Troubleshooting Cisco Nexus Switches and NX-OS

Vinit Jain, CCIE No. 22854
Brad Edgeworth, CCIE No. 31574
Richard Furr, CCIE No. 9173

ciscopress.com
Special Sales

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

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

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

Feedback Information

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

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

We greatly appreciate your assistance.

Editor-in-Chief: Mark Taub
Alliances Manager, Cisco Press: Arezou Gol
Product Line Manager: Brett Bartow
Managing Editor: Sandra Schroeder
Development Editor: Marianne Bartow
Senior Project Editor: Tonya Simpson
Copy Editors: Barbara Hacha, Krista Hansing
Technical Editor(s): Ramiro Garza Rios, Matt Esau
Editorial Assistant: Vanessa Evans
Cover Designer: Chuti Prasertsith
Composition: codemantra
Indexer: Cheryl Lenser
Proofreader: Jeanine Furino
About the Authors

Vinit Jain, CCIE No. 22854 (R&S, SP, Security & DC), is a technical leader with the Cisco Technical Assistance Center (TAC) providing escalation support in areas of routing and data center technologies. Vinit is a speaker at various networking forums, including Cisco Live events globally on various topics. Prior to joining Cisco, Vinit worked as a CCIE trainer and a network consultant. In addition to his CCIEs, Vinit holds multiple certifications on programming and databases. Vinit graduated from Delhi University in Mathematics and earned his Master's in Information Technology from Kuvempu University in India. Vinit can be found on Twitter as @VinuGenie.

Brad Edgeworth, CCIE No. 31574 (R&S & SP), is a systems engineer at Cisco Systems. Brad is a distinguished speaker at Cisco Live, where he has presented on various topics. Before joining Cisco, Brad worked as a network architect and consultant for various Fortune 500 companies. Brad's expertise is based on enterprise and service provider environments with an emphasis on architectural and operational simplicity. Brad holds a Bachelor of Arts degree in Computer Systems Management from St. Edward's University in Austin, Texas. Brad can be found on Twitter as @BradEdgeworth.

Richard Furr, CCIE No. 9173 (R&S & SP), is a technical leader with the Cisco Technical Assistance Center (TAC), supporting customers and TAC teams around the world. For the past 17 years, Richard has worked for the Cisco TAC and High Touch Technical Support (HTTS) organizations, supporting service provider, enterprise, and data center environments. Richard specializes in resolving complex problems found with routing protocols, MPLS, multicast, and network overlay technologies.

About the Technical Reviewers

Ramiro Garza Rios, CCIE No. 15469 (R&S, SP, and Security), is a solutions integration architect with Cisco Advanced Services, where he plans, designs, implements, and optimizes IP NGN service provider networks. Before joining Cisco in 2005, he was a network consulting and presales engineer for a Cisco Gold Partner in Mexico, where he planned, designed, and implemented both enterprise and service provider networks.

Matt Esau, CCIE No. 18586 (R&S) is a graduate from the University of North Carolina at Chapel Hill. He currently resides in Ohio with his wife and two children, ages three and one. Matt is a Distinguished Speaker at Cisco Live. He started with Cisco in 2002 and has spent 15 years working closely with customers on troubleshooting issues and product usability. For the past eight years, he has worked in the Data Center space, with a focus on Nexus platforms and technologies.
Dedications

This book is dedicated to three important women in my life: my mother, my wife, Khushboo, and Sonal. Mom, thanks for being a friend and a teacher in different phases of my life. You have given me the courage to stand up and fight every challenge that comes my way in life. Khushboo, I want to thank you for being so patient with my madness and craziness. I couldn't have completed this book or any other project without your support, and I cannot express in words how much it all means to me. This book is a small token of love, gratitude and appreciation for you. Sonal, thank you for being the driver behind my craziness. You have inspired me to reach new heights by setting new targets every time we met. This book is a small token of my love and gratitude for all that you have done for me.

I would further like to dedicate this book to my dad and my brother for believing in me and standing behind me as a wall whenever I faced challenges in life. I couldn't be where I am today without your invincible support.

—Vinit Jain

This book is dedicated to David Kyle. Thank you for taking a chance on me. You will always be more than a former boss. You mentored me with the right attitude and foundational skills early in my career.

In addition to stress testing the network with Quake, you let me start my path with networking under you. Look where I am now!

—Brad Edgeworth

This book is dedicated to my loving wife, Sandra, and my daughter, Calianna. You are my inspiration. Your love and support drive me to succeed each and every day. Thank you for providing the motivation for me to push myself further than I thought possible. Calianna, you are only two years old now. When you are old enough to read this, you will have long forgotten about all the late nights daddy spent working on this project. When you hold this book, I want you to remember that anything is possible through dedication and hard work.

I would like to further dedicate this book to my mother and father. Mom, thanks for always encouraging me, and for teaching me that I can do anything I put my mind to. Dad, thank you for always supporting me, and teaching me how to be dedicated and work hard. Both of you have given me your best.

—Richard Furr
Acknowledgments

Vinit Jain:

Brad and Richard: Thank you for being part of this yearlong journey. This project wouldn’t have been possible without your support. It was a great team effort, and it was a pleasure working with both of you.

I would like to thank our technical editors, Ramiro and Matt, for your in-depth verification of the content and insightful input to make this project a successful one.

I couldn’t have completed the milestone without the support from my managers, Chip Little and Mike Stallings. Thank you for enabling us with so many resources, as well as being flexible and making an environment that is full of opportunities.

I would like to thank David Jansen, Lukas Krattiger, Vinayak Sudame, Shridhar Dhodapkar, and Ryan McKenna for your valuable input during the course of this book.

Most importantly, I would like to thank Brett Bartow and Marianne Bartow for their wonderful support on this project. This project wouldn’t have been possible without your support.

Brad Edgeworth:

Vinit, thanks again for asking me to co-write another book with you. Richard, thanks again for your insight. I’ve always enjoyed our late-night conference calls.

Ramiro and Matt, thank you for hiding all my mistakes, or at least pointing them out before they made it to print!

This is the part of the book that you look at to see if you have been recognized. Well, many people have provided feedback, suggestions, and support to make this a great book. Thanks to all who have helped in the process, especially Brett Bartow, Marianne Bartow, Jay Franklin, Katherine McNamara, Dustin Schuemann, Craig Smith, and my managers.

PS. Teagan, this book does not contain dragons or princesses, but the next one might!

Richard Furr:

I’d like to thank my coauthors, Vinit Jain and Brad Edgeworth, for the opportunity to work on this project together. It has been equally challenging and rewarding on many levels.

Brad, thank you for all the guidance and your ruthless red pen on my first chapter. You showed me how to turn words and sentences into a book. Vinit, your drive and ambition are contagious. I look forward to working with both of you again in the future.

I would also like to thank our technical editors, Matt Esau and Ramiro Garza Rios, for their expertise and guidance. This book would not be possible without your contributions.

I could not have completed this project without the support and encouragement of my manager, Mike Stallings. Mike, thank you for allowing me to be creative and pursue projects like this one. You create the environment for us to be our best.
Contents at a Glance

Foreword xxvi
Introduction xxvii

Part I Introduction to Troubleshooting Nexus Switches
Chapter 1 Introduction to Nexus Operating System (NX-OS) 1
Chapter 2 NX-OS Troubleshooting Tools 53
Chapter 3 Troubleshooting Nexus Platform Issues 95

Part II Troubleshooting Layer 2 Forwarding
Chapter 4 Nexus Switching 197
Chapter 5 Port-Channels, Virtual Port-Channels, and FabricPath 255

Part III Troubleshooting Layer 3 Routing
Chapter 6 Troubleshooting IP and IPv6 Services 321
Chapter 7 Troubleshooting Enhanced Interior Gateway Routing Protocol (EIGRP) 393
Chapter 8 Troubleshooting Open Shortest Path First (OSPF) 449
Chapter 9 Troubleshooting Intermediate System-Intermediate System (IS-IS) 507
Chapter 10 Troubleshooting Nexus Route-Maps 569
Chapter 11 Troubleshooting BGP 597

Part IV Troubleshooting High Availability
Chapter 12 High Availability 689

Part V Multicast Network Traffic
Chapter 13 Troubleshooting Multicast 733
Part VI  Troubleshooting Nexus Tunneling

Chapter 14  Troubleshooting Overlay Transport Virtualization (OTV)  875

Part VII  Network Programmability

Chapter 15  Programmability and Automation  949

Index  977

Reader Services

Register your copy at www.ciscopress.com/title/9781587145056 for convenient access to downloads, updates, and corrections as they become available. To start the registration process, go to www.ciscopress.com/register and log in or create an account*. Enter the product ISBN 9781587145056 and click Submit. When the process is complete, you will find any available bonus content under Registered Products.

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

Foreword  xxvi
Introduction  xxvii

### Part I  Introduction to Troubleshooting Nexus Switches

### Chapter 1 Introduction to Nexus Operating System (NX-OS)  1

- Nexus Platforms Overview  2
  - Nexus 2000 Series  2
  - Nexus 3000 Series  3
  - Nexus 5000 Series  4
  - Nexus 6000 Series  4
  - Nexus 7000 Series  5
  - Nexus 9000 Series  6
- NX-OS Architecture  8
  - The Kernel  9
  - System Manager (sysmgr)  9
  - Messages and Transactional Services  11
  - Persistent Storage Services  13
  - Feature Manager  14
  - NX-OS Line Card Microcode  17
  - File Systems  19
    - Flash File System  21
    - Onboard Failure Logging  22
    - Logflash  23
- Understanding NX-OS Software Releases and Packaging  25
  - Software Maintenance Upgrades  27
  - Licensing  28
- NX-OS High-Availability Infrastructure  28
  - Supervisor Redundancy  29
  - ISSU  34
- NX-OS Virtualization Features  35
  - Virtual Device Contexts  35
  - Virtual Routing and Forwarding  37
  - Virtual Port Channel  37
Troubleshooting Cisco Nexus Switches and NX-OS

Management and Operations Capabilities 39
  NX-OS Advanced CLI 39
Technical Support Files 44
Accounting Log 45
Feature Event-History 46
Debug Options: Log File and Filters 47
Configuration Checkpoint and Rollback 48
Consistency Checkers 49
Feature Scheduler, EEM, and Python 50
Bash Shell 51
Summary 51
References 51

Chapter 2  NX-OS Troubleshooting Tools 53
Packet Capture: Network Sniffer 53
  Encapsulated Remote SPAN 57
  SPAN on Latency and Drop 60
  SPAN-on-Latency 60
  SPAN-on-Drop 61
Nexus Platform Tools 63
  Ethanalyzer 63
  Packet Tracer 71
NetFlow 72
  NetFlow Configuration 73
  Enable NetFlow Feature 74
  Define a Flow Record 74
  Define a Flow Exporter 75
  Define and Apply the Flow Monitor 76
NetFlow Sampling 77
sFlow 78
Network Time Protocol 81
Embedded Event Manager 83
Logging 87
  Debug Logfiles 90
  Accounting Log 91
  Event-History 92
Summary 93
References 93
## Chapter 3  Troubleshooting Nexus Platform Issues  95

Troubleshooting Hardware Issues  95
  - Generic Online Diagnostic Tests  98
  - Bootup Diagnostics  98
  - Runtime Diagnostics  100
  - GOLD Test and EEM Support  107
  - Nexus Device Health Checks  108
  - Hardware and Process Crashes  108
  - Packet Loss  110
  - Interface Errors and Drops  110
  - Platform-Specific Drops  116
  - Nexus Fabric Extenders  124

Virtual Device Context  130
  - VDC Resource Template  131
  - Configuring VDC  133
  - VDC Initialization  134
  - Out-of-Band and In-Band Management  137
  - VDC Management  137

Line Card Interop Limitations  141

Troubleshooting NX-OS System Components  142
  - Message and Transaction Services  144
  - Netstack and Packet Manager  148
  - Netstack TCPUDP Component  156
  - ARP and Adjacency Manager  160
  - Unicast Forwarding Components  167
  - Unicast Routing Information Base  167
  - UFDM and IPFIB  171
  - EthPM and Port-Client  175

HWRL, CoPP, and System QoS  179
  - MTU Settings  192
  - FEX Jumbo MTU Settings  193

Troubleshooting MTU Issues  194

Summary  195

References  196
# Troubleshooting Cisco Nexus Switches and NX-OS
## Part II Troubleshooting Layer 2 Forwarding
### Chapter 4 Nexus Switching  197
#### Network Layer 2 Communication Overview  197
- Virtual LANs  200
  - VLAN Creation  201
  - Access Ports  203
  - Trunk Ports  204
  - Native VLANs  206
  - Allowed VLANs  206
  - Private VLANs  207
  - Isolated Private VLANs  208
  - Community Private VLANs  212
  - Using a Promiscuous PVLAN Port on Switched Virtual Interface  215
- Trunking VLANs Between Switches  217
- Spanning Tree Protocol Fundamentals  218
  - IEEE 802.1D Spanning Tree Protocol  219
  - Rapid Spanning Tree Protocol  220
  - Spanning-Tree Path Cost  221
  - Root Bridge Election  222
  - Locating Root Ports  224
  - Locating Blocked Switch Ports  225
  - Verification of VLANs on Trunk Links  227
  - Spanning Tree Protocol Tuning  228
  - Multiple Spanning-Tree Protocol (MST)  236
  - MST Configuration  236
  - MST Verification  237
  - MST Tuning  240
- Detecting and Remediating Forwarding Loops  241
  - MAC Address Notifications  242
  - BPDU Guard  243
  - BPDU Filter  244
  - Problems with Unidirectional Links  245
  - Spanning Tree Protocol Loop Guard  245
  - Unidirectional Link Detection  246
  - Bridge Assurance  250
- Summary  252
- References  254
Chapter 5  Port-Channels, Virtual Port-Channels, and FabricPath  255

Port-Channels  255
  Basic Port-Channel Configuration  259
  Verifying Port-Channel Status  260
  Verifying LACP Packets  262
  Advanced LACP Configuration Options  265
  Minimum Number of Port-Channel Member Interfaces  265
  Maximum Number of Port-Channel Member Interfaces  267
  LACP System Priority  268
  LACP Interface Priority  268
  LACP Fast  269
  Graceful Convergence  270
  Suspend Individual  271
  Port-Channel Member Interface Consistency  271
  Troubleshooting LACP Interface Establishment  272
  Troubleshooting Traffic Load-Balancing  272

Virtual Port-Channel  274
  vPC Fundamentals  275
  vPC Domain  275
  vPC Peer-Keepalive  276
  vPC Peer Link  277
  vPC Member Links  277
  vPC Operational Behavior  277
  vPC Configuration  278
  vPC Verification  280
  Verifying the vPC Domain Status  280
  Verifying the Peer-Keepalive  282
  vPC Consistency-Checker  283
  Advanced vPC Features  288
  vPC Orphan Ports  288
  vPC Autorecovery  289
  vPC Peer-Gateway  289
  vPC ARP Synchronization  291
  Backup Layer 3 Routing  292
  Layer 3 Routing over vPC  293
FabricPath 294
  FabricPath Terminologies and Components 296
  FabricPath Packet Flow 297
  FabricPath Configuration 300
  FabricPath Verification and Troubleshooting 303
FabricPath Devices 310
Emulated Switch and vPC+ 310
  vPC+ Configuration 311
  vPC+ Verification and Troubleshooting 314
Summary 320
References 320

Part III Troubleshooting Layer 3 Routing

Chapter 6 Troubleshooting IP and IPv6 Services 321

IP SLA 321
  ICMP Echo Probe 322
  UDP Echo Probe 324
  UDP Jitter Probe 325
  TCP Connect Probe 328
Object Tracking 329
  Object Tracking for the Interface 330
  Object Tracking for Route State 330
  Object Tracking for Track-List State 332
  Using Track Objects with Static Routes 334
IPv4 Services 335
  DHCP Relay 335
  DHCP Snooping 341
  Dynamic ARP Inspection 345
  ARP ACLs 348
  IP Source Guard 349
  Unicast RPF 351
IPv6 Services 352
  Neighbor Discovery 352
  IPv6 Address Assignment 357
  DHCPv6 Relay Agent 357
  DHCPv6 Relay LDRA 360
  IPv6 First-Hop Security 362
RA Guard 363
IPv6 Snooping 365
DHCPv6 Guard 368
First-Hop Redundancy Protocol 370
HSRP 370
HSRPv6 376
VRRP 380
GLBP 385
Summary 391

Chapter 7 Troubleshooting Enhanced Interior Gateway Routing Protocol (EIGRP) 393
EIGRP Fundamentals 393
Topology Table 395
Path Metric Calculation 396
EIGRP Communication 399
Baseline EIGRP Configuration 399
Troubleshooting EIGRP Neighbor Adjacency 401
Verification of Active Interfaces 402
Passive Interface 403
Verification of EIGRP Packets 405
Connectivity Must Exist Using the Primary Subnet 409
EIGRP ASN Mismatch 412
Mismatch K Values 413
Problems with Hello and Hold Timers 414
EIGRP Authentication Issues 416
Interface-Based EIGRP Authentication 418
Global EIGRP Authentication 418
Troubleshooting Path Selection and Missing Routes 419
Load Balancing 421
Stub 421
Maximum-Hops 424
Distribute List 426
Offset Lists 427
Interface-Based Settings 430
Redistribution 430
Classic Metrics vs. Wide Metrics 433
Problems with Convergence 439
  Active Query 441
  Stuck in Active 443
Summary 446
References 447

Chapter 8 Troubleshooting Open Shortest Path First (OSPF) 449
OSPF Fundamentals 449
  Inter-Router Communication 450
  OSPF Hello Packets 450
  Neighbor States 451
  Designated Routers 452
  Areas 453
  Link State Advertisements 453
Troubleshooting OSPF Neighbor Adjacency 456
  Baseline OSPF Configuration 456
  OSPF Neighbor Verification 458
  Confirmation of OSPF Interfaces 460
  Passive Interface 461
  Verification of OSPF Packets 463
  Connectivity Must Exist Using the Primary Subnet 468
  MTU Requirements 469
  Unique Router-ID 471
  Interface Area Numbers Must Match 471
  OSPF Stub (Area Flags) Settings Must Match 473
  DR Requirements 474
  Timers 476
  Authentication 478
Troubleshooting Missing Routes 482
  Discontiguous Network 482
  Duplicate Router ID 485
  Filtering Routes 487
  Redistribution 487
  OSPF Forwarding Address 488
Troubleshooting OSPF Path Selection 494
  Intra-Area Routes 494
  Inter-Area Routes 495
External Route Selection 495
E1 and N1 External Routes 496
E2 and N2 External Routes 497
Problems with Intermixed RFC 1583 and RFC 2328 Devices 499
Interface Link Costs 500
Summary 504
References 505

Chapter 9 Troubleshooting Intermediate System-Intermediate System (IS-IS) 507
IS-IS Fundamentals 507
Areas 508
NET Addressing 509
Inter-Router Communication 511
IS Protocol Header 511
TLVs 512
IS PDU Addressing 512
IS-IS Hello (IIH) Packets 513
Link-State Packets 515
LSP ID 515
Attribute Fields 515
LSP Packet and TLVs 516
Designated Intermediate System 516
Path Selection 517
Troubleshooting IS-IS Neighbor Adjacency 518
Baseline IS-IS Configuration 518
IS-IS Neighbor Verification 520
Confirmation of IS-IS Interfaces 523
Passive Interface 526
Verification of IS-IS Packets 528
Connectivity Must Exist Using the Primary Subnet 535
MTU Requirements 537
Unique System-ID 539
Area Must Match Between L1 Adjacencies 539
Checking IS-IS Adjacency Capabilities 541
DIS Requirements 543
IIH Authentication 544
Troubleshooting Missing Routes  546
  Duplicate System ID  546
  Interface Link Costs  549
  Mismatch of Metric Modes  553
  L1 to L2 Route Propagations  556
  Suboptimal Routing  562
  Redistribution  566

Summary  567
References  568

Chapter 10  Troubleshooting Nexus Route-Maps  569
  Conditional Matching  569
    Access Control Lists  569
    ACLs and ACL Manager Component  570
    Interior Gateway Protocol (IGP) Network Selection  576
    BGP Network Selection  577
    Prefix Matching and Prefix-Lists  577
    Prefix Matching  578
    Prefix Lists  580

Route-Maps  581
  Conditional Matching  582
    Multiple Conditional Match Conditions  584
    Complex Matching  585
  Optional Actions  586
  Incomplete Configuration of Routing Policies  586
  Diagnosing Route Policy Manger  586

Policy-Based Routing  591
Summary  594
References  595

Chapter 11  Troubleshooting BGP  597
  BGP Fundamentals  597
  Address Families  598
  Path Attributes  599
  Loop Prevention  599

BGP Sessions  600
  BGP Identifier  601
  BGP Messages  601
Contents

OPEN  601
UPDATE  602
NOTIFICATION  602
KEEPALIVE  602
BGP Neighbor States  602
Idle  603
Connect  603
Active  604
OpenSent  604
OpenConfirm  604
Established  605
BGP Configuration and Verification  605
Troubleshooting BGP Peering Issues  609
Troubleshooting BGP Peering Down Issues  609
Verifying Configuration  610
Verifying Reachability and Packet Loss  611
Verifying ACLs and Firewalls in the Path  613
Verifying TCP Sessions  615
OPEN Message Errors  617
BGP Debugs  618
Demystifying BGP Notifications  619
Troubleshooting IPv6 Peers  621
BGP Peer Flapping Issues  622
Bad BGP Update  622
Hold Timer Expired  623
BGP Keepalive Generation  624
MTU Mismatch Issues  626
BGP Route Processing and Route Propagation  630
BGP Route Advertisement  631
Network Statement  631
Redistribution  633
Route Aggregation  634
Default-Information Originate  636
BGP Best Path Calculation  636
BGP Multipath  640
EBGP and IBGP Multipath  640
Part IV  Troubleshooting High Availability

Chapter 12  High Availability  689

Bidirectional Forwarding Detection  689
Asynchronous Mode  691
Asynchronous Mode with Echo Function  693
Configuring and Verifying BFD Sessions  693

Nexus High Availability  707
Stateful Switchover  707
ISSU  713
Graceful Insertion and Removal  719
Custom Maintenance Profile  727

Summary  731
References  732

Part V  Multicast Network Traffic

Chapter 13  Troubleshooting Multicast  733

Multicast Fundamentals  734
Multicast Terminology  735
Layer 2 Multicast Addresses  738
Layer 3 Multicast Addresses  739

NX-OS Multicast Architecture  741
Replication  744
Protecting the Central Processing Unit  745
NX-OS Multicast Implementation  747
Static Joins  748
Clearing an MROUTE Entry  748
Multicast Boundary and Filtering  748
Event-Histories and Show Techs  749

IGMP  750
IGMPv2  751
IGMPv3  752
IGMP Snooping  756
IGMP Verification  761

PIM Multicast  771
PIM Protocol State and Trees  772
PIM Message Types  773
Troubleshooting Cisco Nexus Switches and NX-OS

PIM Hello Message 775
PIM Register Message 775
PIM Register-Stop Message 776
PIM Join-Prune Message 776
PIM Bootstrap Message 777
PIM Assert Message 778
PIM Candidate RP Advertisement Message 779
PIM DF Election Message 779
PIM Interface and Neighbor Verification 780
PIM Any Source Multicast 785
PIM ASM Configuration 787
PIM ASM Verification 788
PIM ASM Event-History and MROUTE State Verification 789
PIM ASM Platform Verification 795
PIM Bidirectional 799
BiDIR Configuration 803
BiDIR Verification 805
PIM RP Configuration 811
Static RP Configuration 812
Auto-RP Configuration and Verification 813
BSR Configuration and Verification 820
Anycast-RP Configuration and Verification 830
Anycast RP with MSDP 831
PIM Anycast RP 838
PIM Source Specific Multicast 841
SSM Configuration 843
SSM Verification 845
Multicast and Virtual Port-Channel 848
vPC-Connected Source 849
vPC-Connected Receiver 861
vPC Considerations for Multicast Traffic 870
Duplicate Multicast Packets 870
Reserved VLAN 870
Ethalyzer Examples 871
Summary 871
References 872
Part VI  Troubleshooting Nexus Tunneling

Chapter 14  Troubleshooting Overlay Transport Virtualization (OTV)  875

OTV Fundamentals  875
   Flood Control and Broadcast Optimization  877
   Supported OTV Platforms  878
   OTV Terminology  878
   Deploying OTV  881
   OTV Deployment Models  881
   OTV Site VLAN  882
   OTV Configuration  882
Understanding and Verifying the OTV Control Plane  885
   OTV Multicast Mode  887
   OTV IS-IS Adjacency Verification  888
   OTV IS-IS Topology Table  898
   OTV IS-IS Authentication  905
   Adjacency Server Mode  907
   OTV Control Plane Policing (CoPP)  912
Understanding and Verifying the OTV Data Plane  913
   OTV ARP Resolution and ARP-ND-Cache  915
   Broadcasts  917
   Unknown Unicast Frames  918
   OTV Unicast Traffic with a Multicast Enabled Transport  919
   OTV Multicast Traffic with a Multicast Enabled Transport  924
   OTV Multicast Traffic with a Unicast Transport
      (Adjacency Server Mode)  932
Advanced OTV Features  937
   First Hop Routing Protocol Localization  938
   Multihoming  939
   Ingress Routing Optimization  940
   VLAN Translation  941
   OTV Tunnel Depolarization  942
   OTV Fast Failure Detection  944
Summary  946
References  947
Part VII  Network Programmability

Chapter 15  Programmability and Automation  949

Introduction to Automation and Programmability  949
Introduction to Open NX-OS  950
Shells and Scripting  951
  Bash Shell  951
  Guest Shell  957
  Python  960

NX-SDK  964
NX-API  968
Summary  975
References  975

Index  977
Icons Used in This Book

Command Syntax Conventions

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

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

**Note** This book covers multiple Nexus switch platforms (5000, 7000, 9000, etc). A generic NX-OS icon is used along with a naming syntax for differentiation of devices. Platform-specific topics use a platform-specific icon and major platform number in the system name.
Foreword

The data center is at the core of all companies in the digital age. It processes bits and bytes of data that represent products and services to its customers. The data storage and processing capabilities of a modern business have become synonymous with the ability to generate revenue. Companies in all business sectors are storing and processing more information digitally every year, regardless of their vertical affiliation (construction, medical, entertainment, and so on). This means that the network must be designed for speed, capacity, and flexibility.

The Nexus platform was built with speed and bandwidth capacity in mind. When the Nexus 7000 launched in 2008, it provided high-density 10 Gigabit interfaces at a low per-port cost. In addition, the Nexus switch operating system, NX-OS, brought forth evolutionary technologies like virtual port channels (vPC) that increased available bandwidth and redundancy while overcoming the inefficiencies of Spanning-Tree Protocol (STP). NX-OS introduced technologies such as Overlay Transport Virtualization (OTV), which revolutionized the design of the data center network by enabling host mobility between sites and allowing full data center redundancy. Today, the Nexus platform continues to evolve by supporting 25/40/100 Gigabit interfaces in a high-density compact form factor, and brings other innovative technologies such as VXLAN and Application Centric Infrastructure (ACI) to the market.

NX-OS was built with the mindset of operational simplicity and includes additional tools and capabilities that improve the operational efficiency of the network. Today, websites and applications are expected to be available 24 hours a day, 7 days a week, and 365 days a year. Downtime in the data center directly translates to a financial impact. The move toward digitization and the potential impact the network has to a business makes it more important than ever for network engineers to attain the skills to troubleshoot data center network environments efficiently.

As the leader of Cisco’s technical services for more than 25 years, I have the benefit of working with the best network professionals in the industry. This book is written by Brad, Richard, and Vinit: “Network Rock Stars,” who have been in my organization for years supporting multiple Cisco customers. This book provides a complete reference for troubleshooting Nexus switches and the NX-OS operating system. The methodologies taught in this book are the same methods used by Cisco’s technical services to solve a variety of complex network problems.

Joseph Pinto
SVP, Technical Services, Cisco, San Jose
Introduction

The Nexus operating system (NX-OS) contains a modular software architecture that primarily targets high-speed/high-density network environments like data centers. NX-OS provides virtualization, high availability, scalability, and upgradeability features for Nexus switches.

In particular, the NX-OS is expected to have a measure of resilience during software upgrades or hardware upgrades (failover, OIR), with both sets of operations not affecting nonstop forwarding. NX-OS is required to scale to very large multichassis systems and still operate with the same expectations of resilience in the face of outages of various kinds. The NX-OS feature set includes a variety of features and protocols that have revolutionized data center designs with virtual port channels (vPC), Overlay Transport Virtualization (OTV), and now virtual extensible LAN (VXLAN).

The Nexus 7000 switch debuted in 2008, providing more than 512 10 Gbps ports. Over the years, Cisco has released other Nexus switch families that include the Nexus 5000, Nexus 2000, Nexus 9000, and virtual Nexus 1000. NX-OS has grown in features, allowing Nexus switch deployments in enterprise routing and switching roles.

This book is the single source for mastering techniques to troubleshoot various features and issues running on Nexus platforms with NX-OS operating system. Bringing together content previously spread across multiple sources and Cisco Press titles, it covers updated various features and architecture-level information on how various features function on Nexus platforms and how one can leverage the capabilities of NX-OS to troubleshoot them.

