 75–79,
645

import policy

EVPN route exchange 245

inter-VRF routing 611

route targets in EVPN VXLAN fabric 160, 162, 167

routed overlay 334–336

import-as configuration option 167

In Service (IS) 516

Inclusive Multicast Ethernet Tag (IMET) route 75–79,
645

index, Overlay Index 328

inet.0 table 22

inet.1 table 22

inet.2 table 22

inet.3 table 23

inet6.0 table 22

inheritance, Junos groups 26–27

insert command 28–29

Instance bit, VXLAN header 72

Integrated ESI (I-ESI) 394, 643

Integrated Interconnect DCI deployment 530–538

configuring 558–560

DC2 configuration 569–571

host communication 537–538

host-facing interface 534–537

interconnect domain 560

IP VRFs, extending 562–569

Rack Types 532

reference topology 530–532

remote BGP EVPN peers 561–562

Routing Zone 532

validating 571–574

Virtual Networks 533, 562–569

Integrated Interconnect with IP Transit 394–395,
641–643

bridged overlay data centers, stitching 396–415

EVPN Type-2 symmetric IRB routes, stitching
415–431

EVPN Type-5 routes, stitching 431–436

overview of 379, 394–395

Integrated Interconnect with MPLS Transit 436–452,
641–643

BGP configuration 437–438

control plane flow 442–448

data plane flow 448–452

iBGP DCI peering 441–442

MAC-VRF configuration 439

reference topology 436–437

integrated routing and bridging. *See* IRB (integrated
routing and bridging)

interconnect domain, Apstra 560

interface maps 459

Intermediate System-to-Intermediate System (IS-IS) 43

internal BGP (iBGP) 43

Internet Assigned Numbers Authority (IANA) 44

inter-VRF routing 601–618

- export policy attached to BGP peering 611–612
- fabric-wide validation policy 603
- host address 607–608
- host communication 602–603, 608, 617–618
- import policy attached to BGP peering 611
- IPv4 route as EVPN Type-5 route 612–617
- reference topology 601–602
- route advertisement 614–615
- Route Target policies 604–607
- strategies for 601
- VRF tenant configuration 609–611

ip addr show command 350**IP addresses 359, 372**

- Gateway 328
- IPv4 66, 612–617
- IPv6 60–66
- loopback 73
- Relay agent 361

IP fabric 38**IP Prefix route 75–79****IP Transit, Integrated Interconnect with 394–395**

- bridged overlay data centers, stitching 396–415
- EVPN Type-2 symmetric IRB routes, stitching 415–431
- EVPN Type-5 routes, stitching 431–436
- overview of 379, 394–395

IP VRF configuration

- bridged overlay 3-stage Clos fabric, deploying 511–512
- extending in Apstra Integrated Interconnect DCI deployment 562–569
- host-routed bridging 349–350
- for symmetric IRB 302–303, 305–308

ip-prefix-routes hierarchy 320, 330, 330**IRB (integrated routing and bridging) 31, 279**

- asymmetric. *See* asymmetric IRB (integrated routing and bridging)
- configuring 17–19
- EVPN Type-2 routes, stitching 415–431
- EVPN Type-5 routes, stitching 431–436
- interface configuration 225, 236–238
- symmetric. *See* symmetric IRB (integrated routing and bridging)

IS (In Service) 516**IS-ACTIVE 500, 516****IS-IS 38, 43****isolation, core isolation in EVPN VXLAN fabric 157–159****IS-READY 500, 516**

J-K

JSON, displaying output in 9–10**Juniper Apstra. *See* Apstra****Junos operating systems 1. *See also* CLI (command-line interface)**

- architecture of 1–2
- building networks with. *See* network configuration
- CLI (command-line interface) 3, 4–11
- copy utility 26
- groups 26–27
- insert utility 28–29
- overview of 1
- rescue configuration 25–26

kind key 581

L

L2ALD (Layer 2 Address Learning Daemon) 102, 115–118, 209**L2ALM (Layer 2 Address Learning Manager) 102, 104–107, 265–266****l3-interface option 17****label-switched path (LSP) 23****LACP status 96–97, 131–132****Layer 2 Address Learning Daemon (L2ALD) 102, 115–118, 209****Layer 2 Address Learning Manager (L2ALM) 102****Layer 2 VNI (L2VNI) 78****Layer 3 VNI (L3VNI) 78****Layer 3 VRF routing-instance 522****learn by breaking 575****learning MAC addresses. *See* MAC address learning****least significant bit (LSB) 147****link aggregation**

- configuring 15–16
- validating 17

Link Selection 362

links key 581

Local Bias 154–156

local MAC addresses, learning 102–111

EVPN ARP cache on leaf1 110–111

host h1's MAC address in switching table on leaf1
107–108

L2ALM (Layer 2 Address Learning Manager)
104–107

port analyzer 104

sequence of events 102

software architecture for 101–102

traceoptions 102–103, 107, 108–110

log messages, duplicate MAC addresses reported in 176

logical devices 457, 583

lookup results, forwarding 435–436

loop detection 173–178

connectivity fault management 178–181

DDoS (distributed denial-of-service) protection
policers 176–177

duplicate MAC addresses 176

EVPN database 177–178

example topology 173–174

loop protection with draft-snr-bess-evpn-loop-protect
IETF draft 181–182

real-time monitoring of traffic rates 175

loopback addresses 73

5-stage Clos fabric deployment 595–596

Over-the-Top (OTT) data center interconnect
384–386

loopback routes

CRB EVPN VXLAN fabric configuration 231–232

eBGP (external BGP) underlay 50, 54

LSB (least significant bit) 147

LSP (label-switched path) 23

M

MAC (media access control) addresses 74. *See also*
MAC address learning; MAC-VRFs

asymmetric IRB (integrated routing and bridging)
291–300

bridged overlay data centers stitched via IP Transit
410–415

DHCP in bridged overlay fabric 358–359

duplicate 176

MAC address table 21

MAC mobility 169–173

MAC/IP advertisement route 75–79

Over-the-Top (OTT) data center interconnect 393

routed overlay 338–339

sticky 250–255

symmetric IRB (integrated routing and bridging)
316–318

synchronization across ESI LAG VTEPs 132–139

MAC address learning

local MAC addresses 102–111

with overlapping VLANs 221–222

remote MAC addresses 112–116

software architecture for 101–102

on translated VLAN 213–214

MAC-VRFs

bridged overlay 3-stage Clos fabric, deploying
511–512

configuration for asymmetric IRB 287–289, 636

configuration for Integrated Interconnect with MPLS
Transit 439

configuration for symmetric IRB 301–303

EVPN service types 189–191

order of operations with 200–201

Over-the-Top (OTT) data center interconnect
381–382

overview of 189

routing instance construct 189

shared tunnels with 201–204

symmetric IRB (integrated routing and bridging) 637

VLAN-Aware. *See* VLAN-Aware MAC-VRFs

VLAN-based 191–199

make command 577

manual onboarding, Apstra 463, 475–480

many-to-one model, DHCP (Dynamic Host
Configuration Protocol) 367

mass withdrawal 151–153

maximum transmission unit (MTU) 11

MC-LAG (multi-chassis link aggregation) 32–33, 94,
128

mgmt key 581

mgmt-ipv4 key 581

Microsoft Azure 379

mobility, MAC 169–173

monitor interface traffic command 175

MP-BGP (Multiprotocol BGP) 75

MPLS (Multiprotocol Label Switching) 23, 38

MPLS Transit, Integrated Interconnect with 436–452

- BGP configuration 437–438
- control plane flow 442–448
- data plane flow 448–452
- iBGP DCI peering 441–442
- MAC-VRF configuration 439
- reference topology 436–437

mpls.0 table 23

MSTP (Multiple Spanning Tree Protocol) 13

MTU (maximum transmission unit) 11

Multicast Leave Synch route 76

Multicast Membership Report Synch route 76

multi-chassis link aggregation (MC-LAG) 32–33, 94, 128

multihoming 127–132

- ESI LAG configuration 129–131, 638
- EVPN Type-4 routes and need for designated forwarder 139–146
- LACP status 131–132
- MAC address synchronization across ESI LAG VTEPs 132–139
- MC-LAG (multi-chassis link aggregation) 32–33, 94, 128
- overlay configuration for bridged overlay EVPN VXLAN fabric 94–97
- overview of 127–129

multihop no-nexthop-change configuration option 91

Multiple Spanning Tree Protocol (MSTP) 13

multiple-as configuration option, eBGP (external BGP) underlay 50

Multiprotocol BGP (MP-BGP) 75

Multiprotocol Label Switching (MPLS) 23, 38

N

naming conventions, interface 11

ND (Neighbor Discovery) 59

NDP (Neighbor Discovery Protocol) 60–66

neighbors, BGP (Border Gateway Protocol)

- auto-discovered 59–66

NETCONF 4–5

network configuration 11–25

- AE (Aggregated Ethernet) interfaces 15–16
- default routes on hosts h1 and h2 20
- forwarding table 24–25
- FPC (Flexible PIC Concentrator) 12
- general interface configuration structure 11
- IRB (integrated routing and bridging) 17–19
- Layer 2 trunk and access interface 12–13
- Layer 3 interface and OSPF configuration 23
- link aggregation configuration 15–16
- link aggregation validation 17
- MAC address table 21
- OSPF peering 23–24
- overview of 3
- rescue configuration 25–26
- routing table 22–23
- VRRP (Virtual Router Redundancy Protocol) configuration 17–19
- VRRP (Virtual Router Redundancy Protocol) validation 19–20
- VSTP (Virtual Spanning Tree Protocol) 13–15

network isolation profiles 158

Network Layer Reachability Information (NLRI) 77

Network Virtualization Overlay (NVO) 70

network virtualization overlay, VXLAN as 69–79. *See also* EVPN (Ethernet VPN)

- definition of 70
- flood-and-learn mechanism 73–75
- headers 71–73
- history and evolution of 69–70
- need for 70
- overlay origination and termination options 71
- VNI (VXLAN Network Identifier) 70
- VTEPs (VXLAN Tunnel Endpoints) 71