Who Should Read This Book?

Network engineers, architects, or consultants who want to learn more about the underlying Nexus platform and NX-OS operating system so that they can know how to troubleshoot complex network issues with NX-OS. This book also provides a great reference for those studying for their CCIE Data Center Certification.

How This Book Is Organized

Although this book could be read cover to cover, it is designed to be flexible and allow you to easily move between chapters and sections of chapters to cover just the material that you need more work with.

Part I of the book, “Introduction to Troubleshooting Nexus Switches” provides an overview on the Nexus platform and the components of NX-OS used for troubleshooting network events.
Chapter 1, “Introduction to the Nexus Operating System (NX-OS)”: This chapter introduces the Nexus platform and the major functional components of the Nexus operating system (NX-OS). The chapter discusses the four fundamental pillars of NX-OS: resiliency, virtualization, efficiency, and extensibility.

Chapter 2, “NX-OS Troubleshooting Tools”: This chapter explains the history of packet capture, NetFlow, EEM, logging, and event history.

Chapter 3, “Troubleshooting Nexus Platform Issues”: This chapter examines various Nexus platform components and commands to troubleshoot issues with the supervisor cards and line cards, hardware drops, and fabric issues. This chapter also examines how to troubleshoot interface and PLIM-level issues on the line card. This chapter also covers issues related to CoPP policies and how to troubleshoot CoPP-related issues.

Part II of the book, “Troubleshooting Layer 2 Forwarding,” explains the specific components for troubleshooting Nexus switches during the switching of network packets.

Chapter 4, “Nexus Switching”: This chapter explains how Nexus switches forward packets and explains switch port types, private VLANs, and Spanning-Tree Protocol (STP).

Chapter 5, “Port Channels, Virtual Port-Channels, and FabricPath”: This chapter covers in great detail how vPC, Fabric Path, and vPC+ works and how they add value to the next generation DC design. This chapter focuses on designing, implementing, and troubleshooting issues related to vPC and vPC+.

Part III of the book, “Troubleshooting Layer 3 Routing,” explains the underlying IP components of NX-OS. This includes the routing protocols EIGRP, OSPF, IS-IS, BGP, and the selection of routes for filtering or path manipulation.

Chapter 6, “Troubleshooting IP and IPv6 Services”: This chapter explains how various IPv4 and IPv6 services work and how to troubleshoot the same on Nexus platforms. This chapter also covers FHRP protocols, such as HSRP, VRRP, and Anycast HSRP.

Chapter 7, “Troubleshooting Enhanced Interior Gateway Routing Protocol (EIGRP)”: This chapter explains how to troubleshoot various issues related to EIGRP, including forming EIGRP neighborships, suboptimal routing, and other common EIGRP problems.

Chapter 8, “Troubleshooting Open Shortest Path First (OSPF)”: This chapter explains how to troubleshoot various issues related to OSPF, including forming OSPF neighbor adjacencies, suboptimal routing, and other common OSPF problems.
Chapter 9, “Troubleshooting Intermediate System–Intermediate System (IS-IS)”: This chapter explains how to troubleshoot various issues related to IS-IS, including forming IS-IS neighbor adjacencies, suboptimal routing, and other common IS-IS problems.

Chapter 10, “Troubleshooting Nexus Route-Maps”: This chapter discusses various network selection techniques for filtering or metric manipulation. It explains conditional matching of routes using access control lists (ACL), prefix-lists, and route-maps.

Chapter 11, “Troubleshooting BGP”: This chapter explains how to troubleshoot various issues related to BGP, including BGP neighbor adjacencies, path selection, and other common issues.

Part IV of the book, “Troubleshooting High Availability,” discusses and explains the high availability components of NX-OS.

Chapter 12, “High Availability”: This chapter explains how to troubleshoot high availability components such as bidirectional forward detection (BFD), Stateful Switchover (SSO), In-service software upgrade (ISSU) and Graceful Insertion and Removal (GIR).


Chapter 13, “Troubleshooting Multicast”: This chapter explains the various components of multicast and how multicast network issues can be identified and resolved.

Part VI of the book, “Troubleshooting Nexus Tunneling,” discusses the various tunneling techniques that NX-OS provides.

Chapter 14, “Troubleshooting Overlay Transport Virtualization (OTV)”: This chapter explains the revolutionary overlay transport virtualization technology and how it operates, along with the process for troubleshooting issues with it.

Part VII of the book, “Network Programmability,” provides details on the methods that NX-OS can be configured with APIs and automation.

Chapter 15, “Programmability and Automation”: This chapter examines various application programming interfaces (APIs) that are available with NX-OS and how they enable network operations to automate their network.

On the product web page you also will find a bonus chapter, “Troubleshooting VxLAN and VxLAN BGP EVPN.”
Additional Reading

The authors tried to keep the size of the book manageable while providing only necessary information for the topics involved.

Some readers may require additional reference material and may find the following books a great supplementary resource for the topics in this book.

This chapter covers the following topics:

- OTV Fundamentals
- Understanding and Troubleshooting the OTV Control Plane
- Understanding and Troubleshooting the OTV Data Plane
- Advanced OTV Features

Overlay Transport Virtualization (OTV) is a MAC-in-IP overlay encapsulation that allows Layer 2 (L2) communication between sites that are separated by a Layer 3 (L3) routed network. OTV revolutionized network connectivity by extending L2 applications across multiple data centers without changing the existing network design. This chapter focuses on providing an overview of OTV, the processes for the OTV control and data plane and how to troubleshoot OTV.

**OTV Fundamentals**

The desire to connect data center sites at L2 is driven by the need for Virtual Machine (VM) and workload mobility, or for creating geographically diverse redundancy. Critical networks may even choose to have a fully mirrored disaster recovery site that synchronizes data and services between sites. Having the capability to put services from multiple locations into the same VLAN allows mobility between data centers without reconfiguring the network layer addressing of the host or server when it is moved.
The challenges and considerations associated with connecting two or more data centers at L2 are the following:

- Transport network types available
- Multihoming sites for redundancy
- Allowing each site to be independent from the others
- Creating fault isolation boundaries
- Ensuring the network can be expanded to future locations without disruption to existing sites

Before OTV, L2 data center interconnect (DCI) was achieved with the use of direct fiber links configured as L2 trunks, IEEE 802.1Q Tunneling (Q-in-Q), Ethernet over MPLS (EoMPLS), or Virtual Private LAN Service (VPLS). These options rely on potentially complex configuration by a transport service provider to become operational. Adding a site with those solutions means the service provider needs to be involved to complete the necessary provisioning.

OTV, however, can provide an L2 overlay network between sites using only an L3 routed underlay. Because OTV is encapsulated inside an IP packet for transport, it can take advantage of the strengths of L3 routing; for example, IP Equal Cost Multipath (ECMP) routing for load sharing and redundancy as well as optimal packet paths between OTV edge devices (ED) based on routing protocol metrics. Troubleshooting is simplified as well because traffic in the transport network is traditional IP with established and familiar troubleshooting techniques.

Solutions for L2 DCI such as Q-in-Q, EoMPLS, and VPLS all require the service provider to perform some form of encapsulation and decapsulation on the traffic for a site. With OTV, the overlay encapsulation boundary is moved from the service provider to the OTV site, which provides greater visibility and control for the network operator. The overlay configuration can be modified at will and does not require any interaction with or dependence on the underlay service provider. Modifications to the overlay include actions like adding new OTV sites or changing which VLANs are extended across the OTV overlay.

The previously mentioned transport protocols rely on static or stateful tunneling. With OTV, encapsulation of the overlay traffic happens dynamically based on MAC address to IP next-hop information supplied by OTV's Intermediate System to Intermediate System (IS-IS) control plane. This concept is referred to as MAC address routing, and it is explored in detail throughout this chapter. The important point to understand is that OTV maps a MAC address to a remote IP next-hop dynamically using a control plane protocol.

Multihoming is desirable for redundancy purposes, but could be inefficient if those redundant links and devices never get used. With traditional L2 switching, multihoming
had to be planned and configured carefully to avoid L2 loops and Spanning-Tree Protocol (STP) blocking ports. OTV has considerations for multihoming built in to the protocol. For example, multiple OTV edge devices can be deployed in a single site, and each can actively forward traffic for different VLANs. Between data centers, multiple L3 routed links exist and provide L3 ECMP redundancy and load sharing between the OTV edge devices in each data center site.

Having redundant data centers is useful only if they exist in different fault domains, and problems from one data center do not affect the other. This implies that each data center must be isolated in terms of STP, and traffic forwarding loops between sites must be avoided. OTV allows each data center site to contain an independent STP Root Bridge for the VLANs extended across OTV. This is possible because OTV does not forward STP Bridge Protocol Data Units (BPDU) across the overlay, allowing each site to function independently.

**Flood Control and Broadcast Optimization**

Traditional L2 switches learn MAC addresses when frames arrive on a port. The source MAC address and associated interface mapping are kept until the MAC address is aged out or learned on a new interface. If the destination MAC address is not yet known, a switch performs unicast flooding. When this occurs, the unknown unicast traffic is flooded on all ports of the VLAN in an effort to reach the correct destination. In contrast, OTV learns MAC addresses from the remote data center through the IS-IS control plane protocol and will not flood any unknown unicast traffic across the overlay. Address Resolution Protocol (ARP) traffic is another source of flooded traffic in traditional switched networks. With OTV enabled, ARP is flooded in a controlled manner, and ARP responses are snooped and stored in a local ARP Neighbor Discovery (ND) cache by the OTV edge device. Any subsequent ARP requests for the host are answered by the OTV edge device on behalf of the host, which reduces the amount of broadcast traffic crossing the overlay.

Broadcast and multicast traffic in a VLAN must reach all remote data center locations. OTV relies on IP multicast in the underlay transport network to deliver this type of traffic in an efficient and scalable manner. By utilizing IP multicast transport, OTV eliminates the need for an edge device to perform head-end replication for each remote edge device. Head-end replication means that the originating OTV edge device creates a copy of the frame for each remote edge device. This can become a burden if there are many OTV sites and the packet rate is high. By using IP multicast transport, the OTV edge device needs to create only a single packet. Replication happens automatically by the multicast-enabled routers in the underlay transport network as the packets traverse the multicast tree to the receivers (Remote OTV edge devices).
Supported OTV Platforms

OTV is supported on the Nexus 7000 series and requires the Transport Service license (TRS) to be installed. Most deployments take advantage of Virtual Device Contexts (VDC) to logically separate the routing and OTV responsibilities in a single chassis.

Note OTV is also supported on Cisco ASR1000 series routers. The protocol functionality is similar but there may be implementation differences. This chapter focuses only on OTV on the Nexus 7000 series switches.

VLANs are aggregated into a distribution switch and then fed into a dedicated OTV VDC through a L2 trunk. Any traffic in a VLAN that needs to reach the remote data center is switched to the OTV VDC where it gets encapsulated by the edge device. The packet then traverses the routed VDC as an L3 IP packet and gets routed toward the remote OTV edge device for decapsulation. Traffic that requires L3 routing is fed from the L2 distribution to a routing VDC. The routing VDC typically has a First Hop Redundancy Protocol (FHRP) like Hot Standby Router Protocol (HSRP) or Virtual Router Redundancy Protocol (VRRP) to provide a default-gateway address to the hosts in the attached VLANs and to perform Inter VLAN routing.

Note Configuring multiple VDCs may require the installation of additional licenses, depending on the requirements of the deployment and the number of VDCs.

OTV Terminology

An OTV network topology example is shown in Figure 14-1. There are two data center sites connected by an L3 routed network that is enabled for IP multicast. The L3 routed network must provide IP connectivity between the OTV edge devices for OTV to function correctly. The placement of the ED is flexible as long as the OTV ED receives L2 frames for the VLANs that require extension across OTV. Usually the OTV ED is connected at the L2 and L3 boundary.

Data center 1 contains redundant OTV VDCs NX-2 and NX-4, which are the edge devices. NX-1 and NX-3 perform the routing and L2 VLAN aggregation and connect the access switch to the OTV VDC internal interface. The OTV join interface is a Layer 3 interface connected to the routing VDC. Data center 2 is configured as a mirror of Data center 1; however, the port-channel 3 interface is used as the OTV internal interface instead of the OTV join interface as in Data center 1. VLANs 100–110 are being extended with OTV between the data centers across the overlay.
The OTV terminology introduced in Figure 14-1 is explained in Table 14-1.

### Table 14-1  OTV Terminology

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edge Device (ED)</td>
<td>Responsible for dynamically encapsulating Ethernet frames into L3 IP packets for VLANs that are extended with OTV.</td>
</tr>
<tr>
<td>Authoritative Edge Device (AED)</td>
<td>Forwards traffic for an extended VLAN across OTV. Advertises MAC-address reachability for the VLANs it is active for to remote sites through the OTV IS-IS control plane. The Authoritative Edge Device (AED) is determined based on an ordinal value of 0 (zero) or 1 (one). The edge device with ordinal zero is AED for all even VLANs, and the edge device with ordinal one is AED for all odd VLANs. This ordinal is determined when the adjacency is formed between two edge devices at a site and is not configurable.</td>
</tr>
<tr>
<td>Internal Interface</td>
<td>Interface on the OTV edge device that connects to the local site. This interface provides a traditional L2 interface from the ED to the internal network, and MAC addresses are learned as traffic is received. The internal interface is an L2 trunk that carries the VLANs being extended by OTV.</td>
</tr>
<tr>
<td>Term</td>
<td>Definition</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Join Interface</td>
<td>Interface on the OTV edge device that connects to the L3 routed network and used to source OTV encapsulated traffic. It can be a Loopback, L3 point-to-point interface, or L3 Port-channel interface. Subinterfaces may also be used. Multiple overlays can use the same join interface.</td>
</tr>
<tr>
<td>Overlay Interface</td>
<td>Interface on the OTV ED. The overlay interface is used to dynamically encapsulate the L2 traffic for an extended VLAN in an IP packet for transport to a remote OTV site. Multiple overlay interfaces are supported on an edge device.</td>
</tr>
<tr>
<td>Site VLAN</td>
<td>A VLAN that exists in the local site that connects the OTV edge devices at L2. The site VLAN is used to discover other edge devices in the local site and allows them to form an adjacency. After the adjacency is formed, the Authoritative Edge Device (AED) for each VLAN is elected. The site VLAN should be dedicated for OTV and not extended across the overlay. The site VLAN should be the same VLAN number at all OTV sites.</td>
</tr>
<tr>
<td>Site Identifier</td>
<td>The site-id must be the same for all edge devices that are part of the same site. Value ranges from 0x1 to 0xffffffff. The site-id is advertised in IS-IS packets, and it allows edge devices to identify which edge devices belong to the same site. Edge devices form an adjacency on the overlay as well as on the site VLAN (Dual adjacency). This allows the adjacency between edge devices in a site to be maintained even if the site VLAN adjacency gets broken due to a connectivity problem. The overlay interface will not come up until a site identifier is configured.</td>
</tr>
<tr>
<td>Site Adjacency</td>
<td>Formed across the site VLAN between OTV edge devices that are part of the same site. If an IS-IS Hello is received from an OTV ED on the site VLAN with a different site-id than the local router, the overlay is disabled. This is done to prevent a loop between the OTV internal interface and the overlay. This behavior is why it is recommended to make the OTV internal VLAN the same at each site.</td>
</tr>
<tr>
<td>Overlay Adjacency</td>
<td>OTV adjacency established on the OTV join interface. Adjacencies on the overlay interface are formed between sites, as well as for edge devices that are part of the same site. Edge devices form dual adjacency (site and overlay) for resiliency purposes. For devices in the same site to form an overlay adjacency, the site-id must match.</td>
</tr>
</tbody>
</table>
Deploying OTV

The configuration of the OTV edge device consists of the OTV internal interface, the join interface, and the overlay virtual interface. Before attempting to configure OTV, the capabilities of the transport network must be understood, and it must be correctly configured to support the OTV deployment model.

OTV Deployment Models

There are two OTV deployment models available, depending on the capabilities of the transport network.

- **Multicast Enabled Transport:** The control plane is encapsulated in IP multicast packets. Allows for dynamic neighbor discovery by having each OTV ED join the multicast control-group through the transport. A single multicast packet is sent by the OTV ED, which gets replicated along the multicast tree in the transport to each remote OTV ED.

- **Adjacency Server Mode:** Neighbors must be manually configured for the overlay interface. Unicast control plane packets are created for each individual neighbor and routed through the transport.

The OTV deployment model that is deployed should be decided during the planning phase after verifying the capabilities of the transport network. If multicast is supported in the transport, it is recommended to use the multicast deployment model. If there is no multicast support available in the transport network, use the adjacency server model.

The transport network must provide IP routed connectivity for unicast and multicast communication between the OTV EDs. The unicast connectivity requirements are achieved with any L3 routing protocol. If the OTV ED does not form a dynamic routing adjacency with the data center, it must be configured with static routes to reach the join interfaces of the other OTV EDs.

Multicast routing in the transport must be configured to support Protocol Independent Multicast (PIM). An Any Source Multicast (ASM) group is used for the OTV control-group, and a range of PIM Source Specific Multicast (SSM) groups are used for OTV data-groups. IGMPv3 should be enabled on the join interface of the OTV ED.

**Note**  It is recommended to deploy PIM Rendezvous Point (RP) redundancy in the transport network for resiliency.
OTV Site VLAN

Each OTV site should be configured with an OTV site VLAN. The site VLAN should be trunked from the data center L2 switched network to the OTV internal interface of each OTV ED. Although not required, it is recommended to use the same VLAN at each OTV site in case the site VLAN is accidentally leaked between OTV sites.

With the deployment model determined and the OTV VDC created with the TRANSPORT_SERVICES_PKG license installed, the following steps are used to enable OTV functionality. The following examples are based upon a multicast enabled transport.

OTV Configuration

Before any OTV configuration is entered, the feature must be enabled with the feature otv command. Example 14-1 shows the configuration associated with the OTV internal interface, which is the L2 trunk port that participates in traditional switching with the existing data center network. The VLANs to be extended over OTV are VLAN 100–110. The site VLAN for both data centers is VLAN 10, which is being trunked over the OTV internal interface, along with VLANs 100–110.

Example 14-1  OTV Internal Interface Configuration

```
NX-2# show run | no-more
| Output omitted for brevity
feature otv
vlan 1,10,100-110
interface Ethernet3/5
  description To NX-1 3/19, OTV internal interface
  switchport
  switchport mode trunk
  mtu 9216
  no shutdown
```

The OTV internal interface should be considered as an access switch in the design of the data center’s STP domain.

After the OTV internal interface is configured, the OTV join interface can be configured. The OTV join interface can be configured on M1, M2, M3, or F3 modules and can be a Loopback interface or an L3 point-to-point link. It is also possible to use an L3 port-channel, or a subinterface, depending on the deployment requirements. Example 14-2 shows the relevant configuration for the OTV join interface.
Example 14-2  OTV Join Interface Configuration

```
xr# show run | no-more
! Output omitted for brevity
feature otv

interface port-channel3
description To NX-1 Po3, OTV Join interface
mtu 9216
ip address 10.1.12.1/24
ip router ospf 1 area 0.0.0.0
ip igmp version 3

interface Ethernet3/7
description To NX-1 Eth3/22, OTV Join interface
mtu 9216
channel-group 3 mode active
no shutdown

interface Ethernet3/8
description To NX-1 Eth3/23, OTV Join interface
mtu 9216
channel-group 3 mode active
no shutdown
```

The OTV join interface is an Layer 3 point-to-point interface and is configured for IGMP version 3. IGMPv3 is required so the OTV ED can join the control-group and data-groups required for OTV functionality.

Open Shortest Path First (OSPF) is the routing protocol in this topology and is used in both data centers. The OTV ED learns the unicast routes to reach all other OTV EDs through OSPF. The entire data center was configured with MTU 9216 on all infrastructure links to allow full 1500 byte frames to pass between applications without the need for fragmentation.

Beginning in NX-OS Release 8.0(1), a loopback interface can be used as the OTV join interface. If this option is used, the configuration will differ from this example, which utilizes an L3 point-to-point interface. At least one L3 routed interface must connect the OTV ED to the data center network. A PIM neighbor needs to be established over this L3 interface, and the OTV ED needs to be configured with the correct PIM Rendezvous Point (RP) and SSM-range that matches the routed data center devices and the transport network. Finally, the loopback interface used as the join interface must be configured with ip pim sparse-mode so that it can act as both a source and receiver for the OTV control-group and data-groups. The loopback also needs to be included in the dynamic routing protocol used for Layer 3 connectivity in the data center so that reachability exists to other OTV EDs.
Note  OTV encapsulation increases the size of L2 frames as they are transported across the IP transport network. The considerations for OTV MTU are further discussed later in this chapter.

With the OTV internal interface and join interface configured; the logical interface referred to as the overlay interface can now be configured and bound to the join interface. The overlay interface is used to dynamically encapsulate VLAN traffic between OTV sites. The number assigned to the overlay interface must be the same on all OTV EDs participating in the overlay. It is possible for multiple overlay interfaces to exist on the same OTV ED, but the VLANs extended on each overlay must not overlap.

The OTV site VLAN is used to form a site adjacency with any other OTV EDs located in the same site. Even for a single OTV ED site, the site VLAN must be configured for the overlay interface to come up. Although not required, it is recommended that the same site VLAN be configured at each OTV site. This is to allow OTV to detect if OTV sites become merged, either on purpose or in error. The site VLAN should not be included in the OTV extended VLAN list. The site identifier should be configured to the same value for all OTV EDs that belong to the same site. The `otv join-interface [interface]` command is used to bind the overlay interface to the join interface. The join interface is used to send and receive the OTV multicast control plane messaging used to form adjacencies and learn MAC addresses from other OTV EDs.

Because this configuration is utilizing a multicast capable transport network, the `otv control-group [group number]` is used to declare which IP PIM ASM group will be used for the OTV control plane group. The control plane group will carry OTV control plane traffic such as IS-IS hellos across the transport and allow the OTV EDs to communicate. The group number should match on all OTV EDs and must be multicast routed in the transport network. Each OTV ED acts as both a source and receiver for this multicast group.

The `otv data-group [group number]` is used to configure which Source Specific Multicast (SSM) groups are used to carry multicast data traffic across the overlay. This group is used to transport multicast traffic within a VLAN across the OTV overlay between sites. The number of multicast groups included in the data-group is a balance between optimization and scalability. If a single group is used, all OTV EDs receive all multicast traffic on the overlay, even if there is no receiver at the site. If a large number of groups is defined, multicast traffic can be forwarded optimally, but the number of groups present in the transport network could become a scalability concern. Presently, 256 multicast data groups are supported for OTV.

After the configuration is completed, the Overlay0 interface must be `no shutdown`. OTV adjacencies will then form between the OTV EDs, provided the underlay network
has been properly configured for both unicast and multicast routing. Example 14-3 contains the configuration for `interface Overlay0` on NX-2 as well as the `site-VLAN` and `site-identifier` configurations.

Example 14-3  OTV Overlay Interface Configuration

```bash
NX-2# show running-config | no-more

! Output omitted for brevity

feature otv

otv site-vlan 10

interface Overlay0
description Site A
otv join-interface port-channel3
otv control-group 239.12.12.12
otv data-group 232.1.1.0/24
otv extend-vlan 100-110
no shutdown

otv site-identifier 0x1
```

Note  If multihoming is planned for the deployment, it is recommended to first enable a single OTV ED at each site. After the OTV functionality has been verified, the second OTV ED can be enabled. This phased approach is recommended to allow for simplified troubleshooting if a problem occurs.

Understanding and Verifying the OTV Control Plane

Instead of relying on packet flooding and data plane MAC learning, which is implemented by traditional L2 switches, OTV takes advantage of an IS-IS control plane to exchange MAC address reachability information between sites. The benefit of this approach is that flooding of packets for an unknown unicast address can be eliminated with the assumption that there are no silent hosts.

OTV uses the existing functionality of IS-IS as much as possible. This includes the formation of neighbors and the use of LSPs and PDUs to exchange reachability information. OTV EDs discover each other with IS-IS hello packets and form adjacencies on the site VLAN as well as on the overlay, as shown in Figure 14-2.
IS-IS uses a Type-Length-Value (TLV) method to encode messages between neighbors, which allows flexibility and extendibility. Through various functionality enhancements over time, IS-IS has been extended to carry reachability information for multiple protocols by defining new corresponding TLVs. OTV uses IS-IS TLV type 147 called the MAC-Reachability TLV to carry MAC address reachability. This TLV contains a Topology-ID, a VLAN-ID, and a MAC address, which allows an OTV ED to learn MAC addresses from other OTV EDs and form the MAC routing table.

OTV is an overlay protocol, which means its operation is dependent upon the underlying transport protocols and the reachability they provide. As the control plane is examined in this chapter, it will become apparent that to troubleshoot OTV, the network operator must be able to segment the different protocol layers and understand the interaction between them. The OTV control plane consists of L2 switching, L3 routing, IP multicast, and IS-IS. If troubleshooting is being performed in the transport network, the OTV control plane packets must now be thought of as data plane packets, where the source and destination hosts are actually the OTV EDs. The transport network has control plane protocols that may also need investigation to solve an OTV problem.
OTV Multicast Mode

IS-IS packets on the overlay interface are encapsulated with the OTV IP multicast header and sent from each OTV ED to the transport network. For clarity, this process is depicted for a single OTV ED, NX-2 as shown in Figure 14-3. In actuality, each OTV ED is both a source and a receiver for the OTV control-group on the OTV join interface. The transport network performs multicast routing on these packets, which use a source address of the OTV ED’s join interface, and a group address of the OTV control-group. Replication of the traffic across the transport happens as needed along the multicast tree so that each OTV ED that has joined the OTV control-group receives a copy of the packet. When the packet arrives at the remote OTV ED, the outer IP Multicast header encapsulation is removed, and the IS-IS packet is delivered to OTV for processing.

The transport network’s multicast capability allows OTV to form IS-IS adjacencies as if each OTV ED were connected to a common LAN segment. In other words, think of the control-group as a logical multipoint connection from one OTV ED to all other OTV EDs. The site adjacency is formed over the site VLAN, which connects both OTV EDs in a site across the internal interface using direct L2 communication.

![Figure 14-3 OTV Control Plane with Multicast Transport](image-url)
The behavior of forming Dual Adjacencies on the site VLAN and the overlay began with NX-OS release 5.2(1). Prior to this, OTV EDs in a site only formed site adjacencies.

The IS-IS protocol used by OTV does not require any user configuration for basic functionality. When OTV is configured the IS-IS process gets enabled and configured automatically. Adjacencies form provided that the underlying transport is functional and the configured parameters for the overlay are compatible between OTV EDs.

The IS-IS control plane is fundamental to the operation of OTV. It provides the mechanism to discover both local and remote OTV EDs, form adjacencies, and exchange MAC address reachability between sites. MAC address advertisements are learned through the IS-IS control plane. An SPF calculation is performed, and then the OTV MAC routing table is populated based on the result. When investigating a MAC address reachability issue, the advertisement is tracked through the OTV control plane to ensure that the ED has the correct information from all IS-IS neighbors. If a host-to-host reachability problem exists across the overlay, it is recommended to begin the investigation with a validation of the control plane configuration and operational state before moving into the data plane.

**OTV IS-IS Adjacency Verification**

Verification of the overlay interface is the first step to investigating any OTV adjacency problem. As shown in example 14-4, the `show otv overlay [overlay-identifier]` command provides key information that is required to begin investigating an OTV problem.

**Example 14-4 Status of the Overlay**

```
NX-2# show otv overlay 0

  show otv overlay 0

  OTV Overlay Information
  Site Identifier 0000.0000.0001
  Encapsulation-Format ip - gre

  Overlay interface Overlay0

  VPN name : Overlay0
  VPN state : UP
  Extended vlans : 100-110 (Total:11)
  Control group : 239.12.12.12
  Data group range(s) : 232.1.1.0/24
  Broadcast group : 239.12.12.12
  Join interface(s) : Po3 (10.1.12.1)
  Site vlan : 10 [up]
  AED-Capable : Yes
  Capability : Multicast-Reachable
```
The output of Example 14-4 verifies the Overlay0 interface is operational, which VLANs are being extended, the transport multicast groups for the OTV control-group and data-groups, the join interface, site VLAN, and AED capability. This information should match what has been configured in the overlay interface on the local and remote site OTV EDs.

Example 14-5 demonstrates how to verify that the IS-IS adjacencies are properly formed for OTV on the overlay interface.

Example 14-5  OTV IS-IS Adjacencies on the Overlay

```
NX-2# show otv adjacency
Overlay Adjacency database

Overlay-Interface Overlay0 :

Hostname    System-ID  Dest Addr    Up Time  State
NX-4        64a0.e73e.12c2 10.1.22.1    03:51:57 UP
NX-8        64a0.e73e.12c4 10.2.43.1    03:05:24 UP
NX-6        6c9c.ed4d.d944 10.2.34.1    03:05:29 UP
```

The output of the `show otv site` command, as shown in Example 14-6, is used to verify the site adjacency. The adjacency with NX-4 is in the Full state, which indicates that both the overlay and site adjacencies are functional (Dual Adjacency).