Network Virtualization using Generic Routing Encapsulation (NVGRE) 70

Next Protocol bit, VXLAN header 73

NLRI (Network Layer Reachability Information) 77

nodes key 581

no-dhcp-flood configuration option 528

non-contending networks 36

non-Designated Forwarders 140

normalization, VLAN 214–222

- reference topology 215
- service provider-style configuration 215–222

NVGRE (Network Virtualization using Generic Routing Encapsulation) 70

NVO (Network Virtualization Overlay) 70

O

O bit, VXLAN header 73

OAM (Operations, Administration, and Maintenance) 73

off-box Device Agents 463, 475–480

Offer packets (DHCP)
 definition of 354
 DHCP in bridged overlay fabric 359–361
 DHCP in edge-routed bridging fabric 366
 DHCP server in dedicated services VRF 372

onboarding devices, Apstra
 manual onboarding 463, 475–480
 off-box Device Agents 463, 475–480
 on-box Device Agents 463, 475–480
 ZTP (Zero Touch Provisioning) 463, 464–474

on-box Device Agents 463, 475–480

OOS (Out of Service) 516

OOS-QUARANTINED state 499, 515–516

OOS-READY state 499, 515–516

Open Shortest Path First. *See* OSPF (Open Shortest Path First)

operational expenditure (OPEX) 43

operational mode, CLI (command-line interface) 4

operations, order of 200–201

OSPF (Open Shortest Path First) 38, 43
 configuring 23
 peering 23–24

OTV (Overlay Transport Virtualization) 70

Out of Service (OOS) 516

out-of-band (OOB) connection 464

overlapping VLANs 208–210

overlay architectures
 3-stage Clos fabric 34–39
 5-stage fabric 39–40
 access layer 31
 Bidirectional Forwarding Detection and 182–183
 bridged overlay 3-stage Clos fabric, deploying 489–515
 collapsed spine design 40–41
 configuration for asymmetric IRB 285–287
 core layer 31
 CRB EVPN VXLAN fabric configuration 232–233
 distribution layer 31
 FabricPath 37

IP fabric 38
 limitations of 32
 MC-LAG (Multi-chassis Link Aggregation) 32–33
 routed overlay 325–339
 scale-out strategy 33
 VXLAN as network virtualization overlay 69–79

overlay configuration, for bridged overlay EVPN VXLAN fabric 91–97
 BGP (Border Gateway Protocol) configuration 91–93
 ESI LAG configuration 94–96
 LACP state 96–97
 leaf configuration 94

Overlay Index 328

Overlay Transport Virtualization (OTV) 70

Over-the-Top DCI (data center interconnect)
 addressing and ASN allocation scheme for 381
 BGP configuration 382–383
 BGP EVPN configuration 387–388
 disadvantages of 394
 eBGP IPv4 peering 383–384
 EVPN Type-3 route 388–391
 loopback addresses 384–386
 MAC-IP table 393
 MAC-VRF configuration on leafs 381–382
 overview of 379, 380–394, 643
 reference topology 380–381
 VXLAN tunnels 391–392

Over-the-Top DCI (data center interconnect) Apstra deployment 530–538
 configuring 552–556
 Connectivity Template 542–551
 design of 539
 external generic system 539–542
 host communication 537–538
 host-facing interface 534–537
 Rack Types 532
 reference topology 530–532
 Routing Zone 532
 Virtual Networks 533

P

P bit, VXLAN header 73

Packet Forwarding Engine (PFE) 22, 53, 102

path hunting, BGP (Border Gateway Protocol) 44–49

peer-as-list configuration option 60**peering**

BGP (Border Gateway Protocol) 52, 346

OSPF (Open Shortest Path First) 23–24

PEs (provider edges) 127–128**PFE (Packet Forwarding Engine) 22, 53, 102, 315–316****PIC (Physical Interface Card) 102****PIM (Protocol Independent Multicast) 73****ping command 19, 20**

5-stage Clos fabric deployment 597–598

Apstra edge-routed bridging with symmetric IRB deployment 529

Apstra Integrated Interconnect DCI deployment 538, 573–574

Apstra Over-the-Top (OTT) DCI deployment 538

asymmetric IRB (integrated routing and bridging) 291, 294

bridged overlay 3-stage Clos fabric deployment 515

bridged overlay data centers stitched via IP Transit 410

CRB EVPN VXLAN fabric configuration 231–232

eBGP (external BGP) underlay 59

inter-VRF routing in Apstra deployments 603, 617

Over-the-Top (OTT) data center interconnect 386–387

packet flow in bridged overlay fabric 100

proxy ARP and ARP suppression 118

security policies in Apstra 626, 628–629

symmetric IRB (integrated routing and bridging) 303–304

underlay configuration for bridged overlay EVPN VXLAN fabric 89–90

VLAN-Aware MAC-VRFs 207–208, 216, 221–222

VLAN-based MAC-VRFs 199–199

Pod-based Templates 593–594**point-to-point Layer 3 interface, eBGP (external BGP) underlay 49–50****policy**

bridged overlay 3-stage Clos fabric deployment 503–504

deploying in Apstra 618–629

eBGP (external BGP) underlay 54–59

eBGP (external BGP) underlay routing policy 54–59

EVPN route exchange 245, 246

for filtering EVPN Type-4 routes 141

inter-VRF routing 604–607, 611–612

route targets in EVPN VXLAN fabric 160, 162, 167

routed overlay 332–333, 334–336

stitched EVPN Type-2 Symmetric IRB routes 433

VLAN-based MAC-VRFs 197

port analyzer 104**port groups 583****profiles, device. See device profiles****Protocol Independent Multicast (PIM) 73****protocol router-advertisements 60****protocols bgp hierarchy 329, 330****protocols loop-detect configuration hierarchy 179****provider edges (PEs) 127–128****proxy ARP 116–120****proxy-macip-advertisement configuration option 255–263, 271–276**