Example 14-6  OTV IS-IS Site Adjacency

```
NX-2# show otv site

Dual Adjacency State Description
  Full   - Both site and overlay adjacency up
  Partial - Either site/overlay adjacency down
  Down   - Both adjacencies are down (Neighbor is down/unreachable)
  (!)    - Site-ID mismatch detected

Local Edge Device Information:
  Hostname NX-2
  System-ID 6c9c.ed4d.d942
  Site-Identifier 0000.0000.0001
  Site-VLAN 10 State is Up

Site Information for Overlay0:
  Local device is AED-Capable
  Neighbor Edge Devices in Site: 1
```
Examples 14-5 and 14-6 show a different adjacency uptime for the site and overlay adjacencies because these are independent IS-IS interfaces, and the adjacencies form independently of each other. The site-id for an IS-IS neighbor is found in the output of `show otv internal adjacency`, as shown in Example 14-7. This provides information about which OTV EDs are part of the same site.

**Example 14-7  Verify the Site-ID of an OTV IS-IS Neighbor**

```
MX-2# show otv internal adjacency
Overlay Adjacency database

Overlay-Interface Overlay0 :
  System-ID  Dest Addr  Adj-State TM_State Adj-State inAS Site-ID
  Version
  64a0.e73e.12c2 10.1.22.1  default  default  UP    UP  0000.0000.0001*
  HW-St: Default N backup (null)

  64a0.e73e.12c4 10.2.43.1  default  default  UP    UP  0000.0000.0002*
  HW-St: Default N backup (null)

  6c9c.ed4d.d944 10.2.34.1  default  default  UP    UP  0000.0000.0002*
  HW-St: Default N backup (null)
```

**Note**  OTV has several event-history logs that are useful for troubleshooting. The `show otv isis internal event-history adjacency` command is used to review recent adjacency changes.

A point-to-point tunnel is created for each OTV ED that has an adjacency. These tunnels are used to transport OTV unicast packets between OTV EDs. The output of `show tunnel internal implicit otv brief` should have a tunnel present for each OTV ED reachable on the transport network. The output from NX-2 is shown in Example 14-8.
Example 14-8  OTV Dynamic Unicast Tunnels

```
NX-2# show tunnel internal implicit otv brief

+---------------------------------+-------+-------------+-------------+-------+
<table>
<thead>
<tr>
<th>Interface</th>
<th>Status</th>
<th>IP Address</th>
<th>Encap type</th>
<th>MTU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tunnel16384</td>
<td>up</td>
<td>--</td>
<td>GRE/IP</td>
<td>9178</td>
</tr>
<tr>
<td>Tunnel16385</td>
<td>up</td>
<td>--</td>
<td>GRE/IP</td>
<td>9178</td>
</tr>
<tr>
<td>Tunnel16386</td>
<td>up</td>
<td>--</td>
<td>GRE/IP</td>
<td>9178</td>
</tr>
</tbody>
</table>
```

Additional details about a specific tunnel is viewed with `show tunnel internal implicit otv tunnel_num [number]`. Example 14-9 shows detailed output for tunnel 16384. The MTU, transport protocol source, and destination address are shown, which allows a tunnel to be mapped to a particular neighbor. This output should be verified if a specific OTV ED is having a problem.

Example 14-9  Verify Detailed Dynamic Tunnel Parameters

```
NX-2# show tunnel internal implicit otv tunnel_num 16384
Tunnel16384 is up
  Admin State: up
  MTU 9178 bytes, BW 9 Kbit
  Tunnel protocol/transport GRE/IP
  Tunnel source 10.1.12.1, destination 10.2.43.1
  Transport protocol is in VRF "default"
  Rx
  0 packets input, 1 minute input rate 0 packets/sec
  Tx
  0 packets output, 1 minute output rate 0 packets/sec
  Last clearing of "show interface" counters never
```

When the OTV Adjacencies are established, the AED role is determined for each VLAN that is extended across the overlay using a hash function. The OTV IS-IS system-id is used along with the VLAN identifier to determine the AED role for each VLAN based on an ordinal value. The device with the lower system-id becomes AED for the even-numbered VLANs, and the device with the higher system-id becomes AED for the odd-numbered VLANs.

The `show otv vlan` command from NX-2 is shown in Example 14-10. The VLAN state column lists the current state as Active or Inactive. An Active state indicates this OTV ED is the AED for that VLAN and is responsible for forwarding packets across the overlay and advertising MAC address reachability for the VLAN. This is an important piece of information to know when troubleshooting to ensure the correct device is being investigated for a particular VLAN.
Example 14-10  Verify Which OTV ED Is the AED

```
NX-2# show otv vlan

OTV Extended VLANs and Edge Device State Information (* - AED)

Legend:
(NA) - Non AED, (VD) - Vlan Disabled, (OD) - Overlay Down
(DH) - Delete Holdown, (HW) - HW: State Down
(NFC) - Not Forward Capable

<table>
<thead>
<tr>
<th>VLAN</th>
<th>Auth. Edge Device</th>
<th>Vlan State</th>
<th>Overlay</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>NX-4</td>
<td>inactive(NA)</td>
<td>Overlay0</td>
</tr>
<tr>
<td>101*</td>
<td>NX-2</td>
<td>active</td>
<td>Overlay0</td>
</tr>
<tr>
<td>102</td>
<td>NX-4</td>
<td>inactive(NA)</td>
<td>Overlay0</td>
</tr>
<tr>
<td>103*</td>
<td>NX-2</td>
<td>active</td>
<td>Overlay0</td>
</tr>
<tr>
<td>104</td>
<td>NX-4</td>
<td>inactive(NA)</td>
<td>Overlay0</td>
</tr>
<tr>
<td>105*</td>
<td>NX-2</td>
<td>active</td>
<td>Overlay0</td>
</tr>
<tr>
<td>106</td>
<td>NX-4</td>
<td>inactive(NA)</td>
<td>Overlay0</td>
</tr>
<tr>
<td>107*</td>
<td>NX-2</td>
<td>active</td>
<td>Overlay0</td>
</tr>
<tr>
<td>108</td>
<td>NX-4</td>
<td>inactive(NA)</td>
<td>Overlay0</td>
</tr>
<tr>
<td>109*</td>
<td>NX-2</td>
<td>active</td>
<td>Overlay0</td>
</tr>
<tr>
<td>110</td>
<td>NX-4</td>
<td>inactive(NA)</td>
<td>Overlay0</td>
</tr>
</tbody>
</table>
```

Adjacency problems are typically caused by configuration error, a packet delivery problem for the OTV control-group in the transport network, or a problem with the site VLAN for the site adjacency.

For problems with an overlay adjacency, check the IP multicast state on the multicast router connected to the OTV ED's join interface. Each OTV ED should have a corresponding (S,G) mroute for the control-group. The L3 interface that connects the multicast router to the OTV ED should be populated in the Outgoing Interface List (OIL) for the (*, G) and all active sources (S,G) of the OTV control-group because of the IGMP join from the OTV ED.

The `show ip mroute` command from NX-1 is shown in Example 14-11. The (*, 239.12.12.12) entry has Port-channel 3 populated in the OIL by IGMP. For all active sources sending to 239.12.12.12, the OIL is populated with Port-channel 3 as well, which allows NX-2 to receive IS-IS hello and LSP packets from NX-4, NX-6, and NX-8. The source address for each Source, Group pair (S,G) are the other OTV ED's join interfaces sending multicast packets to the group.
Example 14-11  Verify Multicast Routing for the OTV Control-Group

NX-1# show ip mroute 239.12.12.12

IP Multicast Routing Table for VRF "default"

(*, 239.12.12.12/32), uptime: 1w1d, pim ip igmp
  Outgoing interface list: (count: 1)
    port-channel3, uptime: 16:17:45, igmp

(10.1.12.1/32, 239.12.12.12/32), uptime: 1w1d, ip mrib pim
  Incoming interface: port-channel13, RPF nbr: 10.1.12.1, internal
  Outgoing interface list: (count: 4)
    port-channel3, uptime: 16:17:45, mrib, (RPF)
    Vlan1101, uptime: 16:48:24, pim
    Ethernet3/17, uptime: 6d05h, pim
    Ethernet3/18, uptime: 1w1d, pim

(10.1.22.1/32, 239.12.12.12/32), uptime: 1w1d, pim mrib ip
  Incoming interface: Vlan1101, RPF nbr: 10.1.11.2, internal
  Outgoing interface list: (count: 1)
    port-channel3, uptime: 16:17:45, mrib

(10.2.34.1/32, 239.12.12.12/32), uptime: 1w1d, pim mrib ip
  Incoming interface: Ethernet3/18, RPF nbr: 10.1.13.3, internal
  Outgoing interface list: (count: 1)
    port-channel3, uptime: 16:17:45, mrib

(10.2.43.1/32, 239.12.12.12/32), uptime: 1w1d, pim mrib ip
  Incoming interface: Ethernet3/17, RPF nbr: 10.2.13.3, internal
  Outgoing interface list: (count: 1)
    port-channel3, uptime: 16:17:45, mrib

The presence of a (*, G) from IGMP for a group indicates that at minimum an IGMP join message was received by the router, and there is at least one interested receiver on that interface. A PIM join message is sent toward the PIM RP from the last hop router, and the (*, G) join state should be present along the multicast tree to the PIM RP. When a data packet for the group is received on the shared tree by the last hop router, in this case NX-1, a PIM (S, G) join message is sent toward the source. This messaging forms what is called the source tree, which is built to the first-hop router connected to the source. The source tree remains in place as long as the receiver is still interested in the group.
Example 14-12 shows how to verify the receipt of traffic with the `show ip mroute summary` command, which provides packet counters and bit-rate values for each source.

**Example 14-12  Verify the Current Bit-Rate of the OTV Control-Group**

```
NX-1# show ip mroute 239.12.12.12 summary
IP Multicast Routing Table for VRF "default"

Total number of routes: 6
Total number of (*,G) routes: 1
Total number of (S,G) routes: 4
Total number of (*,G-prefix) routes: 1
Group count: 1, rough average sources per group: 4.0

Source     packets   bytes      aps  pps    bit-rate   oifs
(*,G)      3      4326      1442 0     0.000  bps 1
10.1.12.1 927464 193003108 208  2     3.154  kbps 4
10.1.22.1 872869 173599251 198  3     3.844  kbps 1
10.2.34.1 1060046 203853603 192  3     3.261  kbps 1
10.2.43.1 1000183 203775760 203  3     3.466  kbps 1
```

Because IS-IS adjacency failures for the overlay are often caused by multicast packet delivery problems in the transport, it is important to understand what the multicast state on each router is indicating. The multicast role of each transport router must also be understood to provide context to the multicast routing table state. For example, is the device a first-hop router (FHR), PIM RP, transit router, or last-hop router (LHR)? In the network example, NX-1 is a PIM LHR, FHR, and RP for the control-group.

If NX-1 had no multicast state for the OTV control-group, it indicates that the IGMP join has not been received from NX-2. Because NX-1 is also a PIM RP for this group, it also indicates that none of the sources have been registered. If a (*, G) was present, but no (S, G), it indicates that the IGMP join was received from NX-2, but multicast data traffic from NX-4, NX-6, or NX-8 was not received by NX-1; therefore, the switchover to the source tree did not happen. At that point, troubleshooting moves toward the source and first-hop routers until the cause of the multicast problem is identified.

**Note**  Multicast troubleshooting is covered in Chapter 13, “Troubleshooting Multicast.”
The site adjacency is formed across the site VLAN. There must be connectivity between the OTV ED’s internal interface across the data center network for the IS-IS adjacency to form successfully. Example 14-13 contains the output of `show otv site` where the site adjacency is down, as indicated by the Partial state because the overlay adjacency with NX-4 is UP.

**Example 14-13  OTV Partial Adjacency**

```
NX-2# show otv site

Dual Adjacency State Description
  Full - Both site and overlay adjacency up
  Partial - Either site/overlay adjacency down
  Down - Both adjacencies are down (Neighbor is down/unreachable)
  (!) - Site-ID mismatch detected

Local Edge Device Information:
  Hostname NX-2
  System-ID 6c9c.ed4d.d942
  Site-Identifier 0000.0000.0001
  Site-VLAN 10 State is Up

Site Information for Overlay0:

  Local device is AED-Capable
  Neighbor Edge Devices in Site: 1

  Hostname         System-ID    Adjacency-    Adjacency-  AED-
                  State      Uptime    Capable
  -----------------------------------------------
  NX-4           64a0.e73e.12c2  Partial (!)   00:12:32   Yes

NX-2# show otv adjacency

Overlay Adjacency database

Overlay-Interface Overlay0 :

  Hostname         System-ID    Dest Addr    Up Time    State
                  System-ID    Dest Addr    Up Time    State
  -----------------------------------------------
  NX-4           64a0.e73e.12c2 10.1.22.1    00:01:57    UP
  NX-8           64a0.e73e.12c4 10.2.43.1    00:01:57    UP
  NX-6           6c9c.ed4d.d944 10.2.34.1    00:02:09    UP
```

The `show otv isis site` output confirms that the adjacency was lost on the site VLAN as shown in Example 14-14.
Example 14-14  Verify the OTV Site Adjacency

```
NX-2# show otv isis site

OTV-ISIS site-information for: default

BFD: Disabled

OTV-IS-IS site adjacency local database:

SNPA State Last Chg Hold   Fwd-state Site-ID    Version BFD
64a0.e73e.12c2 LOST 00:01:52 00:03:34 DOWN   0000.0000.0001 3   Disabled

OTV-IS-IS Site Group Information (as in OTV SDB):


Overlay State Next IIH Int Multi
Overlay0 Up 00:00:01 3 20

Overlay Active SG Last CSNP CSNP Int Next CSNP
Overlay0 239.12.12.12 ffff.ffff.ffff.ff-ff 2w1d Inactive

Neighbor SystemID: 64a0.e73e.12c2
```

The IS-IS adjacency being down indicates that IS-IS hellos (IIH Packets) are not being exchanged properly on the site VLAN. The transmit and receipt of IIH packets is recorded in the output of `show otv isis internal event-history iih`. Example 14-15 confirms that IIH packets are being sent, but none are being received across the site VLAN.

Example 14-15  NX-2 OTV IS-IS IIH Event-History

```
NX-2# show otv isis internal event-history iih | inc site
03:51:17.663263 isis_otv default [13901]: [13906]: Send L1 LAN IIH over site-vlan
len 1497 prio 6,dmac 0100.0cdf.dfdf
03:51:14.910759 isis_otv default [13901]: [13906]: Send L1 LAN IIH over site-vlan
len 1497 prio 6,dmac 0100.0cdf.dfdf
03:51:11.940991 isis_otv default [13901]: [13906]: Send L1 LAN IIH over site-vlan
len 1497 prio 6,dmac 0100.0cdf.dfdf
03:51:08.939666 isis_otv default [13901]: [13906]: Send L1 LAN IIH over site-vlan
len 1497 prio 6,dmac 0100.0cdf.dfdf
03:51:06.353274 isis_otv default [13901]: [13906]: Send L1 LAN IIH over site-vlan
len 1497 prio 6,dmac 0100.0cdf.dfdf
03:51:03.584122 isis_otv default [13901]: [13906]: Send L1 LAN IIH over site-vlan
len 1497 prio 6,dmac 0100.0cdf.dfdf
```
This event-history log confirms that the I IH packets are created, and the process
is sending them out to the site VLAN. The same event-history can be checked on
NX-4 to verify if the I IH packets are received. The output from NX-4 is shown in
Example 14-16, which indicates the I IH packets are being sent, but none are received
from NX-2.

Example 14-16  \textit{NX-4 OTV IS-IS I IH Event-History}

\begin{verbatim}
NX-4# show otv isis internal event-history iih | inc site
03:51:19.013078 isis_otv default [24209] [24210]: Send L1 LAN I IH over site-vlan
  len 1497 prio 6, dmac 0100.0cdf.dfdf
03:51:16.293081 isis_otv default [24209] [24210]: Send L1 LAN I IH over site-vlan
  len 1497 prio 6, dmac 0100.0cdf.dfdf
03:51:13.723065 isis_otv default [24209] [24210]: Send L1 LAN I IH over site-vlan
  len 1497 prio 6, dmac 0100.0cdf.dfdf
03:51:10.813105 isis_otv default [24209] [24210]: Send L1 LAN I IH over site-vlan
  len 1497 prio 6, dmac 0100.0cdf.dfdf
03:51:07.843102 isis_otv default [24209] [24210]: Send L1 LAN I IH over site-vlan
  len 1497 prio 6, dmac 0100.0cdf.dfdf
\end{verbatim}

The output in Example 14-15 and Example 14-16 confirms that both NX-2 and NX-4
are sending IS-IS I IH hellos to the site VLAN, but neither side is receiving packets from
the other OTV ED. At this point of the investigation, troubleshooting should follow the
VLAN across the L2 data center infrastructure to confirm the VLAN is properly con-
figured and trunked between NX-2 and NX-4. In this case, a problem was identified on
NX-3 where the site VLAN, VLAN 10, was not being trunked across the vPC peer-link.
This resulted in a Bridge Assurance inconsistency problem over the peer-link, as shown in
the output of Example 14-17.

Example 14-17  \textit{Verify Site-VLAN Spanning-Tree}

\begin{verbatim}
NX-1# show spanning-tree vlan 10 detail
VLAN0010 is executing the rstp compatible Spanning Tree protocol
  Bridge Identifier has priority 24576, sysid 10, address 0023.04ee.be01
  Configured hello time 2, max age 20, forward delay 15
  We are the root of the spanning tree
  Topology change flag not set, detected flag not set
  Number of topology changes 2 last change occurred 0:05:26 ago
    from port-channel12
  Times: hold 1, topology change 35, notification 2
    hello 2, max age 20, forward delay 15
  Timers: hello 0, topology change 0, notification 0
\end{verbatim}
Port 4096 (port-channel1, vPC Peer-link) of VLAN0010 is broken (Bridge Assurance Inconsistent, VPC Peer-link Inconsistent)

Port path cost 1, Port priority 128, Port Identifier 128.4096

Designated root has priority 32778, address 0023.04ee.be01
Designated bridge has priority 0, address 6c9c.ed4d.d941

Designated port id is 128.4096, designated path cost 0

Timers: message age 0, forward delay 0, hold 0

Number of transitions to forwarding state: 0

The port type is network

Link type is point-to-point by default

BPDU: sent 1534, received 0

After correcting the trunked VLAN configuration of the vPC peer-link, the OTV site adjacency came up on the site VLAN, and the dual adjacency state was returned to FULL. The adjacency transitions are viewed in the output of `show otv isis internal event-history adjacency` as shown in Example 14-18.

**Example 14-18**  
**OTV IS-IS Adjacency Event-History**

```
NX-2# show otv isis internal event-history adjacency
03:52:58.909967 isis_otv default [13901]:: LAN adj L1 64a0.e73e.12c2 over site-vlan - UP T 0
03:52:58.909785 isis_otv default [13901]:: LAN adj L1 64a0.e73e.12c2 over site-vlan - INIT (New) T -1
03:52:58.909776 isis_otv default [13901]:: isis_init_topo_adj LAN adj 1 64a0.e73e.12c2 over site-vlan - LAN MT-0
```

The first troubleshooting step for an adjacency problem is to ensure that both neighbors are generating and transmitting IS-IS hellos properly. If they are, start stepping through the transport or underlay network until the connectivity problem is isolated.

If the site VLAN was verified to be functional across the data center, the next step in troubleshooting an adjacency problem is to perform packet captures to determine which device is not forwarding the frames correctly. Chapter 2, “NX-OS Troubleshooting Tools,” covers the use of various packet capture tools available on NX-OS platforms that can be utilized to isolate the problem. An important concept to grasp is that even though these are control plane packets for OTV IS-IS on NX-2 and NX-4, as they are traversing the L3 transport network, they are handled as ordinary data plane packets.

**OTV IS-IS Topology Table**

After IS-IS adjacencies are formed on the overlay and site VLAN, IS-IS transmits and receives Protocol Data Units (PDU) including LSPs for the purpose of creating the OTV MAC routing table. Each OTV ED floods its LSP database so that all neighbors have a
consistent view of the topology. After LSPs are exchanged, the Shortest Path First (SPF) algorithm runs and constructs the topology with MAC addresses as leafs. Entries are then installed into the OTV MAC routing table for the purpose of traffic forwarding.

An example of the OTV IS-IS database is shown in Example 14-19. This output shows the LSP for NX-4 from the IS-IS database on NX-2.

**Example 14-19  The OTV IS-IS Database**

```
NX-2# show otv isis database
OTV-IS-IS Process: default LSP database VPN: Overlay0

OTV-IS-IS Level-1 Link State Database
LSPID            Seq Number  Checksum Lifetime  A/P/O/T
64a0.e73e.12c2.00-00 0x0000069F  0x643C  1198    0/0/0/1
64a0.e73e.12c4.00-00 0x00027EBC  0x13EA  1198    0/0/0/1
6c9c.ed4d.d942.00-00* 0x00000619  0x463D  1196    0/0/0/1
6c9c.ed4d.d942.01-00* 0x00000003  0x2278  0 (1198) 0/0/0/1
6c9c.ed4d.d944.00-00 0x0002AA3A  0x209E  1197    0/0/0/1
6c9c.ed4d.d944.01-00 0x0002790A  0xD43A  1199    0/0/0/1
```

The LSP lifetime shows that LSPs are only a few seconds old because the *Lifetime* counts from 1200 to zero. Issuing the command a few times may also show the *Seq Number* field incrementing, which indicates that the LSP is being updated by the originating IS-IS neighbor with changed information. This could cause OTV MAC routes to be refreshed and reinstalled as the SPF algorithm executes constantly. LSPs may refresh and get updated as part of normal IS-IS operation, but in this case the updates are happening constantly, which is abnormal in a steady-state.

To investigate the problem, check the LSP contents for changes over time. To understand which OTV ED is advertising which LSP, check the hostname to system-id mapping. The Hostname TLV provides a way to dynamically learn the system-id to hostname mapping for a neighbor. To identify which IS-IS database entries belong to which neighbors, use the `show otv isis hostname` command, as shown in Example 14-20. The asterisk (*) indicates the local system-id.

**Example 14-20  OTV IS-IS Dynamic Hostname**

```
NX-2# show otv isis hostname
OTV-IS-IS Process: default dynamic hostname table VPN: Overlay0

Level System ID    Dynamic hostname
1   64a0.e73e.12c2 NX-4
1   64a0.e73e.12c4 NX-8
1   6c9c.ed4d.d942* NX-2
1   6c9c.ed4d.d944 NX-6
```
The contents of an individual LSP are verified with the `show otv isis database detail [lsp-id]`. Example 14-21 contains the LSP received from NX-4 at NX-2 and contains several important pieces of information, such as neighbor and MAC address reachability, the site-id, and which device is the AED for a particular VLAN.

**Example 14-21  OTV IS-IS Database Detail**

```
NX-2# show otv isis database detail 64a0.e73e.12c2.00-00
OTV-IS-IS Process: default LSP database VPN: Overlay0

OTV-IS-IS Level-1 Link State Database
LSPID         Seq Number  Checksum Lifetime  A/P/O/T
64a0.e73e.12c2.00-00 0x000006BB  0xAFD6  1194    0/0/0/1
Instance : 0x000005D0
Area Address : 00
NLPID     : 0xCC 0x8E
Hostname   : NX-4        Length : 4
Extended IS  : 6c9c.ed4d.d944.01 Metric : 40
Vlan     : 100 : Metric   : 0
MAC Address   : 0000.0c07.ac64
Vlan     : 102 : Metric   : 0
MAC Address   : 0000.0c07.ac66
Vlan     : 104 : Metric   : 0
MAC Address   : 0000.0c07.ac68
Vlan     : 108 : Metric   : 0
MAC Address   : 0000.0c07.ac6c
Vlan     : 110 : Metric   : 1
MAC Address   : 0000.0c07.ac6e
Vlan     : 106 : Metric   : 1
MAC Address   : 0000.0c07.ac6a
Vlan     : 110 : Metric   : 1
MAC Address   : 64a0.e73e.12c1
Vlan     : 108 : Metric   : 1
MAC Address   : 64a0.e73e.12c1
Vlan     : 100 : Metric   : 1
MAC Address   : 64a0.e73e.12c1
Vlan     : 104 : Metric   : 1
MAC Address   : c464.135c.6600
MAC Address   : 64a0.e73e.12c1
Vlan     : 106 : Metric   : 1
MAC Address   : 64a0.e73e.12c1
Vlan     : 102 : Metric   : 1
MAC Address   : 6c9c.ed4d.d941
MAC Address   : 64a0.e73e.12c1
Site ID    : 0000.0000.0001
```
To determine what information is changing in the LSP, use the NX-OS `diff` utility. As shown in Example 14-22, the `diff` utility reveals that the Sequence Number is updated, and the LSP Lifetime has refreshed again to 1198. The changing LSP contents are related to HSRP MAC addresses in several VLANs extended by OTV.

### Example 14-22  OTV IS-IS LSP Updating Frequently

```diff
NX-2# show otv isis database detail 64a0.e73e.12c2.00-00 | diff
5,6c5,6
< 64a0.e73e.12c2.00-00 0x0001CD0E 0x0FF1 1196 0/0/0/1
< Instance : 0x0001CC23
---
> 64a0.e73e.12c2.00-00 0x0001CD11 0x193C 1198 0/0/0/1
> Instance : 0x0001CC26
10a11,12
> Vlan : 110 : Metric : 0
> MAC Address : 0000.0c07.ac6e
13,16d14
< Vlan : 108 : Metric : 0
< MAC Address : 0000.0c07.ac6c
< Vlan : 106 : Metric : 0
```
Chapter 14: Troubleshooting Overlay Transport Virtualization (OTV)

The MAC reachability information from the LSP is installed into the OTV MAC routing table. Each MAC address is installed with a next-hop known either via the site VLAN or from an OTV ED reachable across the overlay interface. The OTV MAC routing table in Example 14-23 confirms that MAC address entries are unstable and are refreshing. The Uptime for several entries is less than 1 minute and some were dampened with the (D) flag.

Example 14-23  Instability in the OTV MAC Routing Table

<table>
<thead>
<tr>
<th>VLAN</th>
<th>MAC-Address</th>
<th>Metric</th>
<th>Uptime</th>
<th>Owner</th>
<th>Next-hop(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0000.0c07.ac64</td>
<td>41</td>
<td>00:00:18</td>
<td>overlay</td>
<td>NX-8 (D)</td>
</tr>
<tr>
<td>101</td>
<td>0000.0c07.ac65</td>
<td>1</td>
<td>00:00:07</td>
<td>site</td>
<td>Ethernet3/5</td>
</tr>
<tr>
<td>102</td>
<td>0000.0c07.ac66</td>
<td>41</td>
<td>00:00:12</td>
<td>overlay</td>
<td>NX-8 (D)</td>
</tr>
<tr>
<td>103</td>
<td>0000.0c07.ac67</td>
<td>1</td>
<td>00:00:07</td>
<td>site</td>
<td>Ethernet3/5</td>
</tr>
<tr>
<td>104</td>
<td>0000.0c07.ac68</td>
<td>41</td>
<td>00:00:12</td>
<td>overlay</td>
<td>NX-8</td>
</tr>
<tr>
<td>105</td>
<td>0000.0c07.ac69</td>
<td>1</td>
<td>00:00:07</td>
<td>site</td>
<td>Ethernet3/5</td>
</tr>
<tr>
<td>106</td>
<td>0000.0c07.ac6a</td>
<td>41</td>
<td>00:00:30</td>
<td>overlay</td>
<td>NX-8</td>
</tr>
<tr>
<td>107</td>
<td>0000.0c07.ac6b</td>
<td>1</td>
<td>00:00:03</td>
<td>overlay</td>
<td>NX-6</td>
</tr>
<tr>
<td>108</td>
<td>0000.0c07.ac6c</td>
<td>41</td>
<td>00:00:18</td>
<td>overlay</td>
<td>NX-8 (D)</td>
</tr>
<tr>
<td>109</td>
<td>0000.0c07.ac6d</td>
<td>1</td>
<td>00:00:07</td>
<td>site</td>
<td>Ethernet3/5</td>
</tr>
<tr>
<td>110</td>
<td>0000.0c07.ac6e</td>
<td>41</td>
<td>00:00:12</td>
<td>overlay</td>
<td>NX-8 (D)</td>
</tr>
</tbody>
</table>

Additional information is obtained from the OTV event-traces. Because you are interested in the changes being received in the IS-IS LSP from a remote OTV ED, the show otv isis internal event-history spf-leaf is used to view what is changing and causing the routes to be refreshed in the OTV route table. This output is provided in Example 14-24.
Example 14-24  OTV IS-IS SPF Event-History

```
NX-2# show otv isis internal event-history spf-leaf | egrep "Process 0103-0000.0c07.ac67"
20:12:48.699301 isis_otv default [13901]: [13911]: Process 0103-0000.0c07.ac67 contained in 6c9c.ed4d.d944.00-00 with metric 0
20:12:45.060622 isis_otv default [13901]: [13911]: Process 0103-0000.0c07.ac67 contained in 6c9c.ed4d.d944.00-00 with metric 0
20:12:32.909267 isis_otv default [13901]: [13911]: Process 0103-0000.0c07.ac67 contained in 6c9c.ed4d.d944.00-00 with metric 1
20:12:30.743478 isis_otv default [13901]: [13911]: Process 0103-0000.0c07.ac67 contained in 6c9c.ed4d.d944.00-00 with metric 1
20:12:28.652719 isis_otv default [13901]: [13911]: Process 0103-0000.0c07.ac67 contained in 6c9c.ed4d.d944.00-00 with metric 0
20:12:26.470400 isis_otv default [13901]: [13911]: Process 0103-0000.0c07.ac67 contained in 6c9c.ed4d.d944.00-00 with metric 0
20:12:25.978913 isis_otv default [13901]: [13911]: Process 0103-0000.0c07.ac67 contained in 6c9c.ed4d.d944.00-00 with metric 0
20:12:13.239379 isis_otv default [13901]: [13911]: Process 0103-0000.0c07.ac67 contained in 6c9c.ed4d.d944.00-00 with metric 0
```

It is now apparent what is changing in the LSPs and why the lifetime is continually resetting to 1200. The metric is changing from zero to one.

The next step is to further investigate the problem at the remote AED that is originating the MAC advertisements across the overlay. In this particular case, the problem is caused by an incorrect configuration. The HSRP MAC addresses are being advertised across the overlay through OTV incorrectly. The HSRP MAC should be blocked using the First Hop Routing Protocol (FHRP) localization filter, as described later in this chapter, but instead it was advertised across the overlay resulting in the observed instability.

The previous example demonstrated a problem with the receipt of a MAC advertisement from a remote OTV ED. If a problem existed with MAC addresses not being advertised out to other OTV EDs from the local AED, the first step is to verify that OTV is passing the MAC addresses into IS-IS for advertisement. The `show otv isis mac redistribute route` command shown in Example 14-25 is used to verify that MAC addresses were passed to IS-IS for advertisement to other OTV EDs.

Example 14-25  MAC Address Redistribution into OTV IS-IS

```
NX-2# show otv isis mac redistribute route
OTV-IS-IS process: default VPN: Overlay0
OTV-IS-IS MAC redistribute route
0101-64a0.e73e.12c1, all
Advertised into L1, metric 1 LSP-ID 6c9c.ed4d.d942.00-00
```
The integrity of the IS-IS LSP is a critical requirement for the reliability and stability of the OTV control plane. Packet corruption problems or loss in the transport can affect both OTV IS-IS adjacencies as well as the advertisement of LSPs. Separate IS-IS statistics are available for the overlay and site VLAN, as shown in Examples 14-26 and 14-27, which provide valuable clues when troubleshooting an adjacency or LSP issue.

Example 14-26  \textit{OTV IS-IS Overlay Traffic Statistics}

\begin{verbatim}
NX-2# show otv isis traffic overlay0
OTV-IS-IS process: default
VPN: Overlay0
OTV-IS-IS Traffic for Overlay0:
PDU    Received    Sent RcvAuthErr OtherRcvErr ReTransmit
LAN-IIH   112327    37520     525     11     n/a
CSNP     100939    16964      0      0     n/a
PSNP     71166    19862      0      0     n/a
LSP     817782   280896      0      0      0
\end{verbatim}

Example 14-27  \textit{OTV IS-IS Site-VLAN Statistics}

\begin{verbatim}
NX-2# show otv isis site statistics
OTV-ISIS site-information for: default
OTV-IS-IS Broadcast Traffic statistics for site-vlan:
\end{verbatim}
OTV-IS-IS PDU statistics for site-vlan:

<table>
<thead>
<tr>
<th>PDU</th>
<th>Received</th>
<th>Sent</th>
<th>RcvAuthErr</th>
<th>OtherRcvErr</th>
<th>ReTransmit</th>
</tr>
</thead>
<tbody>
<tr>
<td>LAN-IIH</td>
<td>290557</td>
<td>432344</td>
<td>0</td>
<td>1</td>
<td>n/a</td>
</tr>
<tr>
<td>CSNP</td>
<td>68605</td>
<td>34324</td>
<td>0</td>
<td>0</td>
<td>n/a</td>
</tr>
<tr>
<td>PSNP</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>n/a</td>
</tr>
<tr>
<td>LSP</td>
<td>7</td>
<td>122</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

OTV-IS-IS Global statistics for site-vlan:

- SPF calculations: 0
- LSPs sourced: 2
- LSPs refreshed: 13
- LSPs purged: 0

Incrementing receive errors or retransmits indicate a problem with IS-IS PDUs, which may result in MAC address reachability problems. Incrementing `RcvAuthErr` indicates an authentication mismatch between OTV EDs.

**OTV IS-IS Authentication**

In some networks, using authentication for IS-IS may be desired. This is supported for OTV adjacencies built across the overlay by configuring IS-IS authentication on the overlay interface. Example 14-28 provides a sample configuration for IS-IS authentication on the overlay interface.

**Example 14-28 Configure OTV IS-IS Authentication**

```
NX-2# show running-config
! Output omitted for brevity
feature otv

otv site-vlan 10
key chain OTV-CHAIN
  key 0
    key-string 7 073c046f7c2c2d
interface Overlay0
description Site A
otv isis authentication-type md5
otv isis authentication key-chain OTV-CHAIN
otv join-interface port-channel3
otv control-group 239.12.12.12
otv data-group 232.1.1.0/24
otv extend-vlan 100-110
no shutdown
otv-isis default
otv site-identifier 0x1
```
OTV IS-IS authentication is enabled as verified with the `show otv isis interface overlay [overlay-number]` output in Example 14-29.

**Example 14-29  OTV IS-IS Authentication Parameters**

```
NX-2# show otv isis interface overlay 0
OTV-IS-IS process: default VPN: Overlay0
Overlay0, Interface status: protocol-up/link-up/admin-up
  IP address: none
  IPv6 address: none
  IPv6 link-local address: none
  Index: 0x0001, Local Circuit ID: 0x01, Circuit Type: L1
  Level1
  Adjacency server (local/remote) : disabled / none
  Adjacency server capability : multicast
  Authentication type is MD5
  Authentication keychain is OTV-CHAIN
  Authentication check specified
  LSP interval: 33 ms, MTU: 1400
  Level   Metric  CSNP      Next CSNP   Hello  Multi  Next IIH
  1       40      10 Inactive  20  3    00:00:15

  Level      Adjts  AdjtsUp Pri Circuit ID     Since
  1     0    0 64 6c9c.ed4d.d942.01 23:40:21
```

All OTV sites need to be configured with the same authentication commands for the overlay adjacency to form. Incrementing `RcvAuthErr` for LAN-IIH frames, as shown in the output of Example 14-30, indicates the presence of an authentication mismatch.

**Example 14-30  OTV IS-IS Authentication Error Statistics**

```
NX-2# show otv isis traffic overlay 0
OTV-IS-IS process: default
VPN: Overlay0
OTV-IS-IS Traffic for Overlay0:
  PDU     Received  Sent  RcvAuthErr OtherRcvErr ReTransmit
  LAN-IIH  111899   37370  260   11   n/a
  CSNP     100792  16937   0     0     n/a
  PSNP     71058   19832   0     0     n/a
  LSP     816541  280383   0     0     0
```

The output of `show otv adjacency` and `show otv site` varies depending on which adjacencies are down. The authentication configuration is applied only to the overlay interface, so it is possible the site adjacency is up even if one OTV ED at a site has authentication misconfigured for the overlay.
Example 14-31 shows that the overlay adjacency is down, but the site adjacency is still valid. In this scenario, the state is shown as *Partial*.

**Example 14-31  OTV Overlay IS-IS Adjacency Down**

```
NX-2# show otv adjacency
Overlay Adjacency database

NX-2# show otv site

Dual Adjacency State Description
Full - Both site and overlay adjacency up
Partial - Either site/overlay adjacency down
Down - Both adjacencies are down (Neighbor is down/unreachable)
(!) - Site-ID mismatch detected

Local Edge Device Information:
Hostname NX-2
System-ID 6c9c.ed4d.d942
Site-Identifier 0000.0000.0001
Site-VLAN 10 State is Up

Site Information for Overlay0:
Local device is not AED-Capable (No Overlay Remote Adjacency up)
Neighbor Edge Devices in Site: 1

<table>
<thead>
<tr>
<th>Hostname</th>
<th>System-ID</th>
<th>Adjacency-</th>
<th>Adjacency-</th>
<th>AED-</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>64a0.e73e.12c2</td>
<td>Partial</td>
<td>1w0d</td>
<td>Yes</td>
</tr>
</tbody>
</table>
```

**Adjacency Server Mode**

Starting in NX-OS release 5.2(l), *adjacency server mode* allows OTV to function over a unicast transport. Because a multicast capable transport is not used, an OTV ED in adjacency server mode must replicate IS-IS messages to each neighbor. This is less efficient because it requires each OTV ED to perform additional packet replications and transmit updates for each remote OTV ED.

A multicast transport allows the ED to generate only a single multicast packet, which is then replicated by the transport network. Therefore, it is preferred to use multicast mode whenever possible because of the increase in efficiency. However, in deployments where only two sites exist, or where multicast is not possible in the transport, adjacency server mode allows for a completely functional OTV deployment over IP unicast.
The OTV overlay configuration for each ED is configured to use the adjacency server unicast IP address as shown in Example 14-32. The role of the adjacency server is handled by a user-designated OTV ED. Each OTV ED registers itself with the adjacency server by sending OTV IS-IS hellos, which are transmitted from the OTV join interface as OTV encapsulated IP unicast packets. When the adjacency server forms an adjacency with a remote OTV ED, a list of OTV EDs is created dynamically. The adjacency server takes the list of known EDs and advertises it to each neighbor. All EDs then have a mechanism to dynamically learn about all other OTV EDs so that update messages are created and replicated to each remote ED.

Example 14-32  OTV ED Adjacency Server Mode Configuration on NX-4

```
NX-4# show run otv
! Output omitted for brevity
otv site-vlan 10

interface Overlay0
  otv join-interface port-channel3
  otv extend-vlan 100-110
  otv use-adjacency-server 10.1.12.1 unicast-only
  no shutdown
  otv site-identifier 0x1
```

Example 14-33 shows the configuration for NX-2, which is now acting as the adjacency server. When configuring an OTV ED in adjacency server mode, the `otv control-group [multicast group]` and `otv data-group [multicast-group]` configuration on each OTV ED shown in the previous examples must be removed. The `otv use-adjacency-server [IP address]` is then configured to enable OTV adjacency server mode and the `otv adjacency-server unicast-only` command specifies that NX-2 will be the adjacency server. The join interface and internal interface configurations remain unchanged from the previous examples in this chapter.

Example 14-33  OTV Adjacency Server Configuration on NX-2

```
NX-2# show run otv
! Output omitted for brevity
otv site-vlan 10

interface port-channel3
  description 7009A-Main-OTV Join
  mtu 9216
  ip address 10.1.12.1/24
  ip router ospf 1 area 0.0.0.0
  ip igmp version 3
```
Dynamically advertising a list of known OTV EDs saves the user from having to configure every OTV ED with all other OTV ED addresses to establish adjacencies. The process of registration with the adjacency server and advertisement of the OTV Neighbor List is shown in Figure 14-4. The site adjacency is still present but not shown in the figure for clarity.

![Figure 14-4 OTV EDs Register with the Adjacency Server](image-url)

After the OTV Neighbor List (oNL) is built, it is advertised to each OTV ED from the adjacency server as shown in Figure 14-5.
Each OTV ED then establishes IS-IS adjacencies with all other OTV EDs. Updates are sent with OTV encapsulation in IP unicast packets from each OTV ED. Each OTV ED must replicate its message to all other neighbors. This step is shown in Figure 14-6.

Example 14-34 contains the output of show otv adjacency from NX-4. After receiving the OTV Neighbor List from the adjacency Server, IS-IS adjacencies are formed with all other OTV EDs.

**Example 14-34  OTV Adjacency Server Mode IS-IS Neighbors**

```
NX-4# show otv adjacency
Overlay Adjacency database

Overlay-Interface Overlay0 :
  Hostname          System-ID        Dest Addr       Up Time  State
  NX-8      64a0.e73e.12c4 10.2.43.1    00:20:35 UP
  NX-2      6c9c.ed4d.d942 10.1.12.1    00:20:35 UP
  NX-6      6c9c.ed4d.d944 10.2.34.1    00:20:35 UP
```
Figure 14-6  OTV IS-IS Hellos in Adjacency Server Mode

An OTV IS-IS site adjacency is still formed across the site VLAN, as shown in the output of 
show otv site in Example 14-35.

Example 14-35  OTV Adjacency Server Mode Dual Adjacency

```
x4# show otv site

Dual Adjacency State Description
  Full  - Both site and overlay adjacency up
  Partial - Either site/overlay adjacency down
  Down  - Both adjacencies are down (Neighbor is down/unreachable)
  (!)   - Site-ID mismatch detected

Local Edge Device Information:
  Hostname NX-4
  System-ID 64a0.e73e.12c2
  Site-Identifier 0000.0000.0001
  Site-VLAN 10 State is Up
```
Site Information for Overlay0:

Local device is AED-Capable
Neighbor Edge Devices in Site: 1

<table>
<thead>
<tr>
<th>Hostname</th>
<th>System-ID</th>
<th>Adjacency-State</th>
<th>Adjacency-Uptime</th>
<th>AED-Capable</th>
</tr>
</thead>
<tbody>
<tr>
<td>NX-2</td>
<td>6c9c.ed4d.d942</td>
<td>Full</td>
<td>00:42:04</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Troubleshooting IS-IS adjacency and LSP advertisement problems in OTV adjacency server mode follows similar methodology as with OTV Multicast mode. The difference is that the packets are sent encapsulated in IP Unicast instead of multicasting across the transport network.

Redundant OTV adjacency servers are supported for resiliency purposes. However, the two adjacency servers operate independently, and they do not synchronize state with each other. If multiple adjacency servers are present, each OTV ED registers with each adjacency server. An OTV ED uses the replication list from the primary adjacency server until it is no longer available. If the adjacency with the primary adjacency server goes down, the OTV ED starts using the replication list received from the secondary adjacency server. If the primary OTV ED comes back up before a 10-minute timeout, the OTV EDs revert back to the primary replication list. If more than 10 minutes pass, a new replication-list is pushed by the primary when it finally becomes active again.

**OTV Control Plane Policing (CoPP)**

OTV control plane packets are subject to rate-limiting to protect the resources of the switch, just like any other packet sent to the supervisor. Excessive ARP traffic or OTV control plane traffic could impact the stability of the switch, causing high CPU or protocol adjacency flaps, so protection with CoPP is recommended.

The importance of CoPP is realized when the OTV ARP-ND-Cache is enabled. ARP Reply messages are snooped and added to the local cache so the OTV AED can answer ARP requests on behalf of the target host. These packets must be handled by the control plane and could cause policing drops or high CPU utilization if the volume of ARP traffic is excessive. The OTV ARP-ND-Cache is discussed in more detail later in this chapter.

The `show policy-map interface control-plane` command from the default VDC provides statistics for each control plane traffic class. If CoPP drops are present and ARP resolution failure is occurring, the solution is typically not to adjust the control plane.
policy to allow more traffic, but to instead track down the source of excessive ARP traffic. Ethanalyzer is a good tool for this type of problem along with the event histories for OTV.

Understanding and Verifying the OTV Data Plane

OTV was designed to transport L2 frames between sites in an efficient and reliable manner. Frames arriving at an OTV ED are Unicast, Multicast, or Broadcast, and each type of frame must be encapsulated for transport to the destination OTV ED with information provided by the OTV control plane.

The default overlay encapsulation for OTV is GRE, shown in Figure 14-7. This is also referred to as OTV 1.0 encapsulation.

When a frame arrives on the internal interface, a series of lookups are used to determine how to rewrite the packet for transport across the overlay. The original payload, ethertype, source MAC address, and destination MAC address are copied into the new OTV Encapsulated frame. The 802.1Q header is removed, and an OTV SHIM header is inserted. The SHIM header contains information about the VLAN and the overlay it belongs to. This field in OTV 1.0 is actually an MPLS-in-GRE encapsulation, where the MPLS label is used to derive the VLAN. The value of the MPLS label is equal to 32 + VLAN identifier. For this example, VLAN 101 is encapsulated as MPLS label 133. The outer IP header is added, which contains the source IP address of the local OTV ED and the destination IP address of the remote OTV ED.

Control plane IS-IS frames are encapsulated in a similar manner between OTV EDs across the overlay and also carry the same 42 bytes of OTV Overhead. The MPLS label used for IS-IS control plane frames is the reserved label 1, which is the Router Alert label.

Figure 14-7  OTV 1.0 Encapsulation

When a frame arrives on the internal interface, a series of lookups are used to determine how to rewrite the packet for transport across the overlay. The original payload, ethertype, source MAC address, and destination MAC address are copied into the new OTV Encapsulated frame. The 802.1Q header is removed, and an OTV SHIM header is inserted. The SHIM header contains information about the VLAN and the overlay it belongs to. This field in OTV 1.0 is actually an MPLS-in-GRE encapsulation, where the MPLS label is used to derive the VLAN. The value of the MPLS label is equal to 32 + VLAN identifier. For this example, VLAN 101 is encapsulated as MPLS label 133. The outer IP header is added, which contains the source IP address of the local OTV ED and the destination IP address of the remote OTV ED.

Control plane IS-IS frames are encapsulated in a similar manner between OTV EDs across the overlay and also carry the same 42 bytes of OTV Overhead. The MPLS label used for IS-IS control plane frames is the reserved label 1, which is the Router Alert label.
Note If a packet capture is taken in the transport, OTV 1.0 encapsulation is decoded as MPLS Pseudowire with no control-word using analysis tools, such as Wireshark. Unfortunately, at the time of this writing, Wireshark is not able to decode all the IS-IS PDUs used by OTV.

NX-OS release 7.2(0)DI(I) introduced the option of UDP encapsulation for OTV when using F3 or M3 series modules in the Nexus 7000 series switches. The OTV 2.5 UDP encapsulation is shown in Figure 14-8.

![Figure 14-8 OTV 2.5 Encapsulation](image)

Ethernet Frames arriving from the OTV internal interface have the original payload, ethertype, 802.1Q header, source MAC address, and destination MAC address copied into the new OTV 2.5 Encapsulated frame. The OTV 2.5 encapsulation uses the same packet format as Virtual Extensible LAN (VxLAN), which is detailed in RFC 7348.

The OTV SHIM header contains information about the Instance and Overlay. The instance is the table identifier that should be used at the destination OTV ED to lookup the destination, and the overlay identifier is used by the control plane packets to identify packets belonging to a specific overlay. A control plane packet has the VxLAN Network ID (VNI) bit set to False (zero), while an encapsulated data frame has this value set to True (one). The UDP header contains a variable source port and destination port of 8472.

Fragmentation of OTV frames containing data packets becomes a concern if the transport MTU is not at least 1550 bytes with OTV 2.5, or 1542 bytes with OTV 1.0. This is based on the assumption that a host in the data center has an interface MTU of 1500 bytes and attempts to send full MTU sized frames. When the OTV encapsulation is added, the packet no longer fits into the available MTU size.

The minimum transport MTU requirement for control plane packets is either 1442 for multicast transport, or 1450 for unicast transport in adjacency server mode. OTV sets the Don't Fragment bit in the outer IP header to ensure that no OTV control plane or data plane packets become fragmented in the transport network. If MTU restrictions exist, it could result in OTV IS-IS adjacencies not forming, or the loss of frames for data traffic when the encapsulated frame size exceeds the transport MTU.
Note  The OTV encapsulation format must be the same between all sites (GRE or UDP) and is configured with the global configuration command `otv encapsulation-format ip [gre | udp].`

OTV ARP Resolution and ARP-ND-Cache

When a host communicates with another host in the same IP subnet, the communication begins with the source host resolving the MAC address of the destination host with ARP. ARP messages are shown between Host A and Host C, which are part of the same 10.101.0.0/16 subnet in Figure 14-9.

![Figure 14-9  ARP Request and Reply](image)

Host A broadcasts an ARP request message to the destination MAC address ff:ff:ff:ff:ff:ff with a target IP address of 10.101.2.1. This frame is sent out of all ports that belong to the same VLAN in the L2 switch, including the OTV internal interface of NX-2 and the port connected to Host B. Because NX-2 is an OTV ED for Data Center 1, it receives the frame and encapsulates it using the OTV control-group of 239.12.12.12. NX-2 also creates a MAC address table entry for Host A, known via the internal interface. Host A’s MAC is advertised from NX-2 across the overlay through the IS-IS control plane, providing reachability information to all other OTV EDs. The control-group multicast frame from NX-2 traverses the transport underlay network until it reaches NX-6 where the multicast OTV encapsulation is removed and the frame is sent out of the OTV internal interface toward Host C. Host C processes the broadcast frame and recognizes the IP address as its own. Host C then issues the ARP reply to Host A, which is sent to NX-6. NX-6 at this point has an entry in the OTV MAC routing table for Host A with an IP next-hop of NX-2 since the IS-IS update was received. There is also a MAC address table entry for Host A in VLAN101 pointing to the overlay interface. As the ARP reply from Host C is received at NX-6, a local MAC address table entry is created pointing to the OTV internal interface. This MAC address entry is then advertised to all remote OTV EDs through IS-IS, just as NX-2 did for Host A.
NX-6 then encapsulates the ARP reply and sends it across the overlay to NX-2 in Data Center 1. NX-2 removes the OTV encapsulation from the frame and sends it out of the internal interface where it reaches Host A, following the MAC address table of the VLAN.

The **OTV ARP-ND-Cache** is populated by listening to ARP reply messages. The initial ARP request is sent to all OTV EDs via the OTV control-group. When the ARP reply comes back using the OTV control-group, each OTV ED snoops the reply and builds an entry in the cache. If Host B were to send an ARP request for Host C, NX-2 replies to the ARP request on behalf of Host C, using the cached entry created previously, which reduces unnecessary traffic across the overlay.

**Note** If multiple OTV EDs exist at a site, only the AED forwards packets onto the overlay, including ARP request and replies. The AED is also responsible for advertising MAC address reachability to other OTV EDs through the IS-IS control plane.

The ARP-ND-Cache is populated in the same way for multicast mode or adjacency server mode. With adjacency server mode, the ARP request and response are encapsulated as OTV Unicast packets and replicated for the remote OTV EDs.

If hosts are unable to communicate with other hosts across the overlay, verify the ARP-ND-Cache to ensure it does not contain any stale information. Example 14-36 demonstrates how to check the local ARP-ND-Cache on NX-2.

**Example 14-36  Verify the ARP ND-Cache**

```
NX-2# show otv arp-nd-cache
OTV ARP/ND L3->L2 Address Mapping Cache

Overlay Interface Overlay0
VLAN MAC Address     Layer-3 Address   Age    Expires In
101  442b.03ec.cb00   10.101.2.1     00:02:29 00:06:07
```

OTV also keeps an event-history for ARP-ND cache activity, which is viewed with `show otv internal event-history arp-nd`. Example 14-37 shows this output from the AED for the VLAN 100.

**Example 14-37  ARP ND-Cache Event-History**

```
NX-4# show otv internal event-history arp-nd
ARP-ND events for OTV Process
02:33:17.816397 otv [9790]: [9810]: Updating arp nd cache entry in PSS TLVU. Overlay:249 Mac Info: 0100-442b.03ec.cb00 L3 addr: 10.100.2.1
```

The OTV ARP-ND cache timer is configurable from 60 to 86400 seconds. The default value is 480 seconds or 8 minutes, plus an additional 2-minute grace-period. During the grace-period an AED forwards ARP requests across the overlay so that the reply refreshes the entry in the cache. It is recommended to have the ARP-ND cache time value lower than the MAC aging timer. By default, the MAC aging timer is 30 minutes.

It is possible to disable the OTV ARP-ND-Cache by configuring `no otv suppress-arp-nd` under the overlay interface. The result of this configuration is that all ARP requests are forwarded across the overlay and no ARP reply messages are cached.

**Note** The ARP-ND-Cache is enabled by default. In some environments with a lot of ARP activity, it may cause the CPU of the OTV ED to become high or experience CoPP drops because the supervisor CPU must handle the ARP traffic to create the cache entries.

**Broadcasts**

Broadcast frames received by an OTV ED on the internal interface are forwarded across the overlay by the AED for the extended VLAN. Broadcast frames, such as ARP request, are encapsulated into an L3 multicast packet where the source address is the local OTV EDs join interface, and the group is the OTV Control-group address. The multicast packet is sent to the transport where it gets replicated to each remote OTV ED that has joined the control-group.

When using a multicast enabled transport, OTV allows for the configuration of a dedicated `otv broadcast-group`, as shown in Example 14-38. This allows the operator to separate the OTV control-group from the broadcast group for easier troubleshooting and to allow different handling of the packets based on group address. For example, a different PIM rendezvous point could be defined for each group, or a different Quality of Service (QoS) treatment could be applied to the control-group and broadcast-group in the transport.

**Example 14-38** Dedicated OTV Broadcast Group

```
NX-2# show run otv
! Output omitted for brevity
interface Overlay0
description Site A
```
OTV EDs operating in adjacency server mode without a multicast-enabled transport encapsulate broadcast packets with an OTV unicast packet and replicate a copy to each remote OTV ED using head-end replication.

With either multicast or unicast transport, when the packet is received by the remote OTV ED, the outer L3 packet encapsulation is removed. The broadcast frame is then forwarded to all internal facing L2 ports in the VLAN by the AED.

**Unknown Unicast Frames**

The default behavior for OTV is to only flood frames to an unknown unicast MAC address on the internal interface. These packets are not forwarded across the overlay. This optimization is allowed because OTV operates under the assumption that there are no silent hosts, and an OTV ED sees traffic from all hosts eventually on the internal interface. After that traffic is received, it populates the MAC address table in the VLAN, and the MAC address is advertised by IS-IS to all OTV EDs.

There are situations where a silent host is unavoidable. To allow these hosts to function, OTV provides a configuration option to allow selective unicast flooding beginning in NX-OS 6.2(2). Example 14-39 provides a configuration example to allow flooding of packets to a specific destination MAC address in VLAN 101 across the overlay.

**Example 14-39  Selective Unicast Flooding**

```
NX-2# show run otv
! Output omitted for brevity
feature otv
otv site-identifier 0x1
otv flood mac C464.135C.6600 vlan 101
```

The result of adding this command is a static OTV route entry for the VLAN, which causes traffic to flow across the overlay, as shown in Example 14-40.

**Example 14-40  OTV Routing Table with Selective Unicast Flooding**

```
NX-2# show otv route vlan 101
```

OTV Unicast MAC Routing Table For Overlay0
OTV Unicast Traffic with a Multicast Enabled Transport

Host-to-host communication begins with an ARP request for the destination, as shown previously in Figure 14-9. After this ARP request and reply exchange is finished, the OTV ED at each site has a correctly populated OTV MAC routing table and MAC address table for both hosts.

Figure 14-10 depicts the traffic flow in VLAN 103 between Host A in Data Center 1 and Host C in Data Center 2.

Traffic from Host A is first sent to the L2 switch where it has an 802.1Q VLAN tag added for VLAN 103. The frames follow the MAC address table entries at the L2 switch across the trunk port to reach NX-2 on the OTV internal interface Ethernet3/5. When the packets arrive at NX-2, it performs a MAC address table lookup in the VLAN to determine how to reach Host C’s MAC address 442b.03ec.cb00. The MAC address table of NX-2 is shown in Example 14-41.

Example 14-41  MAC Address Table Entry for Host C

```
NX-2# show mac address-table dynamic vlan 103
Note: MAC table entries displayed are getting read from software.
Use the 'hardware-age' keyword to get information related to 'Age'

Legend:
* - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
age - seconds since last seen, + - primary entry using vPC Peer-Link, E - EVPN entry
```
The MAC address table indicates that Host C’s MAC is reachable across the overlay, which means that the OTV MAC Routing table (ORIB) should be used to obtain the IP next-hop and encapsulation details. The ORIB indicates how to reach the remote OTV ED that advertised the MAC address to NX-2 via IS-IS, which is NX-6 in this example.

Note If multiple OTV EDs exist at a site, ensure the data path is being followed to the AED for the VLAN. This is verified with the `show otv vlan` command. Under normal conditions the MAC forwarding entries across the L2 network should lead to the AED’s internal interface.

NX-2 is the AED for VLAN103 as shown in Example 14-42.

Example 14-42 Verify the AED for VLAN 103

```
NX-2# show otv vlan
OTV Extended VLANS and Edge Device State Information (* - AED)

Legend:
(NA) - Non AED, (VD) - Vlan Disabled, (OD) - Overlay Down
(DH) - Delete Holdown, (HW) - HW: State Down
(NFC) - Not Forward Capable

<table>
<thead>
<tr>
<th>VLAN</th>
<th>Auth. Edge Device</th>
<th>Vlan State</th>
<th>Overlay</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>NX-4</td>
<td>inactive(NA)</td>
<td>Overlay0</td>
</tr>
<tr>
<td>101*</td>
<td>NX-2</td>
<td>active</td>
<td>Overlay0</td>
</tr>
<tr>
<td>102</td>
<td>NX-4</td>
<td>inactive(NA)</td>
<td>Overlay0</td>
</tr>
<tr>
<td>103*</td>
<td>NX-2</td>
<td>active</td>
<td>Overlay0</td>
</tr>
</tbody>
</table>
```
After verifying the AED state for VLAN 103 to ensure you are looking at the correct device, check the ORIB to determine which remote OTV ED will receive the encapsulated frame from NX-2. The ORIB for NX-2 is shown in Example 14-43.