Q-R

Rack Types 459, 481–487, 517–518, 532**Rack-based Templates 591–593****Rapid Spanning Tree Protocol (RSTP) 13****Ready mode 500****redistribute connected configuration option 343****reference topology**

Apstra Integrated Interconnect DCI deployment 530–532

Apstra Over-the-Top (OTT) DCI deployment 530–532

asymmetric ERB (edge-routed bridging) 283

asymmetric IRB (integrated routing and bridging) 283

bridged overlay EVPN VXLAN fabric 82

CRB EVPN VXLAN fabric, configuring 228–229

deployment of security policies in Apstra 618–619

host-routed bridging 340

Integrated Interconnect with MPLS Transit 436–437

inter-VRF routing 601–602

Over-the-Top DCI (data center interconnect) 380–381

packet flow in bridged overlay fabric 97

proxy ARP and ARP suppression 116

replication of BUM traffic and EVPN Type-3 routes 121–122

symmetric IRB (integrated routing and bridging) 300

VLAN normalization 215

VLAN-Aware MAC-VRFs 204

- VLAN-based MAC-VRFs 191–192
- ZTP (Zero Touch Provisioning) 464
- relay (DHCP) 354**
- relay agents 354, 361, 362, 372–374**
- relay configuration, DHCP in edge-routed bridging fabric 362–364**
- remote BGP EVPN peers 561–562**
- remote DC GW (dc2-gw1) 407–409**
- remote MAC addresses, learning 112–116**
 - L2ALD (Layer 2 Address Learning Daemon) 102, 114, 115–118, 209
 - L2ALM (Layer 2 Address Learning Manager) 115–118
 - sequence of events 112
 - software architecture for 101–102
 - switching table 115–118
 - traceoptions 112–114
- remote VTEPS**
 - VLAN-Aware MAC-VRFs 207–208
 - VLAN-based MAC-VRFs 195–196
- rendezvous point (RP) 73**
- replication of BUM traffic and EVPN Type-3 routes 120–127**
 - configuring 121–122
 - confirming 126–127
 - EVPN Type-3 route originated by leaf1 122–123
 - flood list on leaf1 124–127
 - reference topology 120–121
 - remote leafs discovered on leaf1 123
- Request packets (DHCP)**
 - definition of 354
 - DHCP server in dedicated services VRF 372
- request system configuration rescue save command 25**
- request system zeroize command 467**
- rescue configuration 25–26**
- resource assignment, in Apstra 493–497**
- reverse path forwarding (RPF) 22**
- RFCs (requests for comments)**
 - RFC 1997 159
 - RFC 3046 362
 - RFC 3527 362
 - RFC 4360 159
 - RFC 5107 362
 - RFC 5549 44
 - RFC 7348 70, 72, 73
 - RFC 7432 70, 75, 169, 189, 241
 - RFC 7938 38, 43
 - RFC 8365 70, 75
 - RFC 8950 44
 - RFC 9014 379, 394, 641
 - RFC 9135 279, 326
 - RFC 9136 279, 326
 - RFC 9251 76
- RIB (routing information base) 22**
- RIOT (Routing In and Out of Tunnels) 228, 279**
- rollback command 6**
- rollback rescue command 25**
- route advertisement, inter-VRF routing 614–615**
- Route Distinguishers 77–79**
- route exchange, validating 238–250**
 - ESI resolution to remote VTEPs 246
 - Ethernet Segment Identifier (ESI) 241
 - EVPN database for virtual gateway address 238–239, 243
 - EVPN Type-2 route for the virtual gateway address 239–241
 - policy to export EVPN routes 246
 - policy to import EVPN routes 245
 - troubleshooting 246–250
 - virtual gateway address 239–241
- route targets 77, 159–169**
 - auto-derived 164–169
 - configuring 160, 162
 - definition of 159
 - examples of 160–161, 163–164
 - format of 159–160
 - implicit export and import policies for 160, 162, 167
 - inter-VRF routing 604–607
 - VLAN-based MAC-VRFs 196–197
- route types, EVPN (Ethernet VPN) 75–79**
- routed overlay 326–328**
 - configuring on leaf1 330
 - EVPN Type-5 routes 326–328
 - export policy 331–333
 - host route exported into EVPN IP prefix database 332
 - host route imported into EVPN IP prefix database 336
 - host route received from server s1 on leaf1 over eBGP peering 331