**Example 14-43  Verify the ORIB Entry for Host C**

```
NX-2# show otv route vlan 103

OTV Unicast MAC Routing Table For Overlay0

<table>
<thead>
<tr>
<th>VLAN</th>
<th>MAC-Address</th>
<th>Metric</th>
<th>Uptime</th>
<th>Owner</th>
<th>Next-hop(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>103</td>
<td>0000.0c07.ac67</td>
<td>1</td>
<td>00:13:43</td>
<td>site</td>
<td>Ethernet3/5</td>
</tr>
<tr>
<td>103</td>
<td>442b.03ec.cb00</td>
<td>42</td>
<td>00:02:44</td>
<td>overlay</td>
<td>NX-6</td>
</tr>
<tr>
<td>103</td>
<td>64a0.e73e.12c1</td>
<td>1</td>
<td>00:13:43</td>
<td>site</td>
<td>Ethernet3/5</td>
</tr>
<tr>
<td>103</td>
<td>64a0.e73e.12c3</td>
<td>42</td>
<td>00:13:28</td>
<td>overlay</td>
<td>NX-6</td>
</tr>
<tr>
<td>103</td>
<td>6c9c.ed4d.d943</td>
<td>42</td>
<td>00:02:56</td>
<td>overlay</td>
<td>NX-6</td>
</tr>
<tr>
<td>103</td>
<td>c464.135c.6600</td>
<td>1</td>
<td>00:02:56</td>
<td>site</td>
<td>Ethernet3/5</td>
</tr>
</tbody>
</table>
```

Recall that the ORIB data is populated by the IS-IS LSP received from NX-6, which indicates MAC address 442b.03ec.cb00 is an attached host. This is confirmed by obtaining the system-id of NX-6 in `show otv adjacency`, and then finding the correct LSP in the output of `show otv isis database detail`.

At the AED originating the advertisement, the redistribution from the local MAC table into OTV IS-IS is verified on NX-6 using the `show otv isis redistribute route` command, which is shown in Example 14-44.

```
NX-6# show otv isis redistribute route

! Output omitted for brevity

OTV-IS-IS process: default VPN: Overlay0
OTV-IS-IS MAC redistribute route
```

At this point, it has been confirmed that NX-6 is the correct remote OTV ED to receive frames with a destination MAC address of 442b.03ec.cb00 in VLAN 103. The next step in delivering the packet to Host C is for NX-2 to rewrite the packet to impose the OTV header and send the encapsulated frame into the transport network from the join interface.

OTV uses either UDP or GRE encapsulation, and in this example the default GRE encapsulation is being used. There is a point-to-point tunnel created dynamically for each remote OTV ED that has formed an adjacency with the local OTV ED. These tunnels are viewed with `show tunnel internal implicit otv detail`, as shown in Example 14-45.

**Example 14-44  MAC Table Redistribution into OTV IS-IS**

```
NX-6# show otv isis redistribute route

! Output omitted for brevity

OTV-IS-IS process: default VPN: Overlay0
OTV-IS-IS MAC redistribute route
```
Chapter 14: Troubleshooting Overlay Transport Virtualization (OTV)

Example 14-45  Dynamic Tunnel Encapsulation for NX-6

```
NX-6# show tunnel internal implicit otv detail
! Output omitted for brevity
Tunnel16389 is up
  Admin State: up
  MTU 9178 bytes, BW 9 Kbit
  Tunnel protocol/transport GRE/IP
  Tunnel source 10.1.12.1, destination 10.2.34.1
  Transport protocol is in VRF "default"
  Rx
    720357 packets input, 1 minute input rate 1024 packets/sec
  Tx
    715177 packets output, 1 minute output rate 1027 packets/sec
  Last clearing of "show interface" counters never
```

The dynamic tunnels represent the software forwarding component of the OTV encapsulation. The hardware forwarding component for the OTV encapsulation is handled by performing multiple passes through the line card forwarding engine to derive the correct packet rewrite that includes the OTV encapsulation header.

**Note**  The verification of the packet rewrite details in hardware varies depending on the type of forwarding engine present in the line card. Verify the adjacencies, MAC address table, ORIB, and tunnel state before suspecting a hardware programming problem. If connectivity fails despite correct control plane programming, and MAC addresses are learned, engage the Cisco TAC for support.

After the OTV MAC-in-IP encapsulation is performed by NX-2, the packet traverses the Layer 3 transport network with a unicast OTV header appended. The source IP address is the join interface of NX-2 and the destination IP address is the join interface of NX-6. The Layer 3 packet arrives on the OTV join interface of NX-6, which must remove the OTV encapsulation and look up the destination.

The destination IP address of the outer packet header is the OTV join interface address of NX-6, 10.2.34.1. In a similar manner to the encapsulation of OTV, removing the OTV encapsulation also requires multiple forwarding engine passes on the receiving line card.
Because the outer destination IP address belongs to NX-6, it will strip the outer IP header and look into the OTV shim header where the VLAN ID is found. The information from this lookup is originated from the ORIB, which contains the VLAN, MAC address, and destination interface, as shown in Example 14-46.

**Example 14-46  ORIB Entry for Host C on NX-6**

```yaml
NX-6# show otv route
! Output omitted for brevity

OTV Unicast MAC Routing Table For Overlay0

<table>
<thead>
<tr>
<th>VLAN</th>
<th>MAC-Address</th>
<th>Metric</th>
<th>Uptime</th>
<th>Owner</th>
<th>Next-hop(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>103</td>
<td>0000.0c07.ac67</td>
<td>1</td>
<td>4d00h</td>
<td>site</td>
<td>port-channel3</td>
</tr>
<tr>
<td>103</td>
<td>442b.03ec.cb00</td>
<td>1</td>
<td>00:44:32</td>
<td>site</td>
<td>port-channel3</td>
</tr>
<tr>
<td>103</td>
<td>64a0.e73e.12c1</td>
<td>42</td>
<td>4d00h</td>
<td>overlay</td>
<td>NX-2</td>
</tr>
<tr>
<td>103</td>
<td>64a0.e73e.12c3</td>
<td>1</td>
<td>4d00h</td>
<td>site</td>
<td>port-channel3</td>
</tr>
<tr>
<td>103</td>
<td>6c9c.ed4d.d943</td>
<td>1</td>
<td>4d00h</td>
<td>site</td>
<td>port-channel3</td>
</tr>
<tr>
<td>103</td>
<td>c464.135c.6600</td>
<td>42</td>
<td>4d00h</td>
<td>overlay</td>
<td>NX-2</td>
</tr>
</tbody>
</table>
```

The next-pass through the forwarding engine performs a lookup on the VLAN MAC address table to find the correct outgoing interface and physical port. The MAC address table of NX-6 is shown in Example 14-47.

**Example 14-47  MAC Address Table Entry for Host C on NX6**

```yaml
NX-6# show mac address-table dynamic vlan 103
Note: MAC table entries displayed are getting read from software.
Use the 'hardware-age' keyword to get information related to 'Age'

Legend:
* - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
age - seconds since last seen, * - primary entry using vPC Peer-Link, E - EVPN entry
(T) - True, (F) - False, ~~~ - use 'hardware-age' keyword to retrieve age info

<table>
<thead>
<tr>
<th>VLAN/BD</th>
<th>MAC Address</th>
<th>Type</th>
<th>age</th>
<th>Secure NTFY Ports/SWID.SSID.LID</th>
</tr>
</thead>
<tbody>
<tr>
<td>* 103</td>
<td>0000.0c07.ac67</td>
<td>dynamic</td>
<td>~~~</td>
<td>F F Po3</td>
</tr>
<tr>
<td>* 103</td>
<td>442b.03ec.cb00</td>
<td>dynamic</td>
<td>~~~</td>
<td>F F Po3</td>
</tr>
<tr>
<td>O 103</td>
<td>64a0.e73e.12c1</td>
<td>dynamic</td>
<td>-</td>
<td>F F Overlay0</td>
</tr>
<tr>
<td>* 103</td>
<td>64a0.e73e.12c3</td>
<td>dynamic</td>
<td>~~~</td>
<td>F F Po3</td>
</tr>
<tr>
<td>* 103</td>
<td>6c9c.ed4d.d943</td>
<td>dynamic</td>
<td>~~~</td>
<td>F F Po3</td>
</tr>
<tr>
<td>O 103</td>
<td>c464.135c.6600</td>
<td>dynamic</td>
<td>-</td>
<td>F F Overlay0</td>
</tr>
</tbody>
</table>
```
The frame exits Port-channel 3 on the L2 trunk with a VLAN tag of 103. The L2 switch in data center 2 receives the frame and performs a MAC address table lookup to find the port where Host C is connected and delivers the frame to its destination.

**Note** Troubleshooting unicast data traffic when using the adjacency server mode follows the same methodology used for a multicast enabled transport. The difference is that any control plane messages are exchanged between OTV EDs using a unicast encapsulation method and replicated by the advertising OTV ED to all adjacent OTV EDs. The host-to-host data traffic is still MAC-in-IP unicast encapsulated from source OTV ED to the destination OTV ED.

**OTV Multicast Traffic with a Multicast Enabled Transport**

OTV provides support for multicast traffic to be forwarded across the overlay in a seamless manner. The source and receiver hosts do not need to modify their behavior to exchange L2 multicast traffic across an OTV network between sites.

In a traditional L2 switched network, the receiver host sends an Internet Group Management Protocol (IGMP) membership report to indicate interest in receiving the traffic. The L2 switch is typically enabled for IGMP snooping, which listens for these membership reports to optimize flooding of multicast traffic to only the ports where there are interested receivers.

IGMP snooping must also learn where multicast routers (mouters) are connected. Any multicast traffic must be forwarded to an mrouter so that interested receivers on other L3 networks can receive it. The mrouter is also responsible for registering the source with a rendezvous point if PIM ASM is being used. IGMP snooping discovers mouters by listening for Protocol Independent Multicast (PIM) hello messages, which indicate an L3 capable mrouter is present on that port. The L2 forwarding table is then updated to send all multicast group traffic to the mrouter, as well as any interested receivers. OTV EDs use a dummy PIM Hello message to draw multicast traffic and IGMP membership reports to the OTV ED's internal interface.

OTV maintains its own mroute table for multicast forwarding just as it maintains an OTV routing table for unicast forwarding. There are three types of OTV mroute entries, which are described as VLAN, Source, and Group. The purpose of each type is detailed in Table 14-2.

<table>
<thead>
<tr>
<th>Type</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>(V, *, *)</td>
<td>Created when a local mrouter is present in the VLAN, discovered by IGMP snooping. Used to forward traffic to the mrouter for all sources, and all groups.</td>
</tr>
<tr>
<td>(V, *, G)</td>
<td>Created when an IGMP membership report is received for group G. The interface on which the membership report was received is added to the Outgoing Interface (OIF) of the mroute.</td>
</tr>
<tr>
<td>(V, S, G)</td>
<td>Created when source S sends multicast traffic to group G, or as a result of receiving an IS-IS Group Membership Active Source (GMAS-TLV) with (S, G).</td>
</tr>
</tbody>
</table>
The OTV IS-IS control plane protocol is utilized to allow hosts to send and receive multicast traffic within an extended VLAN between sites without the need to send IGMP messages across the overlay. Figure 14-11 shows a simple OTV topology where Host A is a multicast source for group 239.100.100.100, and Host C is a multicast receiver. Both Host A and Host C belong to VLAN 103.

**Figure 14-11  Multicast Traffic Across OTV with Multicast Transport**

In this example, the L3 transport network is enabled for IP multicast. Each OTV ED is configured with a range of Source Specific Multicast (SSM) groups, referred to as the Delivery Group or data-group, which may be used interchangeably. The delivery group configuration of NX-6 is highlighted in the configuration sample provided in Example 14-48.

**Example 14-48  OTV SSM Data-Groups**

```
NX-6# show running-config interface overlay 0
interface Overlay0
    description Site B
    otv join-interface Ethernet3/41
    otv control-group 239.12.12.12
    otv data-group 232.1.1.0/24
    otv extend-vlan 100-110
    no shutdown
```

The delivery group must be coordinated with the L3 transport to ensure that PIM SSM is supported and that the correct range of groups are defined for use as SSM groups. Each OTV ED is configured with the same range of `otv data-groups`, and each OTV ED can be a source for the SSM group. Remote OTV EDs join the SSM group in the transport to receive multicast frames from a particular OTV ED acting as source. The signaling of which SSM group to use is accomplished with IS-IS advertisements between OTV EDs to allow for discovery of active sources and receivers at each site.

The *site group* is the multicast group that is being transported across the overlay using the delivery group. In Figure 14-11, the site group is 239.100.100.100 sourced by Host A and received by Host C. Essentially, OTV is using a *multicast-in-multicast* OTV...
encapsulation scheme to send the site group across the overlay using the delivery group in the transport network.

Troubleshooting is simplified by splitting the end-to-end packet delivery mechanism into two distinct layers of focus: the site group and the delivery group. At the source end, the site group troubleshooting is focused on ensuring that multicast data frames from the source are arriving at the internal interface of the AED for the VLAN. At the receiving site, site group troubleshooting must verify that a receiver has expressed interest in the group by sending an IGMP membership report. IGMP snooping must have the correct ports to reach the receivers from the OTV AEDs internal interface, through any L2 switches in the path. In the transport network, the delivery group must be functional so that any OTV ED acting as a source host successfully sends the multicast-in-multicast OTV traffic into the transport for replication and delivery to the correct OTV ED receivers.

For multicast sent by Host A to be successfully received by Host C, some prerequisite steps must occur. The OTV AED’s internal interface must be seen by the L2 switch as an mrouter port. This is required so that any IGMP membership reports from a receiver are sent to the AED, and any multicast traffic is also flooded to the AED’s OTV internal interface. To achieve this, OTV sends a dummy PIM hello with a source IP address of 0.0.0.0 on the internal interface for each VLAN extended by OTV. The purpose is not to form a PIM neighbor on the VLAN, but to force the detection of an mrouter port by any attached L2 switch, as depicted in Figure 14-12.

![OTV Dummy PIM Hello Messages](image)

Figure 14-12  OTV Dummy PIM Hello Messages

An Ethanalyzer capture of the PIM dummy hello packet from NX-6 on VLAN 103 is shown in Example 14-49.

Example 14-49  Dummy PIM Hello Captured in Ethanalyzer

```
Type: IP (0x0800)
Internet Protocol Version 4, Src: 0.0.0.0 (0.0.0.0),Dst: 224.0.0.13 (224.0.0.13)
Version: 4

```
Example 14-50 shows the IGMP snooping status of the L2 switch in Data Center 2 after receiving the PIM dummy hello packets on VLAN103 from NX-6.
Example 14-50  NX-6 Detected as an MROUTER Port by IGMP Snooping

When Host C’s IGMP membership report message reaches NX-6, it is snooped on the internal interface and added to the OTV mroute table as an IGMP created entry. Remember that any switch performing IGMP snooping must forward all IGMP membership reports to mrouter ports.

Example 14-51 shows the OTV mroute table from NX-6 with the IGMP created (V, *, G) entry and Outgoing Interface (OIF) of Port-channel 3 where the membership report was received.

Example 14-51  OTV MROUTE State on NX-6

NX-6 then builds an IS-IS message to advertise the group membership (GM-Update) to all OTV EDs. NX-2 in Data Center 1 receives the IS-IS GM-Update, as shown in Example 14-52. NX-6 is identified by the IS-IS system-id of 6c9c.ed4d.d944. The correct LSP to check is confirmed with the output of show otv adjacency, which lists the system-id of each OTV ED IS-IS neighbor.

Example 14-52  OTV IS-IS MGROUP Database on NX-2
Group-Address :   IP Multicast : Vlan : 103    Groups : 1
  Group : 239.100.100.100 Sources : 0
  Digest Offset : 0

Note  At this point only Host C joined the multicast group, and there are no sources actively sending to the group.

NX-2 installs an OTV mroute entry in response to receiving the IS-IS GM-Update from NX-6, as shown in Example 14-53. The OIF on NX-2 is the overlay interface. The r indicates the receiver is across the overlay.

Example 14-53  OTV MROUTE Entry on NX-2

NX-2# show otv mroute

OTV Multicast Routing Table For Overlay0

(103, *, 239.100.100.100), metric: 0, uptime: 00:00:47, overlay[r]
  Outgoing interface list: (count: 1)
  Overlay0, uptime: 00:00:47, isis_otv-default

Host A now begins sending traffic to the site group 239.100.100.100 in Data Center 1. Because of the PIM dummy packets being sent by NX-2, the L2 switch creates an IGMP snooping mrouter entry for the port. The L2 switch forwards all multicast traffic to NX-2, where its received by the OTV internal interface. The receipt of this traffic creates an OTV mroute entry, as shown in Example 14-54. The delivery group (S, G) is visible with the addition of the detail keyword. The source of the delivery group is the AED’s OTV join interface, and the group address is one of the configured OTV data-groups.

Example 14-54  OTV (V, S, G) MROUTE Detail on NX-2

NX-2# show otv mroute detail

OTV Multicast Routing Table For Overlay0

(103, *, *), metric: 0, uptime: 00:01:02, overlay[r]
  Outgoing interface list: (count: 1)
  Overlay0, uptime: 00:01:02, isis_otv-default

(103, *, 224.0.1.40), metric: 0, uptime: 00:01:02, igmp, overlay[r]
  Outgoing interface list: (count: 2)
  Eth3/5, uptime: 00:01:02, igmp
The OTV mroute is redistributed automatically into IS-IS, as shown in Example 14-55, where the VLAN, site (S,G), delivery (S,G), and LSP-ID are provided.

Example 14-55  OTV MROUTE Redistribution into OTV IS-IS

The redistributed route is advertised to all OTV EDs through IS-IS. Example 14-56 shows the LSP originated by NX-2, as received by NX-6.

Example 14-56  OTV MGROUP Database Detail on NX-6

Note The show otv isis internal event-history mcast command is useful for troubleshooting the IS-IS control plane for OTV multicast and the advertisement of groups and sources for a particular VLAN.
NX-6 updates this information into its OTV mroute table, as shown in Example 14-57. The $s$ indicates the source is located across the overlay.

**Example 14-57  OTV (V, S, G) MROUTE Detail on NX-6**

```
NX-6# show otv mroute detail

OTV Multicast Routing Table For Overlay0

(103, *, *), metric: 0, uptime: 00:00:42, igmp, overlay(r)
  Outgoing interface list: (count: 2)
    Po3, uptime: 00:00:42, igmp
    Overlay0, uptime: 00:00:41, isis_otv-default

(103, *, 224.0.1.40), metric: 0, uptime: 00:00:42, igmp, overlay(r)
  Outgoing interface list: (count: 2)
    Po3, uptime: 00:00:42, igmp
    Overlay0, uptime: 00:00:40, isis_otv-default

(103, *, 239.100.100.100), metric: 0, uptime: 00:00:40, igmp, overlay(r)
  Outgoing interface list: (count: 2)
    Po3, uptime: 00:00:40, igmp
    Overlay0, uptime: 00:00:38, isis_otv-default

(103, 10.103.1.1, 239.100.100.100), metric: 0, uptime: 00:08:58, overlay(s)
  Outgoing interface list: (count: 0)
    Remote Delivery: $s = 10.1.12.1$, $g = 232.1.1.0$
```

The `show otv data-group` command is used to verify the site group and delivery group information for NX-2 and NX-6, as shown in Example 14-58. This should match what is present in the output of `show otv mroute`.

**Example 14-58  Verify Site Group to Delivery Group Mapping**

```
NX-6# show otv data-group

Remote Active Sources for Overlay0

<table>
<thead>
<tr>
<th>VLAN Active-Source</th>
<th>Active-Group</th>
<th>Delivery-Source</th>
<th>Delivery-Group</th>
<th>Joined-I/F</th>
</tr>
</thead>
<tbody>
<tr>
<td>103</td>
<td>10.103.1.1</td>
<td>239.100.100.100</td>
<td>10.1.12.1</td>
<td>232.1.1.0</td>
</tr>
</tbody>
</table>

NX-2# show otv data-group

Local Active Sources for Overlay0

<table>
<thead>
<tr>
<th>VLAN Active-Source</th>
<th>Active-Group</th>
<th>Delivery-Source</th>
<th>Delivery-Group</th>
<th>Join-IF State</th>
</tr>
</thead>
<tbody>
<tr>
<td>103</td>
<td>10.103.1.1</td>
<td>239.100.100.100</td>
<td>10.1.12.1</td>
<td>232.1.1.0</td>
</tr>
</tbody>
</table>
```
OTV EDs act as source hosts and receiver hosts for the delivery groups used on the transport network. An IGMPv3 membership report from the join interface is sent to the transport to allow the OTV ED to start receiving packets from the delivery group (10.1.12.1, 232.1.1.0).

Verification in the transport is done based on the PIM SSM delivery group information obtained from the OTV EDs. Each AED's join interface is a source for the delivery group. The AED joins only delivery group sources that are required based on the OTV mroute table and the information received through the IS-IS control plane. This mechanism allows OTV to optimize the multicast traffic in the transport so that only the needed data is received by each OTV ED. The use of PIM SSM allows specific source addresses to be joined for each delivery group.

Example 14-59 shows the mroute table of a transport router. In this output 10.1.12.1 is NX-2's OTV join interface, which is a source for the delivery group 232.1.1.0/32. The incoming interface should match the routing table path toward the source to pass the Reverse Path Forwarding (RPF) check. Interface Ethernet3/30 is the OIF and is connected to the OTV join interface of NX-6.

Example 14-59  MROUTE Verification in the Transport Network

```
NX-5# show ip mroute 232.1.1.0
IP Multicast Routing Table for VRF "default"
(10.1.12.1/32, 232.1.1.0/32), uptime: 00:02:29, igmp ip pim
Incoming interface: Ethernet3/29, RPF nbr: 10.1.13.1
Outgoing interface list: (count: 1)
  Ethernet3/30, uptime: 00:02:29, igmp
```

Note  Multicast troubleshooting in the transport network between OTV ED sources and receivers follow standard multicast troubleshooting for the delivery group. The fact that OTV has encapsulated the site group within a multicast delivery group does not change the troubleshooting methodology in the transport. The OTV ED are source and receiver hosts for the delivery group from the perspective of the transport network.

OTV Multicast Traffic with a Unicast Transport (Adjacency Server Mode)

Deployments that rely on a unicast transport network can also forward multicast traffic across the overlay for extended VLANs. This is achieved by encapsulating the site group multicast packet into an IP unicast OTV packet across the transport network as depicted in Figure 14-13. If multiple remote sites have interested receivers, the source site OTV
ED must perform head-end replication of the traffic and send a copy to each site, which becomes inefficient at scale.

Figure 14-13  Multicast Traffic Across OTV with Adjacency Server Mode

In this example, Host A and Host C are both members of VLAN 103. Host A is sending traffic to the site group 239.100.100.100, and Host C sends an IGMP membership report message to the Data Center 2 L2 switch. The L2 switch forwards the membership report to NX-6 because it is an mrouter port in IGMP snooping. The same PIM dummy hello packet mechanism is used on the OTV internal interface, just as with a multicast enabled transport. The arrival of the IGMP membership report on NX-6 triggers an OTV mroute to be created, as shown in Example 14-60, with the internal interface Port-channel 3 as an OIF.

Example 14-60  OTV (V, *, G) MROUTE Detail on NX-6

```
NX-6# show otv mroute detail

OTV Multicast Routing Table For Overlay0

(103, *, *), metric: 0, uptime: 00:03:25, igmp, overlay(r)
Outgoing interface list: (count: 2)
Po3, uptime: 00:03:25, igmp
NX-2 uptime: 00:03:24, isis_otv-default

(103, *, 224.0.1.40), metric: 0, uptime: 00:03:25, igmp
Outgoing interface list: (count: 1)
Po3, uptime: 00:03:25, igmp

(103, *, 239.100.100.100), metric: 0, uptime: 00:03:23, igmp
Outgoing interface list: (count: 1)
Po3, uptime: 00:03:23, igmp
```
The OTV mroute is then redistributed automatically into IS-IS for advertisement to all other OTV EDs, as shown in Example 14-61. The LSP ID should be noted so that it can be checked on NX-2, which is the OTV ED for the multicast source Host A in Data Center 1.

**Example 14-61  OTV MROUTE Redistributed into OTV IS-IS on NX-6**

```
NX-6# show otv isis ip redistribute mroute
OTV-IS-IS process: default OTV-IS-IS IPv4 Local Multicast Group database
VLAN 103: (*, *)
  Receiver in LSP_ID: 6c9c.ed4d.d944.00-00
VLAN 103: IPv4 router attached
VLAN 103: (*, 224.0.1.40)
  Receiver in LSP_ID: 6c9c.ed4d.d944.00-00
VLAN 103: IPv4 router attached
VLAN 103: (*, 239.100.100.100)
  Receiver in LSP_ID: 6c9c.ed4d.d944.00-00
VLAN 103: IPv4 router attached
```

**Note** There is a PIM enabled router present on VLAN 103, as indicated in Example 14-61 by the (*, *) entry.

Because IGMP packets are not forwarded across the overlay, the IS-IS messages used to signal an interested receiver are counted as IGMP proxy-reports. Example 14-62 shows the IGMP snooping statistics of NX-6, which indicate the proxy-report being originated through IS-IS. The IGMP proxy-report mechanism is not specific to OTV adjacency server mode.

**Example 14-62  OTV IGMP Proxy Reports**

```
NX-6# show ip igmp snooping statistics vlan 103
Global IGMP snooping statistics: (only non-zero values displayed)
  Packets received: 1422
  Packets flooded: 437
  STP TCN messages rcvd: 21
VLAN 103 IGMP snooping statistics, last reset: never (only non-zero values displayed)
  Packets received: 1350
  IGMPv2 reports received: 897
  IGMPv2 queries received: 443
  IGMPv2 leaves received: 10
  PIM Hellos received: 2598
  IGMPv2 leaves suppressed: 4
  Queries originated: 4
  IGMPv2 proxy-reports originated: 14
```
Following the path from receiver to the source in Data Center 1, the IS-IS database is verified on NX-2. This is done to confirm that the overlay is added as an OIF for the OTV mroute. Example 14-63 contains the GM-LSP received from NX-6 on NX-2.

Example 14-63  OTV IS-IS MGROUP Database Detail on NX-2

<table>
<thead>
<tr>
<th>LSPID</th>
<th>Seq Number</th>
<th>Checksum</th>
<th>Lifetime</th>
<th>A/P/O/T</th>
<th>Instance</th>
<th>Group-Address</th>
<th>IP Multicast</th>
<th>Vlan</th>
<th>Groups</th>
<th>Group</th>
<th>Sources</th>
<th>Router-capability</th>
<th>Interested Vlans</th>
<th>Vlan</th>
<th>Vlan end</th>
</tr>
</thead>
<tbody>
<tr>
<td>6c9c.ed4d.d944.00-00</td>
<td>0x00000005</td>
<td>0x7579</td>
<td>820</td>
<td>0/0/0/1</td>
<td>0x00000003</td>
<td>239.100.100.100</td>
<td>Vlan : 103</td>
<td>Groups : 2</td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td>Vlan Start 103</td>
<td></td>
<td>Vlan end 103</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>224.0.1.40</td>
<td>Vlan : 103</td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td>Vlan Start 103</td>
<td></td>
</tr>
</tbody>
</table>

The IGMP Snooping table on NX-2 confirms that the overlay is included in the port list, as shown in Example 14-64.

Example 14-64  IGMP Snooping OTV Groups on NX-2

<table>
<thead>
<tr>
<th>Vlan Group Address</th>
<th>Ver Type Port list</th>
</tr>
</thead>
<tbody>
<tr>
<td>103 224.0.1.40</td>
<td>v3  D Overlay0</td>
</tr>
<tr>
<td>103 239.100.100.100</td>
<td>v3  D Overlay0</td>
</tr>
</tbody>
</table>

The OTV mroute on NX-2 contains the (V, *, G) entry, which is populated as a result of receiving the IS-IS GM-LSP from NX-6. This message indicates Host C is an interested receiver in Data Center 2 and that NX-2 should add the overlay as an OIF for the group. The OTV mroute table from NX-2 is shown in Example 14-65. The r indicates the receiver is reachable across the overlay. The (V, S, G) entry is also present, which indicates Host A is actively sending traffic to the site group 239.100.100.100.
Example 14-65  OTV MROUTE Detail on NX-2

```
NX-2# show otv mroute detail

OTV Multicast Routing Table For Overlay0

(103, *, *), metric: 0, uptime: 00:12:22, overlay(r)
Outgoing interface list: (count: 1)
   NX-6 uptime: 00:12:21, isis_otv-default

(103, *, 224.0.1.40), metric: 0, uptime: 00:12:21, overlay(r)
Outgoing interface list: (count: 1)
   NX-6 uptime: 00:12:21, isis_otv-default

(103, *, 239.100.100.100), metric: 0, uptime: 00:12:21, overlay(r)
Outgoing interface list: (count: 1)
   NX-6 uptime: 00:12:21, isis_otv-default

(103, 10.103.0.1, 239.100.100.100), metric: 0, uptime: 00:12:21, site
Outgoing interface list: (count: 1)
   NX-6 uptime: 00:10:51, otv
   Local Delivery: s = 0.0.0.0, g = 0.0.0.0
```

**Note**  The OTV mroute table lists an OIF of NX-6 installed by OTV. This is a result of the OTV Unicast encapsulation used in adjacency server mode. The delivery group has values of all zeros for the group address. This information is populated with a valid delivery group when multicast transport is being used.

NX-2 encapsulates the site group packets in an OTV unicast packet with a destination address of NX-6’s join interface. The OTV unicast packets traverse the transport network until they arrive at NX-6. When the packets arrive at NX-6 on the OTV join interface, the outer OTV unicast encapsulation is removed. The next lookup is done on the inner multicast packet, which results in an OIF for the mroute installed by IGMP on the OTV internal interface. Example 14-66 shows the OTV mroute table of NX-6. The site group multicast packet leaves on Po3 toward the L2 switch in Data Center 2 and ultimately reaches Host C.

Example 14-66  OTV MROUTE Detail on NX-6

```
NX-6# show otv mroute detail
show otv mroute detail

OTV Multicast Routing Table For Overlay0
```
With adjacency server mode, the source is not advertised to the other OTV EDs by NX-2. This is because there is no delivery group used across the transport for remote OTV EDs to join. NX-2 only needs to know that there is an interested receiver across the overlay and which OTV ED has the receiver. The join interface of that OTV ED is used as the destination address of the multicast-in-unicast OTV packet across the transport. The actual encapsulation of the site group multicast frame is done using the OTV unicast point-to-point dynamic tunnel, as shown in Example 14-67.

Example 14-67  Dynamic Tunnel Encapsulation for Multicast Traffic

```
xr# show tunnel internal implicit otv detail
tunnel16390 is up
  Admin State: up
  MTU 9178 bytes, BW 9 Kbit
  Tunnel protocol/transport GRE/IP
  Tunnel source 10.1.12.1, destination 10.2.34.1
  Transport protocol is in VRF "default"
  Rx
  663 packets input, 1 minute input rate 0 packets/sec
  Tx
  156405 packets output, 1 minute output rate 0 packets/sec
  Last clearing of "show interface" counters never
```

Advanced OTV Features

Since its initial release as an NX-OS feature, OTV has continued to evolve. The next section in this chapter discusses some of the advanced features of OTV that allow it to be customized to meet the needs of different network deployments.
**First Hop Routing Protocol Localization**

First Hop Routing Protocols (FHRP), such as Hot Standby Routing Protocol (HSRP) and Virtual Router Redundancy Protocol (VRRP), are commonly used to provide a redundant default gateway for hosts on a VLAN. With OTV the VLAN has been extended across the overlay to multiple sites, which means that a router in Data Center 1 could form an HSRP neighbor with a router in Data Center 2. In addition, hosts in Data Center 2 could potentially use a default router that is physically located in Data Center 1, which results in unnecessary traffic crossing the overlay when it could be easily routed locally.

FHRP isolation is configured on the OTV EDs to allow each site’s FHRP to operate independently. The purpose of this configuration is to filter any FHRP protocol traffic, as well as ARP from hosts trying to resolve the virtual IP across the overlay. A configuration example from NX-2 is shown in Example 14-68.

**Example 14-68  FHRP Localization Configuration on NX-2**

```
NX-2# show running-config
! Output omitted for brevity
feature otv

ip access-list ALL_IPs
  10 permit ip any any
ipv6 access-list ALL_IPv6s
  10 permit ipv6 any any
mac access-list ALL_MACs
  10 permit any any
ip access-list HSRP_IP
  10 permit udp any 224.0.0.2/32 eq 1985
  20 permit udp any 224.0.0.102/32 eq 1985
ipv6 access-list HSRP_IPV6
  10 permit udp any ff02::66/128
mac access-list HSRP_VMAC
  10 permit 0000.0c07.ac00 0000.0000.00ff any
  20 permit 0000.0c9f.f000 0000.0000.00ff any
  30 permit 0005.73a0.0000 0000.0000.00ff any
arp access-list HSRP_VMAC_ARP
  10 deny ip any mac 0000.0c07.ac00 ffff.ffff.f000
  20 deny ip any mac 0000.0c9f.f000 ffff.ffff.f000
  30 deny ip any mac 0005.73a0.0000 ffff.ffff.f000
  40 permit ip any mac any
vlan access-map HSRP_Localization 10
  match mac address HSRP_VMAC
  match ip address HSRP_IP
  match ipv6 address HSRP_IPV6
action drop
```
Recall the topology depicted in Figure 14-1. In Data Center 1 HSRP is configured on NX-1 and NX-3 for all VLANs. HSRP is also configured between NX-5 and NX-7 for all VLANs in Data Center 2. The configuration in Example 14-68 is composed of three filtering components:

- VLAN Access Control List (VACL) to filter and drop HSRP Hellos
- ARP Inspection Filter to drop ARP sourced from the HSRP Virtual MAC
- Redistribution Filter Route-Map on the overlay to filter HSRP Virtual MAC (VMAC) from being advertised through OTV IS-IS

FHRP isolation is a common source of problems due to incorrect configuration. Care should be taken to ensure the filtering is properly configured to avoid OTV IS-IS LSP refresh issues as well as duplicate IP address messages or flapping of the HSRP VMAC.

**Multihoming**

A multihomed site in OTV refers to a site where two or more OTV ED are configured to extend the same range of VLANs. Because OTV does not forward STP BPDUs across the overlay, L2 loops form without the election of an AED.
When multiple OTV EDs exist at a site, the AED election runs using the OTV IS-IS system-id and VLAN identifier. This is done by using a hash function where the result is an ordinal value of zero or one. The ordinal value is used to assign the AED role for each extended VLAN to one of the forwarding capable OTV EDs at the site.

When two OTV EDs are present, the device with the lower system-id is the AED for the even-numbered VLANs, and the higher system-id is the AED for the odd-numbered VLANs. The AED is responsible for advertising MAC addresses and forwarding traffic for an extended VLAN across the overlay.

Beginning in NX-OS 5.2(1) the dual site adjacency concept is used. This allows OTV EDs with the same site identifier to communicate across the overlay as well as across the site VLAN, which greatly reduces the chance of one OTV ED being isolated and creating a dual active condition. In addition, the overlay interface of an OTV ED is disabled until a site identifier is configured, which ensures that OTV is able to detect any mismatch in site identifiers. If a device becomes non-AED capable, it proactively notifies the other OTV ED at the site so it can take over the role of AED for all VLANs.

### Ingress Routing Optimization

Egress routing optimization is accomplished with FHRP isolation. Ingress routing optimization is another challenge that needs to be considered in some OTV deployments. OTV allows a VLAN to be extended to multiple sites providing a transparent L2 overlay. This can result in a situation where more than one site is advertising the same L3 prefix to other sites, which may cause suboptimal forwarding.

Figure 14-14 shows that NX-11 has Equal Cost Multipath (ECMP) routes to reach the 10.103.0.0/16 subnet through either NX-9 or NX-10. Depending on the load-sharing hash, packets originating behind NX-11 reach either Data Center 1 or Data Center 2. If for example the destination of the traffic was Host C, and NX11 choose to send the traffic to NX-9 as next-hop, a suboptimal forwarding path is used. NX-9 then has to try to resolve where Host C is located to forward the traffic. The packets reach the internal interface of NX-2, which then performs an OTV encapsulation and routes the packets back across the overlay to reach Host C.

A common solution to this problem is to deploy OTV and Locator-ID Separation Protocol (LISP) together. LISP provides ingress routing optimization by discovering the location of a host and using the LISP control plane to advertise its location behind a specific Routing Locator (RLOC). LISP also provides options for supporting host mobility between sites. If a full LISP deployment is not required, LISP with Interior Gateway Protocol (IGP) assist can be used to redistribute routes from LISP into an IGP protocol.
Another solution to this problem is to advertise more specific, smaller subnets from each site along with the /16 summary to the rest of the routing domain. Routing follows the more specific subnet to Data Center 1 or Data Center 2, and if either partially fails, the /16 summary can still be used to draw in traffic. Assuming OTV is still functional in the partially failed state through a backdoor link, the traffic then relies on the overlay to cross from Data Center 1 to Data Center 2. The best solution to this problem depends on the deployment scenario and if the two OTV sites are acting as Active/Standby or if they are Active/Active from a redundancy perspective.

**Note** For more information on LISP, refer to [http://lisp.cisco.com](http://lisp.cisco.com).

**VLAN Translation**

In some networks, a VLAN configured at an OTV site may need to communicate with a VLAN at another site that is using a different VLAN numbering scheme. There are two solutions to this problem:

- VLAN mapping on the overlay interface
- VLAN mapping on an L2 Trunk port

VLAN mapping on the overlay interface is not supported with Nexus 7000 F3 or M3 series modules. If VLAN mapping is required with F3 or M3 modules, VLAN mapping on the OTV internal interface, which is an L2 trunk, must be used.

Example 14-69 demonstrates the configuration of VLAN mapping on the overlay interface. VLAN 200 is extended across the overlay. The local VLAN 200 is mapped to VLAN 300 at the other OTV site.
Example 14-69  VLAN Mapping on the Overlay Interface

```
NX-2# show running-config interface overlay 0
interface Overlay0
  description Site A
  otv join-interface port-channel13
  otv control-group 239.12.12.12
  otv data-group 232.1.1.0/24
  otv extend-vlan 100-110, 200
  otv vlan mapping 200 to 300
  no shutdown

NX-2# show otv vlan-mapping
Original VLAN -> Translated VLAN
-----------------------------
  200  ->  300
```

If F3 or M3 modules are being used, the VLAN mapping must be performed on the OTV internal interface, as shown in Example 14-70. This configuration translates VLAN 200 to VLAN 300, which is then extended across OTV to interoperate with the remote site VLAN scheme.

Example 14-70  VLAN Mapping on the L2 Trunk

```
NX-2# show running-config interface Ethernet3/5
interface Ethernet3/5
  description 7009A-Main-VDC OTV inside
  switchport
  switchport mode trunk
  switchport vlan mapping 200 300
  mtu 9216
  no shutdown
```

OTV Tunnel Depolarization

L3 routers with multiple ECMP routes to a destination apply a load-sharing hash function to choose an exit interface for a particular flow. A flow is typically the 5-tuple, which consists of the following:

- L3 Source Address
- L3 Destination Address
- Layer 4 Protocol
- Layer 4 Protocol Source Port
- Layer 4 Protocol Destination Port
A problem typical to tunneled traffic is that it may become polarized as it traverses a multihop L3 ECMP network. These flows are referred to as elephants because they are typically moving a lot of traffic and can saturate single links of interface bundles, or of ECMP paths. Tunneled traffic uses a fixed 5-tuple because of the tunnel header and consistent source and destination address. This causes the input to the hash algorithm to stay the same, even though multiple diverse flows could be encapsulated inside the tunnel.

This polarization problem happens when each layer of the transport network applies the same hash function. Using the same inputs results in the same output interface decision at each hop. For example, if a router chose an even-numbered interface, the next router also chooses an even-numbered interface, and the next one also chooses an even-numbered interface, and so on.

OTV provides a solution to this problem. When using the default GRE/IP encapsulation for the overlay, secondary IP addresses can be configured in the same subnet on the OTV join interface, as shown in Example 14-71. This allows OTV to build secondary dynamic tunnels between different pairs of addresses. The secondary address allows the transport network to provide different hash results and load-balance the overlay traffic more effectively.

**Example 14-71  Secondary IP Address to Avoid Polarization**

```
NX-2# show running-config interface port-channel3
interface port-channel3
    description 7009A-Main-OTV Join
    mtu 9216
    no ip redirects
    ip address 10.1.12.1/24
    ip address 10.1.12.4/24 secondary
    ip router ospf 1 area 0.0.0.0
    ip igmp version 3
```

The status of the secondary OTV adjacencies are seen with the `show otv adjacency detail` command, as shown in Example 14-72.

**Example 14-72  OTV Adjacencies with Secondary IP Address**

```
NX-2# show otv adjacency detail
Overlay Adjacency database
Overlay-Interface Overlay0 :
    Hostname    System-ID   Dest Addr   Up Time  State
    NX-4        64a0.e73e.12c2  10.1.22.1  00:03:07 UP
    Secondary src/dest:  10.1.12.4  10.1.22.1  UP
```
Chapter 14: Troubleshooting Overlay Transport Virtualization (OTV)

<table>
<thead>
<tr>
<th>HW-St: Default</th>
<th>NX-8</th>
<th>64a0.e73e.12c4</th>
<th>10.2.43.1</th>
<th>00:03:07 UP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Secondary src/dest:</td>
<td>10.1.12.4</td>
<td>10.2.43.1</td>
<td>UP</td>
<td></td>
</tr>
<tr>
<td>HW-St: Default</td>
<td>NX-6</td>
<td>6c9c.ed4d.d944</td>
<td>10.2.34.1</td>
<td>00:03:06 UP</td>
</tr>
<tr>
<td>Secondary src/dest:</td>
<td>10.1.12.4</td>
<td>10.2.34.1</td>
<td>UP</td>
<td></td>
</tr>
</tbody>
</table>

**Note** OTV tunnel depolarization is enabled by default. It is disabled with the `otv depolarization disable` global configuration command.

When OTV UDP encapsulation is used, the depolarization is applied automatically with no additional configuration required. The Ethernet frames are encapsulated in a UDP packet that uses a variable UDP source port and a UDP destination port of 8472. By having a variable source port, the OTV ED is able to influence the load-sharing hash of the transport network.

**Note** OTV UDP encapsulation is supported starting in NX-OS release 7.2(0)D1(1) for F3 and M3 modules.

**OTV Fast Failure Detection**

OTV’s dual adjacency implementation forms an adjacency on the site VLAN as well as across the overlay for OTV EDs, which have a common site identifier. When an OTV ED becomes unreachable or goes down, the other OTV ED at the site must take over the AED role for all VLANs. Detecting this failure condition quickly minimizes traffic loss during the transition.

The site VLAN IS-IS adjacency can be configured to use Bidirectional Forwarding Detection (BFD) on the site VLAN to detect IS-IS neighbor loss. This is useful to detect any type of connectivity failure on the site VLAN. Example 14-73 shows the configuration required to enable BFD on the site VLAN.

**Example 14-73**  BFD for OTV IS-IS on the Site VLAN

```
NX-2# show otv adjacency detail
! Output omitted for brevity
feature otv
feature bfd
otv site-vlan 10
```
Advanced OTV Features

The status of BFD on the site VLAN is verified with the `show otv isis site` command, as shown in Example 14-74. Any BFD neighbor is also present in the output of the `show bfd neighbors` command.

**Example 14-74  Confirm BFD Neighbor on the Site VLAN**

```
NX-2# show otv isis site

OTV-ISIS site-information for: default

BFD: Enabled [IP: 10.111.111.1]

OTV-ISIS site adjacency local database:

<table>
<thead>
<tr>
<th>SNPA</th>
<th>State</th>
<th>Last Chg</th>
<th>Hold</th>
<th>Fwd-state</th>
<th>Site-ID</th>
<th>Version</th>
<th>BFD</th>
</tr>
</thead>
<tbody>
<tr>
<td>64a0.e73e.12c2</td>
<td>UP</td>
<td>00:00:40</td>
<td>00:01:00</td>
<td>DOWN</td>
<td>0000.0000.0100</td>
<td>3</td>
<td>Enabled [Nbr IP: 10.111.111.2]</td>
</tr>
</tbody>
</table>

OTV-ISIS Site Group Information (as in OTV SDB):


Overlay State Next IIH Int Multi
Overlay1 Up 0.933427 3 20

Overlay Active SG Last CSNP CSNP Int Next CSNP
Overlay1 0.0.2.0 ffff.ffff.ffff.ff-ff 1d14h 00:00:02

Neighbor SystemID: 64a0.e73e.12c2
IPv4 site groups:
0.0.2.0
```
For the overlay adjacency, the presence of a route to reach the peer OTV ED's join interface can be tracked to detect a reachability problem that eventually causes the IS-IS neighbor to go down. Example 14-75 shows the configuration to enable next-hop adjacency tracking for the overlay adjacency of OTV EDs, which use the same site identifier.

**Example 14-75 Configuring OTV Next-Hop Adjacency Tracking**

```
NX-2# show run otv
! Output omitted for brevity
feature otv
otv-isis default
track-adjacency-nexthop
vpn Overlay0
  redistribute filter route-map OTV_HSRP_filter
```

Example 14-76 contains the output of `show otv isis track-adjacency-nexthop`, which verifies the feature is enabled and tracking next-hop reachability of NX-4.

**Example 14-76 Verify OTV Next-Hop Adjacency Tracking**

```
NX-2# show otv isis track-adjacency-nexthop
OTV-ISIS process: default
  OTV-ISIS adj for nexthop: 10.1.12.2, VRF: default
    Hostname: 64a0.e73e.12c2, Overlay: Overlay1
```

This feature depends on a nondefault route, learned from a dynamic routing protocol for the peer OTV ED's join interface. When the route disappears, OTV IS-IS brings down the adjacency without waiting for the hold timer to expire, which allows the other OTV ED to assume the AED role for all VLANs.

**Summary**

OTV was introduced in this chapter as an efficient and flexible way to extend L2 VLANs to multiple sites across a routed transport network. The concepts of MAC routing and the election of an AED were explained as an efficient way to solve the challenges presented by other DCI solutions without relying on STP. The examples and end-to-end walk-through for the control plane, unicast traffic, and multicast traffic provided in this chapter can be used as a basis for troubleshooting the various types of connectivity problems that may be observed in a production network environment.
References


Symbols

* (asterisk) in RegEx, 683
[] (brackets) in RegEx, 680
^ (caret) in RegEx, 679
[^] (caret in brackets) in RegEx, 681
, (comma) utility, 41
$ (dollar sign) in RegEx, 679–680
- (hyphen) in RegEx, 680–681
() (parentheses) in RegEx, 681–682
. (period) in RegEx, 682
| (pipe) in RegEx, 681–682
+ (plus sign) in RegEx, 682
? (question mark) in RegEx, 683
_ (underscore) in RegEx, 677–678
(*, G) join from NX-4 and NX-3 example, 865
ACLs (access control lists), 569–570
ACL Manager, 570–576
for BFD in hardware example, 700–702
BGP network selection, 577
formats example, 571–572
IGP network selection, 576–577
to match traffic on NX-1 example, 810
for permitting BGP traffic example, 613
programming and statistics for DAI example, 346–348
statistics example, 572–573
verifying, 613–615
action-on-failure for on-demand diagnostic tests example, 107
activating maintenance mode with custom profiles example, 730–731
active interfaces, verifying, 402–403
active query in EIGRP, 441–442
Active state, 604
Active/Standby redundancy mode, 29–34
AD (administrative distance), 600

A

access ports, 203–204
accounting log, 45–46, 91
ACL Manager, 570–576
address assignment (IPv6), 357–362
DHCPv6 relay agent, 357–359
DHCPv6 relay LDRA, 360–362
address families, 598–599
adjacency internal forwarding trace example, 162
adjacency manager clients example, 165
adjacency server mode in OTV, 907–912, 932–937
adjacency verification in OTV, 888–898
advanced verification of EIGRP neighbors example, 423
advertising community value example, 685–686
AFI (address-family identifier), 598–599
aggregate-address command, 634–635
allowed VLANs, 206
AM (Adjacency Manager), 160–175
anycast RP, configuring and verifying, 830–841
anycast traffic, 734
architecture of NX-OS, 8–9
feature manager, 14–16
file systems, 19–25
kernel, 9
line card microcode, 17–19
Messages and Transactional Services (MTS), 11–12
multicast architecture, 741–743
CLI commands, 743
CPU protection, 745–747
implementation, 747–750
replication, 744–745
Persistent Storage Services (PSS), 13–14
system manager (sysmgr), 9–11
area settings mismatches
in IS-IS, 539–541
in OSPF, 473–474
areas
in IS-IS, 508–509
in OSPF, 453
ARP (Address Resolution Protocol), 160–175
ACL configuration and verification, 348–349
dynamic ARP inspection (DAI), 345–349
entry for multicast source example, 796
event history example, 163–164
event-history logs and buffer size example, 92
ND-Cache event-history example, 916–917
in OTV, 915–917
synchronization in vPC, 291–292
table example, 162
ARP-ND-Cache, 915–917
ASM (any source multicast), 785–787
configuring, 787–788
event-history and MROUTE state verification, 789–795
platform verification, 795–799
verifying, 788–789
ASN (autonomous system number), 597–598
ASN mismatch, 412–413
AS-Path access lists, 684
assert message (PIM), 778–779
asterisk (*) in RegEx, 683
asynchronous mode in BFD, 691–692
asynchronous mode with echo function in BFD, 693
attach module CLI usage from supervisor example, 18–19
attribute modifications for route-maps, 586
attributes (BGP), 637
authentication
  in EIGRP, 418–419
  in FabricPath, 302
  in IS-IS, 544–546
    on overlay interface, 905–907
  in OSPF, 478–482
automation, 949–950. See also programmability
Open NX-OS, 950–951
shells and scripting, 951
  bash shell, 951–957
  Guest shell, 957–960
  Python, 960–964
AS (autonomous system), 597
autorecovery (vPC), 289
auto-RP
  configuration on NX-3 example, 817–818
  configuring and verifying, 813–820
  event-history on NX-4 example, 819–820
  listener configuration on NX-2 example, 818–819
  mapping agent configuration on NX-4 example, 815–816
B
backup Layer 3 routing in vPC, 292–293
bad BGP updates, 622–623
baseline configuration
EIGRP (Enhanced Interior Gateway Protocol), 399–401
IS-IS (Intermediate System-to-Intermediate System), 518–520
OSPF (Open Shortest Path First), 456–458
bash shell, 51, 951–957
best path calculation in BGP, 636–639
BFD (bidirectional forwarding detection), 689–691, 944–945
asynchronous mode, 691–692
asynchronous mode with echo function, 693
configuring and verifying sessions, 693–707
control packet fields, 691–692
with echo function configuration and verification example, 702–703
event-history logs example, 696–697
failure log example, 703
failure reason codes, 703
feature status example, 695
for OTV IS-IS on site VLAN example, 944–945
over port-channel example, 706–707
over port-channel (micro session configuration) example, 706
over port-channel per-link configuration example, 704–705
session-based event-history example, 697–699
transition history logs example, 699–700
bfd per-link command, 704–705
BGP (Border Gateway Protocol), 597–598
  address families, 598–599
  attributes detail example, 652–653
best path calculation, 636–639
best path selection example, 638–639
configuration and verification, 605–609
convergence, 646–649
event-history example, 674–675
event-history for inbound prefixes example, 666
event-history for outbound prefixes example, 667
filter-lists example, 670, 672–673
flaps due to MSS issue example, 628
and IBP redistribution example, 633–634
IPv6 peer troubleshooting, 621–622
keepalive debugs example, 619
logs collection, 687
loop prevention, 599–600
message sent and OutQ example, 625
messages
KEEPALIVE, 602
NOTIFICATION, 602
OPEN, 601–602
types of, 601
UPDATE, 602
multipath, 640–643
neighbor states, 602–603
Active, 604
Connect, 603–604
Established, 605
Idle, 603
OpenConfirm, 604
OpenSent, 604
network selection, 577
path attributes (PA), 599
peer flapping troubleshooting, 622
bad BGP updates, 622–623
Hold Timer expired, 623–624
Keepalive generation, 624–626
MTU mismatches, 626–630
peering down troubleshooting, 609–610
ACL and firewall verification, 613–615
configuration verification, 610–611
debug logfiles, 618–619
notifications, 619–621
OPEN message errors, 617–618
reachability and packet loss verification, 611–613
TCP session verification, 615–617
policy statistics for prefix-list example, 667–668
policy statistics for route-map example, 675
regex queries
for AS _100 example, 678
for AS _100_ example, 678
with AS 40 example, 680
for AS 100 example, 678
for AS 300 example, 679
with asterisk example, 683
with brackets example, 680
with caret example, 679
with caret in brackets example, 681
with dollar sign example, 680
with hyphen example, 681
with parentheses example, 682
with period example, 682
broadcast optimization in OTV 981

with plus sign example, 682
with question mark example, 683
route advertisement, 631
with aggregation, 634–635
with default-information originate command, 636
with network statement, 631–633
with redistribution, 633–634
route filtering and route policies, 662–663
communities, 684–686
with filter lists, 669–673
looking glass and route servers, 687
AS-Path access lists, 684
with prefix lists, 663–669
regular expressions, 676–683
with route-maps, 673–676
route processing, 630–631
route propagation, 630–631
route refresh capability example, 656
route-map configuration example, 673–674
router ID (RID), 601
scaling, 649–650
maxas-limit command, 662
maximum-prefixes, 659–661
with route reflectors, 657–659
soft reconfiguration inbound versus route refresh, 654–657
with templates, 653–654
tuning memory consumption, 650–653
sessions, 600–601
table for regex queries example, 677
table on NX-2 example, 662–663
table output after prefix-list configuration example, 665
table output with route-map filtering example, 674
table with filter-list applied example, 670–671
template configuration example, 654
update generation process, 643–646
wrong peer AS notification message example, 617
BiDIR (Bidirectional), 799–803
configuring, 803–804
terminology, 800
verifying, 805–811
blocked switch ports
identification, 225–227
modifying location, 229–232
bloggerd, 47
bootstrap message (PIM), 777–778
bootup diagnostics, 98–99
Bourne-Again Shell (Bash), 951–957
BPDU (Bridge Protocol Data Unit), 220
filter, 244–245
guard, 243–244
guard configuration example, 243
brackets ([ ]) in RegEx, 680
BRIB and URIB route installation example, 648
bridge assurance, 250–252
configuration example, 250
engaging example, 251
brief review of MST status example, 237–238
broadcast domains, 198. See also VLANs (virtual LANs)
broadcast optimization in OTV, 877
broadcast traffic
multicast traffic versus, 734–735
in OTV, 917–918
BSR (bootstrap router), configuring and verifying, 820–830
on NX-1 example, 822–823
on NX-2 example, 826–827
on NX-3 example, 825–826
on NX-4 example, 824–825
buffered logging, 88–89