- host route received in bgp.evpn.0 table 334
- host-routed bridging 340–351
- import policy 334–336
- IRB interface MAC address 338–339
- overview of 325
- policy control points 330
- route lookup 336–338, 339
- router MAC 316–318
- Router MAC community 308
- router-on-a-stick design 225, 608
- Routing In and Out of Tunnels (RIOT) 228
- routing information base (RIB) 22
- routing instance construct, MAC-VRFs 189
- Routing Zones
 - Apstra Integrated Interconnect DCI deployment 532
 - Apstra Over-the-Top (OTT) DCI deployment 532
 - bridged overlay 3-stage Clos fabric, deploying 508–511
- RP (rendezvous point) 73
- RPD 266–267
- RPF (reverse path forwarding) 22
- RSTP (Rapid Spanning Tree Protocol) 13

S

- scale-out strategy 33, 43
- scale-up strategy 43
- Secure Shell Protocol (SSH) 4–5
- Security Group Tag (SGT) 72
- segments, Ethernet 75–79, 129, 147–151, 638
- Selective Multicast Ethernet Tag (SMET) route 76
- Server Identifier Override 362
- server-group configuration, DHCP 364
- service provider style 214–222
- service types, EVPN (Ethernet VPN) 189–191
- set command 4
- set forwarding-options evpn-vxlan shared-tunnels command 193, 201–204
- set interfaces irb unit *[unit-number]* virtual-gateway-v4-mac *[mac-address]* configuration option 238
- set protocols evpn no-core-isolation command 159
- set routing-instances macvrf-v100-1 protocols evpn interconnect ? command 396
- set system login command 4
- set system services command 4
- SGT (Security Group Tag) 72

- sharding 33
- shared tunnels, with MAC-VRFs 201–204
- show | compare command 6, 10–11
- show arp command 271
- show arp hostname command 258
- show bfd session command 184–187
- show bgp l2vpn evpn route type prefix self-originate command 347–348
- show bgp summary command 52, 548, 594
 - auto-discovered BGP neighbors 64
 - bridged overlay 3-stage Clos fabric, deploying 503
 - host-routed bridging 346
 - underlay configuration for bridged overlay EVPN VXLAN fabric 87–88
- show bgp summary group evpn-gw command 572
- show bgp summary group overlay command 93, 286
- show bgp summary group overlay-dci command 388
- show bgp summary group underlay command 283–285
- show bridge mac-table command 452
- show bridge-domain entry command 219–220, 295–297
- show chassis aggregated-devices command 130–131
- show chassis hardware command 12
- show configuration interfaces command 599
- show ddos-protection protocols vxlan statistics command 176–177
- show ethernet-switching context-history mac-addr command 106, 114
- show ethernet-switching evpn arp-table command 110, 118, 292–293
- show ethernet-switching flood [extensive] command 124–126
- show ethernet-switching flood route bd-flood command 355
- show ethernet-switching instance *[instance-name]* vlan *[vlan-name]* detail command 116–117
- show ethernet-switching mac-ip-table command 110–111
- show ethernet-switching table command 21, 107–108, 358
 - CRB (centrally routed bridging) 242, 256–257, 269, 273
 - EVPN Type-4 routes and need for designated forwarder 135–137
 - packet flow in bridged overlay fabric 99
 - packet walks for hosts in different subnets 264
 - remote MAC addresses, learning 115

- sticky MAC addresses 252–255
- show ethernet-switching vxlan-tunnel-end-point esi command 151, 153
- show ethernet-switching vxlan-tunnel-end-point esi esi-identifier command 246
- show ethernet-switching vxlan-tunnel-end-point remote command 123, 358
- show evpn database command 108–109
- show evpn database mac-address command 238–240
 - EVPN route exchange, validating 243
 - extensive keyword 268, 406–407
 - Integrated Interconnect with MPLS Transit 442, 445
 - MAC mobility 171–172
- show evpn instance [*instance-name*] extensive command 142–146
- show evpn instance command 192, 195, 202–203
- show evpn ip-prefix-database command 612
- show evpn ip-prefix-database direction exported command 434
- show evpn ip-prefix-database direction exported prefix command 332
- show evpn ip-prefix-database direction imported prefix command 336
- show firewall command 28–29
- show firewall family inet filter ACL_VLAN_10_IN command 625–628
- show forwarding-options analyzer command 104
- show forwarding-options command 193–194
- show forwarding-options dhcp-relay command 363
- show interfaces command 26–27, 61, 229–230, 329–330, 383
- show interfaces irb command 17–19, 255, 289–290, 339, 520–521
- show interfaces vme command 468
- show ip route vrf Tenant1 command 345–350
- show ipv6 neighbors command 62
- show l2 manager ctxt-history mac-address command 104
- show l2 manager mac-address command 104
- show l2 manager mac-table command 104
- show lacp interface [*ae-number*] extensive command 131–132
- show lacp interface [*intf-name*] extensive command 96–97
- show lacp interface command 17
- show log bgp.log command 306
- show log command 106–107
- show log evpn.log | grep command 267
- show log h1-evpn.log command 113
- show log h1-l2ald.log command 114
- show log l2-learn.log | grep command 266
- show log macvrf-evpn.log command 306
- show log messages | grep DUPLICATE command 176
- show loop-detect enhanced interface command 180–181
- show mac-vrf forwarding command 195–196
- show mac-vrf forwarding mac-ip-table command 292–294, 393
- show mac-vrf forwarding mac-table command 298, 405–413
 - asymmetric IRB (integrated routing and bridging) 291–292, 298
 - Integrated Interconnect with MPLS Transit 442, 447, 448
 - VLAN-Aware MAC-VRFs 221–222
- show mac-vrf forwarding mac-table instance command 213–214
- show mac-vrf forwarding mac-table operational mode command 291–292
- show mac-vrf forwarding vxlan-tunnel-end-point esi command 448
- show mac-vrf forwarding vxlan-tunnel-end-point esi esi-identifier command 404, 410–412
- show mac-vrf forwarding vxlan-tunnel-end-point remote command 207, 391–392
- show nhdb id [*next-hop id*] extensive command 295–296
- show nhdb id command 295–297
- show policy __vrf-export-Tenant1-internal__ command 333
- show policy __vrf-import-default-switch-internal__ command 245
- show policy __vrf-import-Tenant1-internal__ command 335
- show policy-options command 84–86, 607, 611–612
- show policy-options community FROM_SPINE_FABRIC_TIER command 505
- show policy-options policy-statement allow-loopback command 52
- show policy-options policy-statement dci command 384
- show policy-options policy-statement EVPN_GW_IN command 554–556
- show policy-options policy-statement EVPN_GW_OUT command 567–569

- show policy-options policy-statement export-h1 command 369
- show policy-options policy-statement ip-to-evpn command 433
- show policy-options policy-statement leaf-to-spine command 56
- show policy-options policy-statement RoutesToExt-default-DCI-Routing-Policy command 547
- show policy-options policy-statement s1 command 331
- show policy-options policy-statement SPINE_TO_LEAF_FABRIC_OUT command 505
- show policy-options policy-statement spine-to-leaf command 54
- show protocols bgp command 50–51, 57, 61, 382–383, 437–438
- show protocols bgp group dc1 command 615
- show protocols bgp group dci-overlay command 441
- show protocols bgp group evpn-gw command 554, 566–567
- show protocols bgp group l3clos-l command 505
- show protocols bgp group l3clos-s command 503
- show protocols bgp group l3rtr command 547
- show protocols bgp group overlay command 91–92, 232–233, 285–287, 345–346
- show protocols bgp group overlay-dci command 388
- show protocols bgp group underlay command 85–87, 230–231, 283–285, 344–345
- show protocols bgp group underlay-dci command 384
- show protocols router-advertisement command 61
- show protocols vstp command 13–14
- show route advertising-protocol bgp command 389, 505
- show route advertising-protocol command 336, 614
- show route forwarding-table destination command 89, 295, 295–296, 314, 337–338, 364, 449
- show route receive-protocol bgp command 331, 615
- show route table bgp.evpn.0 advertising-protocol bgp command 256
- show route table bgp.evpn.0 command 122–123, 137–139, 258–259, 348, 572
- show route table bgp.evpn.0 evpn-mac-address command 270
- show route table command 22, 371–374
 - 5-stage Clos fabric deployment 596, 598, 599
 - Apstra edge-routed bridging with symmetric IRB deployment 530
 - Apstra Over-the-Top (OTT) DCI deployment 556–566
 - bridged overlay 3-stage Clos fabric deployment 505
 - bridged overlay data centers stitched via IP Transit 401–409
 - eBGP (external BGP) underlay 57–58
 - EVPN route exchange, validating 243–245
 - host-routed bridging 348–349
 - Integrated Interconnect with MPLS Transit 443–447, 450–452
 - inter-VRF routing in Apstra deployments 608, 612–613, 617
 - Over-the-Top (OTT) data center interconnect 386, 389
 - routed overlay 332–333, 334
 - sticky MAC addresses 250–251
 - stitched EVPN Type-2 Symmetric IRB routes 422–431, 435–436
 - symmetric IRB (integrated routing and bridging) 306–308, 317
 - VLAN-Aware MAC-VRFs 207
 - VLAN-based MAC-VRFs 196–198
- show route table inet.0 command 89
- show routing-instances command 369
 - Apstra edge-routed bridging with symmetric IRB deployment 522
 - Apstra Integrated Interconnect DCI deployment 565
 - asymmetric IRB (integrated routing and bridging) 287–289
 - EVPN Type-5 stitched routes 431–436
 - Integrated Interconnect configuration 397–398
 - Integrated Interconnect with MPLS Transit 439–440
 - inter-VRF routing in Apstra deployments 611
 - MAC-VRFs 193, 194
 - Over-the-Top (OTT) data center interconnect 382
 - stitched EVPN Type-2 Symmetric IRB routes 416–422
 - symmetric IRB (integrated routing and bridging) 301, 302–303, 305–306
 - VLAN-Aware MAC-VRFs 205–207
 - VLAN-based MAC-VRFs 193
- show routing-instances evpn-1 command 507, 565–566
- show routing-options command 85–87
- show routing-options forwarding-table command 53–54
- show run command 343
- show spanning-tree bridge command 14–15
- show spanning-tree interface command 14–15
- show system commit command 7