candidate RP advertisement message (PIM), 779
capture filters in Ethanalyzer, 65–67
capturing
debbug in logfile on NX-OS example, 90
LACP packets with Ethanalyzer example, 265
packets. See packet capture
caret (^) in RegEx, 679
caret in brackets ([^]) in RegEx, 681
CD (collision domain), 197–198
cd command, 20
changing
LACP port priority example, 269
MST interface cost example, 240
MST interface priority example, 241
OSPF reference bandwidth on R1 and R2 example, 503
spanning tree protocol system priority example, 228–229
checking
for feature manager errors example, 16
feature manager state for feature example, 15
IS-IS metric configuration example, 555
Cisco and CLI Python libraries on NX-OS example, 961–962
Cisco proprietary request object fields, 969–970
Cisco proprietary response object fields, 971
classic metrics
on all Nexus switches example, 436
versus wide metrics
in EIGRP, 433–439
on NX-1 example, 435
clear bgp command, 654–657
clear ip mroute command, 748
CLI, 39–44
collecting show tech-support to investigate OSPF problem example, 45
comma (,) utility, 41
commands
access port configuration, 203
aggregate-address, 634–635
bash shell, 951–957
bfd per-link, 704–705
clear bgp, 654–657
clear ip mroute, 748
CLI, 39–44
conditional matching options, 583–584
configure maintenance profile, 728–730
debbug bgp keepalives, 618–619
debbug bgp packets, 623
debbug bgp updates, 671–672
debbug ip bgp brib, 643–645
debbug ip bgp update, 643–645
debbug ip eigrp packets, 405–406
debug ip ospf, 464
debug ip pim data-register receive, 790
default-information originate, 636
default-information originate, 636
default-information originate, 636
default-information originate, 636
default-information originate, 636
debug isis, 529–530
default-information originate, 636
default-information originate, 636
default-information originate, 636
default-information originate, 636
default-information originate, 636
default-information originate, 636
default-information originate, 636
feature bfd, 693
feature netflow, 74
feature nxapi, 972
file system commands
dir bootflash: 21
dir logflash: 24
list of, 20
show file logflash: 24–25
Guest shell, 957–960
IGMP snooping configuration parameters, 758–761
install all, 719
install all kickstart, 714–718
maxas-limit, 662
maximum-prefix, 659–661
for multicast traffic, 743
no configure maintenance profile, 728–730
no system mode maintenance, 724–725
python, 50, 960–961
redirection, 39
run bash, 51
show accounting log, 45–46
show bfd neighbors, 694–695, 704–705
show bfd neighbors detail, 702–703
show bgp, 606–607, 638–639
show bgp convergence detail, 648–649
show bgp event-history, 647–648
show bgp event-history detail, 642–643, 646, 665–667, 674–675
show bgp ipv4 unicast policy statistics neighbor, 675
show bgp policy statistics neighbor filter-list, 672
show bgp policy statistics neighbor prefix-list, 667–668
show bgp private attr detail, 652–653
show bgp process, 607–609
show cli list, 42–43
show cli syntax, 43
show clock, 82
show copp diff profile, 188
show cores, 29
show cores vdc-all, 108
show diagnostic bootup level, 99
show diagnostic content module, 101–103
show diagnostic ondemand setting, 106–107
show diagnostic result module, 103–105
show event manager policy internal, 85–86
show event manager system-policy, 84–85
show fabricpath conflict all, 310
show fabricpath isis adjacency, 304–305
show fabricpath isis interface, 303–304
show fabricpath isis topology, 306
show fabricpath isis vlan-range, 305–306
show fabricpath route, 307
show fabricpath switch-id, 303, 315
show fabricpath unicast routes vdc, 308–309
show fex, 126–128
show forwarding distribution ip igmp snooping vlan, 765
show forwarding distribution ip multicast route group, 797
show forwarding internal trace v4-adj-history, 162
show forwarding internal trace v4-pfx-history, 172–173
show forwarding ipv4 adjacency, 162–163
show forwarding ipv4 route, 173–174
show forwarding route, 173–174
show glbp, 386–388
show glbp brief, 386–388
show guestshell detail, 958–959
show hardware, 98
show hardware capacity interface, 113
show hardware flow, 76–77
show hardware internal cpu-mac eobc stats, 118–119
show hardware internal cpu-mac inband counters, 123
show hardware internal cpu-mac inband events, 122–123
show hardware internal cpu-mac inband stats, 119–122
show hardware internal dev-port-map, 797–798
show hardware internal errors, 114, 124
show hardware internal forwarding asic rate-limiter, 184–185
show hardware internal forwarding instance, 309
show hardware internal forwarding rate-limiter usage, 182–184
show hardware internal statistics module pktflow dropped, 116–118
show hardware mac address-table, 764
show hardware rate-limiter, 745–746
show hardware rate-limiters, 181–182
show hsrp brief, 373–374
show hsrp detail, 373–374
show hsrp group detail, 377–378
show incompatibility-all system, 713–714
show interface, 110–112, 193, 194, 203–204
show interface counters errors, 112–113
show interface port-channel, 261–262
show interface trunk, 204–205
show interface vlan 10 private-vlan mapping, 216
show ip access-list, 572–573
show ip adjacency, 165–166
show ip arp, 161–162, 796
show ip arp inspection statistics vlan, 345–346
show ip arp internal event-history, 163–164
show ip arp internal event-history event, 92
show ip dhcp relay, 337–338
show ip dhcp relay statistics, 337–338
show ip dhcp snooping, 342
show ip dhcp snooping binding, 342–343
show ip eigrp, 404
show ip eigrp interface, 402, 415–416
show ip eigrp neighbor detail, 410–411
show ip eigrp topology, 395, 398
show ip eigrp traffic, 405
show ip igmp groups, 845–846
show ip igmp interface, 853–854
show ip igmp interface vlan, 768–769
show ip igmp internal event-history debugs, 769
show ip igmp internal event-history igmp-internal, 769–770
show ip igmp route, 769
show ip igmp snooping groups, 845–846
show ip igmp snooping groups vlan, 764
show ip igmp snooping internal event-history vlan, 766
show ip igmp snooping mrouter, 854–855
show ip igmp snooping otv groups, 935
show ip igmp snooping statistics, 864–865
show ip igmp snooping statistics global, 767
show ip igmp snooping statistics vlan, 767–768, 934–935
show ip igmp snooping vlan, 757, 763–764
show ip interface, 374
show ip mroute, 770–771, 794–795, 892–893, 932
show ip mroute summary, 894
show ip msdp internal event-history route, 837–838
show ip msdp internal event-history tcp, 837–838
show ip msdp peer, 835–836
show ip ospf, 461
show ip ospf event-history, 464–465
show ip ospf interface, 461, 475–476
show ip ospf internal event-history adjacency, 47
show ip ospf internal event-history rib, 169–170
show ip ospf internal event-history txlist urib, 169
show ip ospf neighbors, 458–459
show ip ospf traffic, 463
show ip pim df, 805–806, 809
show ip pim group-range, 829–830
show ip pim interface, 782–783, 852–853
show ip pim internal event-history bidir, 806
show ip pim internal event-history data-header-register, 840–841
show ip pim internal event-history data-register-receive, 790
show ip pim internal event-history hello, 783–784
show ip pim internal event-history join-prune, 792–793, 806–807, 808, 846–847, 858, 865
show ip pim internal event-history null-register, 790, 791, 840–841, 857
show ip pim internal event-history rp, 819–820, 827–828
show ip pim internal event-history vpc, 857, 865–867
show ip pim internal vpc rpf-source, 856–857, 866–867
show ip pim neighbor, 781
show ip pim rp, 814–819, 822–827
show ip pim statistics, 783, 828–829
show ip prefix-list, 580–581
show ip route, 171, 419–421
show ip sla configuration, 324
show ip sla statistics, 323
show ip traffic, 154–156, 611–612
show ip verify source interface, 349–350
show ipv6 dhcp guard policy, 369–370
show ipv6 dhcp relay statistics, 358–359
show ipv6 icmp vaddr, 378–379
show ipv6 interface, 378–379
show ipv6 nd, 355–356
show ipv6 nd raguard policy, 364
show ipv6 neighbor, 354
show ipv6 snooping policies, 369–370
show isis, 525–526
show isis adjacency, 520–523
show isis database, 558–560
show isis event-history, 530–531
show isis interface, 523–525, 526–527
show isis traffic, 528–529
show key chain, 417, 546
show lacp counters, 262–263
show lacp internal info interface, 263–264
show lacp neighbor, 264
show lacp system-identifier, 264
show logging log, 88
show logging logfile, 959
show logging onboard internal kernel, 148
show logging onboard module 10 status, 23
show mac address-table, 198–199
show mac address-table dynamic vlan, 796, 919–920, 923
show mac address-table multicast, 764
show mac address-table vlan, 305–306
show maintenance profile, 727–728
show maintenance timeout, 726
show module, 96–98, 708
show monitor session, 56–57
show ntp peer-status, 82
show ntp statistics, 83
show nxapi-server logs, 973–975
show nxsdk internal event-history, 967
show nxsdk internal service, 965–966
show otv adjacency, 889, 906–907, 910
show otv arp-nd-cache, 916
show otv data-group, 931
show otv internal adjacency, 890
show otv internal event-history arp-nd, 916–917
show otv isis database, 899
show otv isis database detail, 900–902
show otv isis hostname, 899
show otv isis interface overlay, 906
show otv isis internal event-history adjacency, 898
show otv isis internal event-history iih, 896–897
show otv isis internal event-history spf-leaf, 902–903
show otv isis ip redistribute mroute, 930, 934
show otv isis mac redistribute route, 903–904
show otv isis redistribute route, 921–922
show otv isis site, 895–896
show otv isis site statistics, 904–905
show otv isis traffic overlay0, 904, 906
show otv mroute, 928, 929
show otv mroute detail, 929–930, 931, 933
show otv overlay, 888
show otv route, 902, 923
show otv route vlan, 921
show otv site, 889–890, 895, 911–912
show otv vlan, 891–892, 920
show policy-map interface, 114
show policy-map interface control-plane, 189–190
show policy-map system type network-qos, 194–195
show port-channel compatibility-parameters, 272
show port-channel load-balance, 273–274
show port-channel summary, 260–261, 272, 704–705
show port-channel traffic, 273
show processes log pid, 29
show processes log vdc-all, 109–110
show queueing interface, 114
show queueing interface, 193, 194
show routing clients, 167–168
show routing event-history, 647–648
show routing internal event-history msgs, 169–170
show routing ip multicast event-history rib, 770
show routing ip multicast source-tree detail, 868–869
show routing memory statistics, 171
show run aclmgr, 572
show run all | include glean, 161
show run copp all, 186
show run netflow, 76
show run otv, 908–909, 917–918
show run pim, 781
show run sflow, 79
show run vdc, 137
show running-config, 45
show running-config copp, 188–189
show running-config diff, 43–44
show running-config mmode, 730
show running-config sla sender, 324
show sflow, 79–80
show sflow statistics, 80
show snapshots, 725–726
show sockets client detail, 157–158
show sockets connection tcp, 615–616
show sockets connection tcp detail, 157
show sockets internal event-history events, 616–617
show sockets statistics all, 159
show spanning-tree, 225–227, 237–238, 281–282
show spanning-tree inconsistentports, 246, 252
show spanning-tree interface, 227
show spanning-tree mst, 238–239
show spanning-tree mst configuration, 237
show spanning-tree mst interface, 239–240
show spanning-tree root, 222–224, 225
show spanning-tree vlan, 897–898
show system inband queuing statistics, 150
show system internal access-list input entries detail, 190
show system internal access-list input statistics, 340–341, 348–349, 359, 367–368, 700–702
show system internal access-list interface, 339–340, 367–368, 700–702
show system internal access-list interface e4/2 input statistics module 4, 573–574
show system internal aclmgr access-lists policies, 574–575
show system internal aclmgr ppf node, 575–576
show system internal adjmgr client, 164–165
show system internal adjmgr internal event-history events, 167
show system internal bfd event-history, 695–699
show system internal bfd transition-history, 699–700
show system internal copp info, 191–192
show system internal eltm info interface, 195
show system internal ethpm info interface, 175–178, 195
show system internal fabricpath switch-id event-history errors, 310
show system internal feature-mgr feature action, 16
show system internal feature-mgr feature bfd current status, 695
show system internal feature-mgr feature state, 15
show system internal fex info fport, 128–130
show system internal fex info sat port, 128
show system internal flash, 13–14, 24, 88–89
show system internal forwarding adjacency entry, 173–174
show system internal forwarding route, 173–174
show system internal forwarding table, 350
show system internal mmode logfile, 731
show system internal mts buffer summary, 145–146
show system internal mts buffers detail, 146–147
show system internal mts event-history errors, 148
show system internal mts sup sap description, 146–147
show system internal mts sup sap sap-id, 11–12
show system internal mts sup sap stats, 147–148
show system internal pixm info ltl, 765
show system internal pktmgr client, 151–152
show system internal pktmgr interface, 152–153
show system internal pktmgr stats, 153
show system internal port-client event-history port, 179
show system internal port-client
link-event, 178–179
show system internal qos queueing
stats interface, 114–115
show system internal rpm as-path-
access-list, 672–673
show system internal rpm clients,
588–589
show system internal rpm event-
history rsw, 588, 672–673
show system internal rpm ip-prefix-
list, 589, 668–669
show system internal sal info
database vlan, 350
show system internal sflow info, 80
show system internal sup opcodes,
147
show system internal sysmgr
gsync-pending, 32
show system internal sysmgr
service, 10
show system internal sysmgr service
all, 10, 11, 146
show system internal sysmgr service
dependency srvname, 142–143
show system internal sysmgr state,
31–32, 710–711
show system internal ufdm event-
history debugs, 171–172
show system internal vpcm info
interface, 318–320
show system mode, 720–722
show system redundancy ha status,
709
show system redundancy status,
29–30, 708–709
show system reset-reason, 29, 110
show tech adjmgr, 167
show tech arp, 167
show tech bfd, 704

show tech bgp, 687
show tech dhcp, 362
show tech ethpm, 179
show tech glbp, 390
show tech hsrp, 379
show tech netstack, 617, 687
show tech nxapi, 975
show tech nxsdk, 967
show tech routing ipv4 unicast, 687
show tech rpm, 687
show tech track, 334
show tech vpc, 294
show tech vrrp, 385
show tech vrrpv3, 385
show tech-support, 44–45, 320,
749–750
show tech-support detail, 124, 141
show tech-support eem, 87
show tech-support eltm, 195
show tech-support ethpm, 130, 195
show tech-support fabricpath, 310
show tech-support fex, 130
show tech-support ha, 719
show tech-support issu, 719
show tech-support mmode, 731
show tech-support netflow, 78
show tech-support netstack, 160
show tech-support pktmgr, 160
show tech-support sflow, 80
show tech-support vdc, 141
show tunnel internal implicit otv
brief, 890–891
show tunnel internal implicit otv
detail, 922, 937
show tunnel internal implicit otv
tunnel_num, 891
show udld, 247–248
990 commands

show udld internal event-history errors, 248–249
show vdc detail, 137–138
show vdc internal event-history, 140–141
show vdc membership, 139–140
show vdc resource detail, 138–139
show vdc resource template, 131–132
show virtual-service, 959–960
show virtual-service tech-support, 960
show vlan, 201–202, 214
show vlan private-vlan, 210–211
show vpc, 280–281, 284–285, 314–315
show vpc consistency-parameters, 285–286
show vpc consistency-parameters vlan, 286–287
show vpc consistency-parameters vpc, 287
show vpc orphan-ports, 288
show vpc peer-keepalive, 282–283
show vrrp, 380–381
show vrrp statistics, 381–382
show vrrpv3, 383–384
show vrrpv3 statistics, 384–385
soft-reconfiguration inbound, 654–657
source, 963
system maintenance mode always-use-custom-profile, 728–730
system mode maintenance, 720–722
system mode maintenance dont-generate-profile, 730–731
system mode maintenance on-reload reset-reason, 726–727
system mode maintenance timeout, 726

system switchover, 711–712
test packet-tracer, 71–72
communities in BGP, 684–686
community PVLANs, 207, 212–215
comparing before and after maintenance snapshots example, 725–726
complex matching route-maps example, 585
conditional matching, 569
with ACLs, 569–570
ACL Manager, 570–576
BGP network selection, 577
IGP network selection, 576–577
with prefix lists, 580–581
with prefix matching, 578–579
route-maps, 582–584
command options, 583–584
complex matching, 585–586
multiple match conditions, 584–585
configuration checkpoints, 48–49
configuration rollbacks, 48–49
configure maintenance profile command, 728–730
configuring
ARP ACLs, 348–349
ASM (any source multicast), 787–788
AS-path access list, 684
auto-RP configuration on NX-3, 817–818
auto-RP listener configuration on NX-2, 818–819
auto-RP mapping agent configuration on NX-4, 815–816
BFD (bidirectional forwarding detection)
  with echo function, 702–703
  for OSPF example, 694
  over port-channel per-link, 704–705
  sessions, 693–707
BGP (Border Gateway Protocol), 605–609
  route-map, 673–674
  table output after prefix-list, 665
  template, 654
BiDIR (Bidirectional), 803–804
BPDU guard, 243
bridge assurance, 250
BSR (bootstrap router)
  on NX-1, 822–823
  on NX-2, 826–827
  on NX-3, 825–826
  on NX-4, 824–825
console logging example, 88
CoPP NetFlow, 78
custom maintenance profiles example, 728–730
DAI (dynamic ARP inspection), 345–346
DHCP relay, 336–337
DHCP snooping, 342
DHCPv6 guard, 369–370
dynamic ARP inspection, 346
EEM, 85–86
EIGRP (Enhanced Interior Gateway Protocol)
  baseline configuration, 399–401
  with custom K values, 414
  with modified hello timer, 416
  with passive interfaces, 404–405
  stub configuration, 424
error recovery service, 244
ERSPAN, 59
FabricPath, 300–302
FEX (Fabric Extender), 126
FHRP localization configuration on NX-2, 938–939
filtering SPAN traffic, 57
GLBP (Gateway Load-Balancing Protocol), 386
HSRP (Hot Standby Routing Protocol), 372–373
HSRPv6, 377
IP SLA ICMP echo probe, 323
IP SLA TCP connect probe, 328
IP source guard, 350
IPv6 RA guard, 364
IPv6 snooping, 367
IS-IS (Intermediate System-to-Intermediate System)
  baseline configuration, 518–520
  L2 route-leaking, 564–565
  metric transition mode, 555
  with passive interfaces, 528
  routing and topology table after static metric configuration, 552–553
jumbo MTU system, 193
L1 route propagation example, 560
L2 and L3 rate-limiter and exception, 184–185
LACP fast and verifying LACP speed state example, 270
Layer 3 routing over vPC example, 294
loop guard, 246
with maximum hops example, 425
maximum links example, 267
minimum number of port-channel member interfaces example, 265–266
MST (Multiple Spanning-Tree Protocol), 236–237
multicast vPC
  on NX-3, 851–852
  on NX-4, 850–851
NetFlow, 73–77
  flow exporter definition, 75–76
  flow monitor and interface, 76
  flow monitor definition, 76–77
  flow record definition, 74–75
  sampler and interface, 78
NTP, 81–82
NX-1 redistribution, 431, 488, 567
NX-1 to redistribute 172.16.10/24 into OSPF, 489–490
NX-2 redistribution, 587
NX-2's PBR, 592–593
NX-3 anycast RP with MSDP, 832–833
NX-4 anycast RP with MSDP, 834–835
NX-API feature configuration, 972
NX-OS BGP, 606
on-reload reset-reason, 726–727
OSPF (Open Shortest Path First)
  baseline configuration, 456–458
  to ignore interface MTU example, 470
  network types example, 476
  with passive interfaces, 462–463
OTV (Overlay Transport Virtualization), 882–885
  adjacency server on NX-2, 908–909
  ED adjacency server mode on NX-4, 908
  internal interface, 882
  IS-IS authentication example, 905
  join interface, 883
  next-hop adjacency tracking example, 946
  overlay interface, 885
packet tracer, 71–72
PIM (Protocol Independent Multicast)
  anycast RP on NX-4, 840
  ASM on NX-1, 788
  auto-RP candidate-RP on NX-1, 814–815
  BiDIR on NX-1, 803–804
  sparse mode on interface example, 781
  SSM on NX-2, 843–844
  SSM on NX-4, 844–845
  static RP on NX-3, 812
PIM RP, 811–812
  anycast RP, 830–841
  Auto-RP, 813–820
  BSR (bootstrap router), 820–830
  static RP, 812–813
port down upon MAC move notification example, 242–243
port-channels, 259–260
promiscuous PVLAN SVI example, 216
route-maps, 586
sample distribute list configuration, 427
sample MST configuration on NX-1, 236–237
sample offset list configuration, 428
scale factor configuration, 190, 191–192
scheduler job example, 50
sFlow, 79
SPAN (Switched Port Analyzer), 55–56
SPAN-on-drop, 61
SPAN-on-latency, 61
SSM (source specific multicast), 843–845
syslog logging, 90
trunk port, 204
UDLD, 247
unicast RPF, 351–352
URPF (Unicast Reverse Path Forwarding), 351–352
VDC (Virtual Device Contexts), 133–134
virtual link, 484
vPC (virtual port-channel), 278–280
  autorecovery example, 289
  peer-gateway example, 291
vPC+, 311–314
vPC-connected receiver, 861–869
vPC-connected source, 849–861
VRRP (Virtual Router Redundancy Protocol), 380
VRRPv3 migration, 382
confirming
  BFD neighbor on site VLAN example, 945
  IS-IS interfaces, 523–526
  OBFL is enabled on module example, 23
  OSPF interfaces, 460–461
  redundancy and synchronization state example, 31–32
  confusing EIGRP ASN configuration example, 412
  Connect state, 603–604
  consistency checkers, 49–50
  vPC, 283–287
  console logging, 88
  control plane (OTV), 885–886
    adjacency server mode, 907–912
    adjacency verification, 888–898
    authentication, 905–907
    CoPP, 912–913
    IS-IS topology table, 898–905
    multicast mode, 887–888
  convergence in BGP, 646–649
  convergence problems, 439–441
    active query, 441–442
    stuck in active (SIA) queries, 443–446
  CoPP (control plane policing), 179–192
    classes, 745
    NetFlow configuration and verification example, 78
    strict policy on Nexus example, 186–188
  copy command, 20
  core interfaces (FabricPath), verifying, 303–304
  corrupt BGP update message example, 623
  count or wc utility usage example, 40
  count utility, 40
  CPU protection, 745–747
  creating and debugging base shell scripts example, 953–954
CSMA/CD (Carrier Sense Multiple Access/Collision Detect), 197

custom maintenance profiles, 727–731

DAI (dynamic ARP inspection), 345–349
ACL programming, 346–348
ARP ACLs, 348–349
configuring and verifying, 345–346
data plane (OTV)
ARP resolution and ARP-ND-Cache, 915–917
broadcasts, 917–918
encapsulation, 913–915
multicast traffic with multicast enabled transport, 924–932
multicast traffic with unicast transport, 932–937
selective unicast flooding, 918–919
unicast traffic with multicast enabled transport, 919–924
Dead Interval Time, 476–478
debug bgp keepalives command, 618–619
debug bgp packets command, 623
debug bgp updates command, 671–672
debug bgp updates output example, 671–672
debug commands with filter example, 649
debug filters, 47–48
debug ip bgp briib command, 643–645
debug ip bgp update command, 643–645
debug ip eigrp packets command, 405–406
debug ip ospf command, 464
debug ip pim data-register receive command, 790
debug ip pim data-register send command, 790
debug isis command, 529–530
debug log file and debug filter example, 47–48
debug logfiles, 47–48, 90, 618–619
debug mmode logfile command, 731
debug sockets tcp pcb command, 156–157
debugs for BGP update and route installation in BRIB example, 644–645
decimal format, converting to dot-decimal, 473
dedicated OTV broadcast group example, 917–918
default FA in OSPF type-5 LSA example, 490
default-information originate command, 636
delete command, 20
dense mode (DM), 771–772
dependencies in feature manager, 14
deployment models for OTV, 881
deployment of community PVLANs on NX-1 example, 213
deployment of isolated PVLAN on NX-1 example, 209–210
detailed VLAN 115 IGMP snooping group membership example, 764
detecting inconsistent port state example, 251
determining current supervisor redundancy state example, 29–30
determining the SoC instances on module 3 of NX-2 example, 797–798
DF election message (PIM), 779–780
DHCP (Dynamic Host Configuration Protocol)
relay configuration example, 337
snooping ACL programming example, 343–345
snooping binding database example, 343
snooping configuration and validation example, 342
DHCP relay, 335–341
ACL verification, 339–341
configuring, 336–337
verifying, 337–338
DHCP snooping, 341–345
ACL programming, 343–345
binding database, 342–343
configuring, 342
DHCPv6
guard configuration and policy verification example, 369–370
relay ACL line card statistics example, 359
relay statistics example, 358–359
DHCPv6 Guard, 368–370
DHCPv6 relay agent, 357–359
DHCPv6 relay LDRA, 360–362
diagnostic tests. See GOLD (Generic Online Diagnostic) tests
diff utility, 40
different OSPF areas on Ethernet1/1 interfaces example, 472
different OSPF hello timers example, 477
dir bootflash: command, 21
dir command, 20
dir logflash: command, 24
DIS (Designated Intermediate System), 516–517, 543–544
disabling BGP client-to-client reflection example, 658
discontiguous networks in OSPF, 482–485
display filters in Ethanalyzer, 65–67
displaying
active EIGRP interfaces example, 402
EIGRP neighbors example, 401
IS-IS neighbors example, 521
IS-IS neighbors with summary and detail keywords example, 521–522
OSPF neighbors example, 459
distribute list, 426–427
dollar sign ($) in RegEx, 679–680
domains (vPC), 275–276, 280–282
dot-decimal format, converting decimal to, 473
drop threshold for syslog logging example, 190–191
DRs (Designated Routers), 452, 474–476
dummy PIM hello captured in Ethanalyzer example, 926–927
duplicate multicast packets, 870
duplicate router-ID example, 471
duplicate router-ID in OSPF, 485–487
duplicate system-ID example, 539
duplicate System-ID in IS-IS, 546–549
dynamic ARP inspection configuration and verification example, 346
dynamic tunnel encapsulation
for multicast traffic example, 937
for NX-6 example, 922

EBGP (external BGP), 600, 640–643
echo command, 951–952
EEM (Embedded Event Manager), 47, 50, 83–87, 107, 964
configuration and verification example, 85–86
system policy example, 84–85
with TCL script example, 86
egrep utility, 41–42
egress multicast replication, 744–745
EIGRP (Enhanced Interior Gateway Protocol), 393–394
adjacency dropping due to retry limit example, 410
adjacency failure due to holding timer example, 415
configuring
baseline configuration, 399–401
with custom K values example, 414
with modified hello timer example, 416
with passive interfaces example, 404–405
convergence problems, 439–441
active query, 441–442
stuck in active (SIA) queries, 443–446
interface level authentication example, 418
neighbor adjacency troubleshooting, 401–402
ASN mismatch, 412–413
authentication, 416–419
connectivity with primary subnet, 409–412
Hello and hold timers, 414–416
K values mismatch, 413–414
passive interfaces, 403–405
verifying active interfaces, 402–403
verifying EIGRP packets, 405–409
packet debugs example, 406
packet types, 399
path attributes for 10.1.1.0/24 example, 428–429
path metric calculation, 396–398
path selection and missing routes troubleshooting, 419–421
classic metrics versus wide metrics, 433–439
distribute list, 426–427
hop counts, 424–425
interface-based settings, 430
load balancing, 421
offset lists, 427–430
redistribution, 430–432
stub routers, 421–424
process level authentication example, 419
reference topology, 394
route-maps, 587
stub configuration example, 424
terminology, 394
topology for 10.1.1.0/24 network example, 440–441
topology for specific prefix example, 398
topology table, 395–396
traffic counters with SIA queries and replies example, 444–445
traffic statistics example, 405
ELAM (embedded logic analyzer module), 19
email utility, 42
Empty echo, 249
emulated switches
  in FabricPath, 310–311
  verifying, 315
enabling
  authentication on FP ports example, 302
  bash-shell feature and using bash commands example, 952
  BFD feature example, 693
  FabricPath feature example, 301
  FP core ports, FP VLAN, and CE edge ports example, 301
  MAC address lookup mode example, 757
  NetFlow, 74
  vPC ARP synchronization example, 292
encapsulation in OTV data plane, 913–915
encrypted authentication in OSPF, 480–482
entering bash shell example, 51
EOBC status and error counters example, 119
EPLD (electronic programmable logic device), 26
error recovery service configuration and demonstration example, 244
ERSPAN (Encapsulated Remote SPAN), 57–60
configuring, 59
session verification, 59–60
Established state, 605
Ethanalyzer, 63–71
capture and display filters, 65–67
capture example, 68
capture of client connection example, 973
capture of IGMP messages on NX-2 example, 767
GLBP (Gateway Load-Balancing Protocol) and, 388–390
HSRP (Hot Standby Routing Protocol) and, 375–376
for HSRPv6, 379
IPv6 Neighbor Discovery, 354–355
multicast traffic examples, 871
write and read example, 69–70
ethanalyzer local interface command, 65
ethanalyzer local read command, 68
EtherChannels. See port-channels
Ethernet protocol, 197
EthPM (Ethernet Port Manager), 175–179
event history logs, 16, 46–47, 92, 749–750, 789–795
  ARP (Address Resolution Protocol)
    buffer size example, 92
    ND-Cache event-history example, 916–917
  auto-RP on NX-4 example, 819–820
  BFD (bidirectional forwarding detection), 696–697
    session-based event-history example, 697–699
  BGP (Border Gateway Protocol), 674–675
    for inbound prefixes example, 666
    multipath example, 643
for outbound prefixes example, 667
update generation example, 646
BiDIR join-prune
  on NX-1, 808
  on NX-4, 807
BiDIR on NX-4 example, 806
for hello messages example, 784
hello packet visibility from IS-IS, 530–531
IGMP (Internet Group Management Protocol)
  internal events example, 770
  snooping VLAN event-history example, 766
IS-IS (Intermediate System-to-Intermediate System), event-history indicates different areas example, 540
and MROUTE state verification, 789–795, 799
MSDP on NX-4, 837–838
null register on NX-4 example, 841
NX-1 and NX-2 example, 536–537
NX-1 IGMP debugs example, 769
NX-1 IS-IS adjacency with MTU mismatch example, 538
NX-1 OSPF adjacency with MTU mismatch example, 469
NX-2 OTV IS-IS IIH example, 896
NX-4 OTV IS-IS IIH example, 897
OSPF (Open Shortest Path First), with mismatched area flags example, 473
OTV (Overlay Transport Virtualization)
  IS-IS adjacency event-history example, 898
  IS-IS SPF event-history example, 903
for RP from NX-4 with BSR example, 827–828
RPM (Route Policy Manager)
  client for prefix-lists example, 668–669
  viewing, 588
spanning tree protocol, viewing, 234
SSM join-prune
  on NX-2, 847
  on NX-4, 847
UDLD example, 248–249
examining
  accounting log example, 45–46
  interface MTU example, 538
  interface’s MTU example, 470
  MTS queue for SAP example, 12
  NX-2’s L2 detailed LSPDB example, 559–560
exclude utility, 42
executing
  command with multiple arguments example, 41
  consistency checker example, 49
external OSPF path selection for type-1 networks example, 497
external routes
  on NX-2 example, 432
  in OSPF, 495–499