show system connections inet | grep command 101
show system login | display set command 10
show system login command 9–10
show system rollback compare command 9
show vlans command 12, 17
show vrrp detail command 19–20
silent hosts 121, 319–322
single ASN (autonomous system number) schemes 487
Single-Active mode, Ethernet Segments 129
SMET (Selective Multicast Ethernet Tag) route 76
software architecture for MAC address learning
 101–102
source and destination application points, Apstra
 619–620
spines
 3-stage Clos fabric 36–40
 5-stage Clos fabric 39–40, 594–595
 ASN schemes for 44–49
 collapsed spine design 39
split horizon with EVPN Type-1 routes 153–156
SSH (Secure Shell Protocol) 4–5
StackWise Virtual 32, 128
start shell command 7–8, 104
startup-config key 581
stitching bridged overlay data centers with IP Transit
 396–415
 configuration on dc1-gw1 and dc2-gw1 397–398
 DC2 leaf installing host h1's address 409
 DCI route 406–407
 EVPN Type-1 routes 399–404
 EVPN Type-2 routes 404–406
 interconnection options 396
 MAC lookup 410–415
 remote DC GW (dc2-gw1) 407–409
stitching EVPN Type-2 symmetric IRB routes 415–431
stitching EVPN Type-5 routes 431–436
“A Study of Non-Blocking Switching Networks” (Clos)
 34
Subsequent Address Family Indicator 75
superspines 39–40
suppression, ARP 116–120
SVI (switch virtual interfaces) 31
switch virtual interfaces (SVIs) 31, 225
switch-options configuration hierarchy 94
symmetric IRB (integrated routing and bridging) 637.
See also routed overlay

bridge-route-route-bridge model 279–283, 415
 control plane 304–308
 data plane 313–319
 definition of 279
 deployment in Apstra 517–531. *See also* Apstra edge-
 routed bridging with symmetric IRB deployment
 EVPN Type-2 symmetric IRB routes, stitching
 415–431
 EVPN Type-5 routes, stitching 431–436
 host communication 303–304
 IP VRF configuration 302–303
 MAC-VRF configuration 301–303, 637
 overview of 279–283
 reference topology 300
 routing between VNIs 588–590
 silent hosts 319–322
synchronization of MAC addresses across ESI LAG
 VTEPs 132–139
system login command 7

T

TCP (Transmission Control Protocol) 101
templates
 5-stage Clos fabric deployment 591–594
 Apstra Over-the-Top (OTT) DCI deployment
 542–551
 bridged overlay 3-stage Clos fabric, deploying
 512–514
 creating 487–489
 definition of 460
tenant configuration, inter-VRF routing 609–611
test policy command 58–59
TFTP file transfer, ZTP (Zero Touch Provisioning)
 468–469
top command 5
top-of-rack (ToR) switches 40
topology key 581
ToR (top-of-rack) devices 40, 340
traceoptions
 CRB (centrally routed bridging) 265
 local MAC addresses, learning 102–103, 107,
 108–110
 remote MAC addresses, learning 112–114
traceroute command 617
Transaction IDs (DHCP) 354

translation, VLAN 210–214

TRILL (Transparent Interconnections of Lots of Links) 37

troubleshooting EVPN route exchange 246–250

Type-1 routes 144–156, 243–245, 645–646

aliasing with 144–151

bridged overlay data centers stitched via IP Transit 399–404

EVPN route exchange, validating 243–245

fast convergence with 151–153

split horizon with 153–156

VPN multihoming with 127–129

Type-2 routes 239–241, 404–406, 415–431, 646

Type-3 routes 120–127, 205–207, 388–391, 645–646

Type-4 routes 646

designated forwarder 139–146

need for designated forwarder with 139–146

VPN multihoming with 127–129

Type-5 routes 646

DHCP server in dedicated services VRF 371

host-routed bridging 347–349

inter-VRF routing 612–617

routed overlay 326–328

stitching 431–436

U

UDP header 73

Undeploy 500

underlay configuration, for bridged overlay EVPN

VXLAN fabric 83–90

BGP (Border Gateway Protocol) 84–87

ECMP path 89

interface configuration 83

VTEP-to-VTEP reachability 89–90

underlay configuration, for eBGP (external BGP) 43–44, 49–59

3-stage Clos fabric 38

BGP configuration 50–51

configuration for asymmetric IRB 283–285

CRB EVPN VXLAN fabric configuration 230–231

eBGP peering 52

equal cost paths for another leaf's loopback address 53

equal cost routes in PFE 53–54

loopback interface 50

loopback reachability from leaf1 59

multiple paths of leaf1's loopback address 54

point-to-point Layer 3 interface 49–50

policy to advertise loopbacks 52

routing policy 54–59

unique ASN (autonomous system number) schemes 487

unit option 11

unnumbered (BGP). See auto-discovered BGP neighbors

up command 5

user configuration 4

user interface, Apstra 457

V

/var/db/config/ path 7

virtual data center fabrics

Containerlab installation 575–579

description of 575

instantiating with Containerlab 579–582

instantiating with vJunos-switch image 579–582

learn by breaking with 575

orchestrating with Apstra 583–590

overview of 575

vJunos-switch image build 575–579

Virtual Extensible LAN. See VXLAN (Virtual Extensible LAN)

virtual gateway address 242

Virtual Networks

5-stage Clos fabric deployment 595

Apstra edge-routed bridging with symmetric IRB deployment 518–519

Apstra Integrated Interconnect DCI deployment 533

Apstra Over-the-Top (OTT) DCI deployment 533

bridged overlay 3-stage Clos fabric, deploying 508

extending in Apstra Integrated Interconnect DCI deployment 562–569

Virtual Port Channel (vPC) 32, 128

Virtual Private LAN Service (VPLS) 69–70

Virtual Private Wire Service (VPWS) 69

Virtual Router Redundancy Protocol (VRRP) 31

configuring 17–19

validating 19–20

virtual routing and forwarding (VRF) 23

Virtual Spanning Tree Protocol (VSTP) 13–15

- Virtual Switching System (VSS)** 32, 128
- virtual-gateway-accept-data configuration option** 262
- vJunosEvolved** 575
- vJunos-switch image**
 - building 575–579
 - description of 575
 - instantiating virtual topology with 579–582
 - verifying 579
- VLAN-Aware MAC-VRFs**
 - configuring 205
 - EVPN Type-3 routes generated per VNI 205–207
 - overlapping VLANs 208–210
 - packet capture 205
 - reference topology 204
 - remote VTEPS 207–208
 - service provider style 214
 - VLAN normalization 214–222
 - VLAN translation 210–214
- VLAN-based MAC-VRFs 191–199**
 - configuring 194
 - EVPN Instances 192–193, 195
 - EVPN routes 196–198
 - host communication 198–199
 - internal import policy 197
 - reference topology 191–192
 - remote VTEPS 195–196
 - tenant isolation with 192
- vlan-rewrite configuration option, VLAN-Aware MAC-VRFs** 213
- VLANs (virtual LANs).** *See also* VLAN-Aware MAC-VRFs; VLAN-based MAC-VRFs
 - normalization 208–210, 214–222
 - overlapping 208–210
 - translation 210–214
- VNI (VXLAN Network Identifier)** 70, 588–590
- vPC (virtual Port-Channel)** 32
- VPLS (Virtual Private LAN Service)** 69–70
 - definition of 69
 - disadvantages of 69–70
- VPN multihoming 127–132**
 - ESI LAG configuration 129–131
 - EVPN Type-4 routes and need for designated forwarder 139–146
 - LACP status 131–132
 - MAC address synchronization across ESI LAG VTEPs 132–139
 - MC-LAG (multi-chassis link aggregation) 128
 - overview of 127–129
 - VPWS (Virtual Private Wire Service)** 69
 - VRF (virtual routing and forwarding) 23.** *See also* inter-VRF routing; IP VRF configuration; MAC-VRFs
 - dedicated services VRF, DHCP in 367–374
 - DHCP server in dedicated services VRF 367–374
 - host-routed bridging IP VRF route table 349–350
 - inter-VRF routing 601–618
 - Over-the-Top (OTT) data center interconnect 381–382
 - symmetric IRB (integrated routing and bridging) 302–303
 - vrf-export configuration option** 201
 - vrf-import configuration option** 330
 - vrnetlab project**
 - cloning fork of 576
 - listing directory of 576–577
 - VRRP (Virtual Router Redundancy Protocol)**
 - configuring 17–19
 - validating 19–20
 - VSS (Virtual Switching System)** 32, 128
 - VSTP (Virtual Spanning Tree Protocol)** 13–15
 - VTEPs (VXLAN Tunnel Endpoints)** 38, 50, 71
 - vtv fpc0 command** 104
 - vtvsh command** 343
 - VXLAN (Virtual Extensible LAN).** *See also* DHCP (Dynamic Host Configuration Protocol); EVPN (Ethernet VPN)
 - CRB EVPN VXLAN fabric 234–235
 - definition of 70
 - flood list 357–358
 - flood-and-learn mechanism 73–75
 - history and evolution of 69–70
 - IP headers 73
 - need for 70
 - as network virtualization overlay 69–79
 - overlay origination and termination options 71
 - Over-the-Top (OTT) data center interconnect 391–392
 - stitched EVPN Type-2 Symmetric IRB routes 431–433
 - UDP headers 73

VNI (VXLAN Network Identifier) 70

VTEPs (VXLAN Tunnel Endpoints) 71

VXLAN headers 38, 71–73

VXLAN-GPE (Generic Protocol Extension for VXLAN) 72–73

VXLAN Network Identifier (VNI) 70

VXLAN Tunnel Endpoints (VTEPs) 38, 50, 71

VXLAN-GPE (Generic Protocol Extension for VXLAN) 72–73

X-Y-Z

XML, displaying output in 9–10

YAML 575

ZTP (Zero Touch Provisioning) 463, 464–474

completed configuration 473–474

custom configuration for Junos OS 471s

DHCP configuration on ZTP server 466–467

Ethernet address of vme interface on leaf1 467–468

out-of-band (OOB) connection 464

reference topology 464

services run as Docker containers 465

TFTP file transfer 468–469

ztp.json file 469–471

ztp.json file 469–471