F

FabricPath. See also vPC+
  advantages of, 294–296
  authentication, 302
  configuring, 300–302
  devices, 310
emulated switches, 310–311
packet forwarding, 297–300
terminology, 296–297
topology information example, 306
verifying, 303–310
core interfaces, 303–304
IS-IS adjacency, 304–305
software table in hardware, 308–309
switch-IDs, 303, 310
topologies, 306
in URIB, 307
VLANs (virtual LANs), 305–306
failure detection in OTV, 944–946.
See also BFD (bidirectional forwarding detection)
feature bash-shell command, 951–952
feature bfd command, 693
feature dependency hierarchy, 142–143
feature manager, 14–16
feature netflow command, 74
feature nxapi command, 972
feature sets, installing, 15
FEX (Fabric Extender), 2–3, 124–130
configuring, 126
detail example, 127–128
internal information example, 128–130
jumbo MTU settings, 193–194
verifying, 126–128
FHRP (First-Hop Redundancy Protocol), 370
GLBP (Gateway Load-Balancing Protocol), 385–390
configuring, 386
Ethanalyzer and, 388–390
HSRP (Hot Standby Routing Protocol), 370–379
ARP table population, 375
configuring, 372–373
Ethanalyzer and, 375–376
HSRPv6, 376–379
multicast group, 374
verifying, 373–374
version comparison, 371
localization, 938–939
VRRP (Virtual Router Redundancy Protocol), 380–385
configuring, 380
statistics, 381–382
verifying, 380–381
VRRPv3, 382–385
FHS (First-Hop Security), 362–370
attacks and mitigation techniques, 363
DHCPv6 Guard, 368–370
IPv6 snooping, 365–368
RA Guard, 363–364
file systems, 19–25
commands
dir bootflash: 21
dir logflash: 24
list of, 20
show file logflash: 24–25
flash file system, 21–22
logflash, 23–25
onboard failure logging (OBFL), 22–23
filter lists, 669–673
filtering routes
in BGP, 662–663
AS-Path access lists, 684
communities, 684–686
with filter lists, 669–673
looking glass and route servers, 687
with prefix lists, 663–669
regular expressions, 676–683
with route-maps, 673–676
in OSPF, 487
filtering traffic
Ethanalyzer capture and display filters, 65–67
multicast traffic, 748–749
SPAN (Switched Port Analyzer), 57
firewalls, verifying, 613–615
flapping peer issues. See peer flapping (BGP) troubleshooting
flash file system, 21–22
flow exporter definition, 75–76
flow monitor definition, 76–77
flow record definition, 74–75
FNF (Flexible NetFlow), 72–73
Forward Delay, 220
forwarding addresses in OSPF, 488–494
forwarding loops
BPDU filter, 244–245
BPDU guard, 243–244
detecting and remediating, 241–242
MAC address notifications, 242–243
unidirectional links, 245
bridge assurance, 250–252
loop guard, 245–246
UDLD (unidirectional link detection), 246–250
FSM (Finite State Machine), 602–603

G

GIR (Graceful Insertion and Removal), 719–727

GLBP (Gateway Load-Balancing Protocol), 385–390
configuring, 386
Ethanalyzer and, 388–390
global EIGRP authentication, 418–419
GOLD (Generic Online Diagnostic) tests, 98
bootup diagnostics, 98–99
diagnostic test results example, 103–105
EEM (Embedded Event Manager), 107
runtime diagnostics, 100–107
graceful consistency checkers, 284
graceful convergence (LACP), 270
granular verification of EIGRP packets with ACL example, 409
granular view of MST topology example, 239
Guest shell, 957–960
guest shell details example, 959
gunzip command, 20
gzip command, 20

H

hardware crashes, 108–110
hardware forwarding verification on module 3 example, 799
hardware interface resources and drops example, 113
hardware internal errors example, 124
hardware rate-limiters for glean traffic example, 161, 167
hardware troubleshooting, 95–98
GOLD (Generic Online Diagnostic) tests, 98
bootup diagnostics, 98–99
EEM (Embedded Event Manager), 107
runtime diagnostics, 100–107
health checks, 108
  hardware and process crashes, 108–110
  interface errors and drops, 110–115
  packet loss, 110
  platform-specific drops, 116–124
health checks, 108
  hardware and process crashes, 108–110
  interface errors and drops, 110–115
  packet loss, 110
  platform-specific drops, 116–124
hello message (PIM), 775
Hello packets
  in IS-IS, 513–514
    authentication, 544–546
    visibility, 530–531
  in OSPF, 450–451
    visibility, 465
Hello Time, 220, 476–478
Hello timers
  in EIGRP, 414–416
  in OSPF, 476–478
high availability. See also BFD (bidirectional forwarding detection); FHRP (First-Hop Redundancy Protocol); vPC (virtual port-channel)
custom maintenance profiles, 727–731
GIR (Graceful Insertion and Removal), 719–727
ISSU (in-service software upgrade), 713–719
stateful switchover (SSO), 707–712
VDC policies, 133
high-availability infrastructure, 28–29
  in-service software upgrade (ISSU), 34–35
  supervisor redundancy, 29–34
historical information of FIB route example, 172–173
history
  of Nexus platforms, 1–2
  of NX-OS, 1–2
HM (health-monitoring) diagnostic tests, 100–105
Hold Timer expired, 623–624
hold timers in EIGRP, 414–416
hop counts, 424–425
HSRP (Hot Standby Routing Protocol), 278, 370–379
  ARP table population, 375
  configuring, 372–373
  Ethanalyzer and, 375–376
  multicast group, 374
  verifying, 373–374
  version comparison, 371
HSRPv6, 376–379
  configuration example, 377
  group detail example, 378
  virtual address verification example, 379
HWRL (hardware rate limiters), 179–192, 745–747
hyphen (-) in RegEx, 680–681
IANA (Internet Assigned Numbers Authority), 597
iBGP (internal BGP), 600
  multipath, 640–643
ICMP echo probes, 322–324
id -a command, 951–952
Identifying
active EIGRP interfaces example, 403
EIGRP example AS, 413
if passive IS-IS is configured for a
level example, 526–527
if passive OSPF interfaces are
configured example, 461
matching sequence for specific prefix
pattern example, 580–581
member link for specific network
traffic example, 274
root ports example, 223–224
root ports on NX-4 and NX-5
example, 224–225
Idle state, 603
IEEE 802.1D standards, 219–220
IGMP (Internet Group Management
Protocol). See also vPC (virtual
port-channel)
created MROUTE entry on NX-1
example, 769, 771
event-history of internal events
example, 770
IGMPv1, 750
IGMPv2, 751–752
IGMPv3, 752–756
state on NX-3 example, 863–864
state on NX-4 example, 862–863
verifying, 761–771
IGMP snooping, 756–761
MFDM entry example, 765
OTV groups on NX-2 example, 935
statistics on NX-4 example, 864–865
status for VLAN 115 example,
763–764
VLAN event-history example, 766
IGMPv1, 750
IGMPv2, 751–752
IGMPv3, 752–756, 846
IGP (Interior Gateway Protocol),
576–577
IIH (IS-IS Hello) packets, 513–514,
544–546
in-band management (VDC),
134–136
in-band Netstack KLM statistics
example, 150, 152
include utility, 42
incompatible OSPF timers example,
477
incomplete configuration of
route-maps, 586
indication of EIGRP K values
mismatch example, 414
ingress routing optimization,
940–941
initializing VDC (Virtual Device
Contexts), 134–136
instability in OTV MAC routing table
example, 902
install all command, 719
install all kickstart command,
714–718
installing
custom RPM package example,
965–966
feature sets, 15
NX-SDK, 965
and removing RPM packages from
bash shell example, 955–957
inter-area routes in OSPF, 495
interfaces. See also passive interfaces
EIGRP
authentication, 418
settings, 430
IPv6 services 1003

error counters example, 113
errors and drops, 110–115
FabricPath, verifying, 303–304
IS-IS
  confirming, 523–526
  link costs, 549–553
OSPF
  area number mismatches, 471–473
  confirming, 460–461
  link costs, 500–504
PIM, verifying, 780–785
PktMgr statistics example, 153
port-channels
  consistency, 271–272
  establishment troubleshooting, 272
priority. See port priority
queueing statistics example, 114–115
status
  object tracking for, 330
  reflecting UDLD error example, 248
STP cost, 221–222
internal flash directories example, 88–89
internal interfaces (OTV), configuring, 882
inter-router communication
  in IS-IS, 511
  in OSPF, 450
intra-area routes in OSPF, 494
I/O module MFIB verification on module 3 example, 798
IP SLA (Service Level Agreement), 321–322
  ICMP echo probes, 322–324
  object tracking, 331
  statistics example, 323
  TCP connect probes, 328–329
  UDP echo probes, 324–325
  UDP jitter probes, 325–327
IPFIB process, 171–175
IPSG (IP Source Guard), 349–350
IPv4 services, 335
  DHCP relay, 335–341
    ACL verification, 339–341
    configuring, 336–337
    verifying, 337–338
  DHCP snooping, 341–345
    ACL programming, 343–345
    binding database, 342–343
    configuring, 342
dynamic ARP inspection (DAI), 345–349
  ARP ACLs, 348–349
    configuring and verifying, 345–346
IP Source Guard (IPSG), 349–350
Unicast Reverse Path Forwarding (URPF), 351–352
IPv6 services, 352
  address assignment, 357–362
    DHCPv6 relay agent, 357–359
    DHCPv6 relay LDRA, 360–362
  First-Hop Security (FHS), 362–370
    attacks and mitigation techniques, 363
    DHCPv6 Guard, 368–370
    IPv6 snooping, 365–368
    RA Guard, 363–364
  Neighbor Discovery (ND), 352–356
    Ethanalyzer capture example, 355
    interface information example, 355–356
peer troubleshooting, 621–622
RA guard configuration example, 364
snooping, 365–368
IS-IS (Intermediate System-to-
Intermediate System), 507
databases, 508–509
collection with passive interfaces
debug example, 528
database for area 49.1234 example, 563
database with L2 route leaking
debug example, 565–566
DIS (Designated Intermediate
System), 516–517
event-history indicates different areas
debug example, 540
hello debugs example, 529–530
hierarchy in, 507–508
IIH packets, 513–514
interface verification example, 523–525
inter-router communication, 511
L2 route-leaking configuration
debug example, 564–565
LSPs (link state packets), 515–516
MAC addresses, 512–513
metric transition mode configuration
deployment and verification example, 555
mismatch of interface types example, 543–544
missing routes troubleshooting
duplicate System-ID, 546–549
interface link costs, 549–553
L1 to L2 route propagations, 556–561
metric calculation, 553–556
redistribution, 566–567
suboptimal routing, 562–566
neighbor adjacency troubleshooting
area settings mismatches, 539–541
baseline configuration, 518–520
catching adjacency capabilities, 541–543
confirming interfaces, 523–526
DIS requirements, 543–544
IIH authentication, 544–546
MTU requirements, 537–539
primary interfaces, 526–528
primary subnets, 535–537
unique System-ID, 539
verifying neighbors, 520–523
verifying packets, 528–535
NET addressing, 509–510
OSPF, compared, 508
OTV control plane, 885–886
adjacency server mode, 907–912
adjacency verification, 888–898
authentication, 905–907
CoPP, 912–913
IS-IS topology table, 898–905
multicast mode, 887–888
packet types, 511–512
path selection troubleshooting,
definitions and processing order, 517–518
protocol verification example, 525–526
routing and topology table after
static metric configuration
example, 552–553
TLVs, 512
topology for area 49.1234 example, 563
topology table with mismatched metric types example, 554–555
traffic statistics example, 529
verifying adjacency in FabricPath, 304–305
isolate and shutdown maintenance mode example, 721–722
isolated PVLANs, 207, 208–212
ISSU (in-service software upgrade), 34–35, 713–719

**J**

join interfaces (OTV), configuring, 883
join-prune message (PIM), 776–777
json utility, 42
JSON-RPC request object fields, 968–969
JSON-RPC response object fields, 970–971
jumbo MTU system configuration example, 193

**K**

K values mismatch, 413–414
Keepalive generation, 624–626
KEEPALIVE message, 602
kernel, 9

**L**

L1 adjacency is affected by L1 IIH authentication on NX-1 example, 545
L1 IIH authentication on NX-1 example, 545
L2 and L3 rate-limiter and exception configuration example, 184–185
LACP (link-aggregation control packets), 256–258
advanced configuration options, 265–268
interface establishment troubleshooting, 272
port-channel configuration, 259–260
system priority, 268–271
verifying, 262–265
LACP fast, 269–270
last utility, 40–41
Layer 2 communications
multicast addresses, 738–739
overview, 197–199
troubleshooting flowchart, 253
Layer 2 overlay. See OTV (Overlay Transport Virtualization)
Layer 3 routing
backup routing in vPC, 292–293
multicast addresses, 739–741
over vPC, 293–294
LDRA (Lightweight DHCPv6 Relay Agent), 360–362
license manager, 15
licensing, 28
line card interop limitations, 141–142
line card microcode, 17–19
listing files on standby supervisor example, 22
load balancing, 421
Local Bridge Identifier, 220
locate UUID for service name example, 11
logflash, 23–25
logging, 87–90
  accounting log, 91
  BGP logs collection, 687
  buffered logging, 88–89
  console logging, 88
  debug logfiles, 90
  event history logs. See event history
  levels, 87
  syslog server, 90
long-lived software releases, 26
looking glass servers, 687
loop guard, 245–246
loop prevention
  with BGP, 599–600
  in route reflectors, 658–659
loop-free topologies. See STP
  (Spanning Tree Protocol)
LSAs (link state advertisements), 453–456
LSPs (link state packets), 515–516

M

MAC addresses
  address table example, 316
  in FabricPath, 305–306
  host C example, 919–920
  host C on NX-6 example, 923
  in IS-IS, 512–513
  multicast source example, 796
  for multicast traffic, 738–739
  preventing forwarding loops,
    242–243
  redistribution into OTV IS-IS
    example, 903–904, 921–922
  viewing, 198–199
in vPC+, 315–316
maintenance mode (GIR), 719–724
maintenance mode timeout settings
  example, 726
maintenance profiles, 727–731
maintenance software releases, 25
major software releases, 25
manageability, 950
match route-map command options
  example, 634
Max Age, 220
maxas-limit command, 662
maximum-prefixes in BGP, 659–661
MD5 authentication in OSPF,
  480–482
member interfaces (port-channels),
  consistency, 271–272
member links (vPC), 277
messages
  BGP (Border Gateway Protocol)
    KEEPALIVE, 602
    NOTIFICATION, 602
    OPEN, 601–602
    types of, 601
    UPDATE, 602
  PIM (Protocol Independent
    Multicast)
    assert message, 778–779
    bootstrap message, 777–778
    candidate RP advertisement
      message, 779
    DF election message, 779–780
    hello message, 775
    join-prune message, 776–777
    register message, 775–776
    register-stop message, 776
    types of, 773–774
metric calculation
  for common LAN interface speeds example, 433
  for EIGRP paths, 396–398
  in IS-IS, 553–556
MFDM verification on NX-2 example, 797
minor software releases, 25
mismatched OSPF hello timers example, 478
missing path of only one route example, 426
missing routes troubleshooting
  EIGRP (Enhanced Interior Gateway Protocol), 419–421
    classic metrics versus wide metrics, 433–439
    distribute list, 426–427
    hop counts, 424–425
    interface-based settings, 430
    load balancing, 421
    offset lists, 427–430
    redistribution, 430–432
    stub routers, 421–424
IS-IS (Intermediate System-to-Intermediate System)
  duplicate System-ID, 546–549
  interface link costs, 549–553
  L1 to L2 route propagations, 556–561
  metric calculation, 553–556
  redistribution, 566–567
  suboptimal routing, 562–566
OSPF (Open Shortest Path First)
  discontiguous networks, 482–485
  duplicate router-ID, 485–487
  filtering routes, 487
  forwarding addresses, 488–494
  redistribution, 487–488
mkdir command, 20
modification of spanning tree
  protocol port cost example, 231–232
move command, 20
MRIB creating (*, G) state example, 770
MROUTE entries
  clearing, 748
  from NX-3 and NX-4 after IGMP join example, 860
  from NX-3 and NX-4 after SPT join example, 859
MROUTE state
  on NX-1 after SPT switchover example, 794–795
  on NX-1 with no receivers example, 791
  on NX-2 after SPT switchover example, 794
  on NX-2 with Active Source example, 790
  on NX-4 after SPT switchover example, 794
  on NX-4 with receiver example, 792
MROUTE types, 924
MROUTE verification, 789–795
  on NX-2 example, 795
  in transport network example, 932
MSDP (Multicast Source Discovery Protocol), 831–838
  event-history on NX-4 example, 837–838
  peer status on NX-4 example, 835–836
  SA state and MROUTE status on NX-3 example, 836–837
MST (Multiple Spanning-Tree Protocol), 236
configuring, 236–237
tuning, 240–241
verifying, 237–240

MTS (Messages and Transactional Services), 11–12, 144–148
message stuck in queue example, 146
OBFL logs example, 148
SAP statistics example, 147–148
MTU mismatches, 626–630

MTU requirements
in IS-IS, 537–539
in OSPF, 469–470
MTU settings, 192–195

MTU verification
under ELTM process example, 195
under ethpm process example, 195

multicast enabled transport
multicast traffic with, 924–932
unicast traffic with, 919–924

multicast mode in OTV, 887–888
multicast source tree detail on NX-4 and NX-3 example, 869

multicast traffic, 733–735
Ethanalyzer examples, 871
IGMP. See IGMP (Internet Group Management Protocol)
Layer 2 addresses, 738–739
Layer 3 addresses, 739–741
with multicast enabled transport, 924–932

NX-OS architecture, 741–743
CLI commands, 743
CPU protection, 745–747
implementation, 747–750
replication, 744–745
PIM. See PIM (Protocol Independent Multicast)
terminology, 735–738
with unicast transport, 932–937
vPC (virtual port-channel), 848–849
duplicate packets, 870
receiver configuration and verification, 861–869
reserved VLAN, 870
source configuration and verification, 849–861

multicast vPC
configuring
on NX-3, 851–852
on NX-4, 850–851
IGMP interface on NX-4 example, 853–854
PIM interface on NX-4 example, 852–853
source MROUTE entry on NX-3 and NX-4 example, 855
source registration from NX-3 example, 857
multihoming in OTV, 939–940
multipath (BGP), 640–643
multiple match options example
route-map example, 585
multiple match variables example
route-map example, 584
multiple subnets in VLANs, 203

N
naming conventions for software releases, 25–27
native VLANs, 206
ND (Neighbor Discovery), 352–356
neighbor adjacency troubleshooting

EIGRP (Enhanced Interior Gateway Protocol), 401–402
- ASN mismatch, 412–413
- authentication, 416–419
- connectivity with primary subnet, 409–412
- Hello and hold timers, 414–416
- $K$ values mismatch, 413–414
- passive interfaces, 403–405
- verifying active interfaces, 402–403
- verifying EIGRP packets, 405–409

IS-IS (Intermediate System-to-Intermediate System)
- area settings mismatches, 539–541
- baseline configuration, 518–520
- checking adjacency capabilities, 541–543
- confirming interfaces, 523–526
- DIS requirements, 543–544
- IIH authentication, 544–546
- MTU requirements, 537–539
- passive interfaces, 526–528
- primary subnets, 535–537
- unique System-ID, 539
- verifying neighbors, 520–523
- verifying packets, 528–535

OSPF (Open Shortest Path First)
- area settings mismatches, 473–474
- authentication, 478–482
- baseline configuration, 456–458
- confirming interfaces, 460–461
- connectivity with primary subnet, 468
- DR requirements, 474–476
- interface area number mismatches, 471–473
- MTU requirements, 469–470
- passive interfaces, 461–463
- timers, 476–478
- unique router-ID, 471
- verifying neighbors, 458–460
- verifying packets, 463–467

neighbor states

in BGP, 602–603
- Active, 604
- Connect, 603–604
- Established, 605
- Idle, 603
- OpenConfirm, 604
- OpenSent, 604

in OSPF, 451–452

neighbors (PIM), verifying, 780–785

NET addressing in IS-IS, 509–510

NetFlow, 72–73
- configuring, 73–77
  - flow exporter definition, 75–76
  - flow monitor definition, 76–77
  - flow record definition, 74–75
- sampling, 77–78
- statistics, 77

Netstack, 148–160
- socket accounting example, 159
- socket client details example, 158

network automation, 950

network broadcasts, 198

network communications, Layer 2
- overview, 197–199
- troubleshooting flowchart, 253

network hubs, 198
network QoS policy verification example, 195

network sniffing, 53–57
   Ethanalyzer, 63–71
   packet tracer, 71–72
   SPAN (Switched Port Analyzer), 54–57
      configuring, 55–56
      ERSPAN, 57–60
      filtering traffic, 57
      SPAN-on-Drop, 61–62
      SPAN-on-Latency (SOL), 60–61
      verifying, 56

network statement BGP route advertisement, 631–633

network switches, 198

network types in OSPF, 474

network-admin and dev-ops user role permissions example, 953

next-hop adjacency tracking, 946

Nexus 2000 series, 2–3
Nexus 3000 series, 3–4
Nexus 5000 series, 4
Nexus 6000 series, 4–5
Nexus 7000 series, 5–6
   hardware rate limiters example, 746
   in-band events example, 123
   in-band status example, 120–122
   packet flow drop counters example, 116–118
Nexus 9000 series, 6–7
   in-band status example, 120–122
Nexus core files example, 108
Nexus in-band counters example, 123
Nexus interface details and capabilities example, 111–112

Nexus platforms

history of, 1–2
Nexus 2000 series, 2–3
Nexus 3000 series, 3–4
Nexus 5000 series, 4
Nexus 6000 series, 4–5
Nexus 7000 series, 5–6
Nexus 9000 series, 6–7
Nexus process crash example, 109–110
no configure maintenance profile command, 728–730
no system mode maintenance command, 724–725
no-more utility, 42

normal traffic flow to NX-6’s loopback 0 interface example, 593
NOTIFICATION message, 602
notifications in BGP, 619–621
NTP (Network Time Protocol), 81–83
   configuring, 81–82
   statistics, 83

NX-1 and NX-2 detect bad subnet mask example, 468
NX-1 and NX-2 event-history example, 536–537
NX-1 and NX-2 routing table for adjacency example, 412
NX-1 and NX-3’s routing table example, 564
NX-1 configuration to redistribute 172.16.10.0/24 into OSPF example, 489–490
NX-1 detects NX-2 as neighbor example, 410
NX-1 does not detect NX-2 example, 537
NX-1 external OSPF path selection for type-2 network example, 498–499
NX-1 IGMP debugs event-history example, 769
NX-1 IGMP interface VLAN 115 state example, 768–769
NX-1 IS-IS adjacency event-history with MTU mismatch example, 538
NX-1 OSPF adjacency event-history with MTU mismatch example, 469
NX-1 redistribution configuration example, 431, 488, 567
NX-1 stuck in INIT state with NX-2 example, 535
NX-1's routing table example, 420
NX-1's routing table with missing NX-4's 10.4.4.0/24 network example, 547
NX-1's routing table with missing NX-4's loopback interface example, 485–486
NX-1's spanning tree protocol information example, 226
NX-2 and NX-4's routing table after L1 route propagation example, 561
NX-2 OTV IS-IS IIH event-history example, 896
NX-2 redistribution configuration example, 587
NX-2 VLAN 115 IGMP snooping statistics example, 767–768
NX-2's LSPDB example, 558
NX-2's PBR configuration example, 592–593
NX-3 anycast RP with MSDP configuration example, 832–833
NX-3 external OSPF path selection for type-2 network example, 499
NX-3's LSP after enabling route propagation example, 561
NX-4 anycast RP with MSDP configuration example, 834–835
NX-4 OTV IS-IS IIH event-history example, 897
NX-6 detected as MROUTER port by IGMP snooping example, 928
NX-API, 968–975
Cisco proprietary request object fields, 969–970
Cisco proprietary response object fields, 971
feature configuration example, 972
JSON-RPC request object fields, 968–969
JSON-RPC response object fields, 970–971
server logs example, 973–975
NX-OS
architecture of, 8–9
  feature manager, 14–16
  file systems, 19–25
  kernel, 9
  line card microcode, 17–19
  Messages and Transactional Services (MTS), 11–12
  Persistent Storage Services (PSS), 13–14
  system manager (sysmgr), 9–11
BGP (Border Gateway Protocol)
  configuration example, 606
  peering verification example, 607
  process example, 608–609
  table output example, 607
component logging level example, 89
detection of forwarding loop example, 242
high-availability infrastructure, 28–29
in-service software upgrade (ISSU), 34–35
supervisor redundancy, 29–34
virtualization

Virtual Device Contexts (VDCs), 35–37
virtual port channels (vPC), 37–39
Virtual Routing and Forwarding (VRF), 37

NX-SDK, 964–967
event history example, 967

O

OBFL (onboard failure logging), 22–23
object tracking, 329
  for interface status, 330
  for route status, 330–331
  with static routes, 334
  for track-list state, 332–333
offline diagnostics, 107
offset list configuration example, 428
offset lists, 427–430
on-demand diagnostics, 105–107
on-reload reset-reason configuration and verification example, 726–727
OPEN message, 601–602, 617–618
Open NX-OS, 950–951
OpenConfirm state, 604
OpenSent state, 604
ORIB entry for host C on NX-6 example, 923
orphan ports (vPC), 288
OSPF (Open Shortest Path First), 449
  adjacency failure example, 475
  areas, 453
  configuration with passive interfaces example, 462–463
  Designated Routers (DRs), 452
encrypted authentication example, 480–481
event-history with mismatched area flags example, 473
hello and packet debugs example, 464
Hello packets, 450–451
interface output example, 461
interface output in brief format example, 460
inter-router communication, 450
IS-IS, compared, 508
LSAs (link state advertisements), 453–456
missing routes troubleshooting
  discontiguous networks, 482–485
duplicate router-ID, 485–487
filtering routes, 487
forwarding addresses, 488–494
redistribution, 487–488
neighbor adjacency troubleshooting
  area settings mismatches, 473–474
  authentication, 478–482
  baseline configuration, 456–458
  confirming interfaces, 460–461
  connectivity with primary subnet, 468
  DR requirements, 474–476
  interface area number mismatches, 471–473
  MTU requirements, 469–470
  passive interfaces, 461–463
timers, 476–478
unique router-ID, 471

verifying neighbors, 458–460
verifying packets, 463–467
neighbor states, 451–452
neighbors stuck in EXSTART
neighbor state example, 469
network types, 474
path selection troubleshooting, 494
  external routes, 495–499
  inter-area routes, 495
  interface link costs, 500–504
  intermixed RFC 1583 and RFC 2328 devices, 499–500
  intra-area routes, 494
plaintext authentication example, 479
route distribution to URIB example, 169
routing table example, 456
traffic statistics example, 463

OTV (Overlay Transport Virtualization), 875–877
(V, *, G) MROUTE detail on NX-6 example, 933
(V, S, G) MROUTE detail on NX-2 example, 929–930
(V, S, G) MROUTE detail on NX-6 example, 931
adjacencies with secondary IP address example, 943–944
adjacency server configuration on NX-2 example, 908–909
adjacency server mode dual adjacency example, 911–912
adjacency server mode IS-IS neighbors example, 910
advanced features
  fast failure detection, 944–946
  FHRP localization, 938–939
  ingress routing optimization, 940–941
multihoming, 939–940
multihoming, 942–944
VLAN mapping, 941–942
configuring, 882–885
control plane, 885–886
adjacency server mode, 907–912
adjacency verification, 888–898
authentication, 905–907
CoPP, 912–913
IS-IS topology table, 898–905
multicast mode, 887–888
data plane
ARP resolution and ARP-ND-Cache, 915–917
broadcasts, 917–918
encapsulation, 913–915
multicast traffic with multicast enabled transport, 924–932
multicast traffic with unicast transport, 932–937
selective unicast flooding, 918–919
unicast traffic with multicast enabled transport, 919–924
deployment models, 881
dynamic unicast tunnels example, 891
ED adjacency server mode configuration on NX-4 example, 908
flood control and broadcast optimization, 877
IGMP proxy reports example, 934–935
internal interface configuration example, 882
IS-IS (Intermediate System-to-Intermediate System)
adjacencies on overlay example, 889
adjacency event-history example, 898
authentication error statistics example, 906
authentication parameters example, 906
database detail example, 900–901
database example, 899
dynamic hostname example, 899
LSP updating frequently example, 901–902
MGROUP database detail on NX-2 example, 935
MGROUP database on NX-2 example, 928–929
overlay traffic statistics example, 904
site adjacency example, 889–890
site-VLAN statistics example, 904–905
SPF event-history example, 903
join interface configuration example, 883
MGROUP database detail on NX-6 example, 930
MROUTE
detail on NX-2 example, 936
detail on NX-6 example, 936–937
entry on NX-2 example, 929
redistributed into IS-IS on NX-6 example, 934
redistribution into OTV IS-IS example, 930
state on NX-6 example, 928
overlay interface configuration  
example, 885
overlay IS-IS adjacency down  
exmaple, 907
partial adjacency example, 895
routing table with selective unicast  
flooding example, 918–919
site VLAN, 882
SSM data-groups example, 925
supported platforms, 878
terminology, 878–880

out-of-band management (VDC),  
134–136
output of RR reflected prefix  
exmaple, 659
overlay interfaces (OTV)  
configuring, 885
IS-IS authentication on, 905–907
verifying, 888–898

PA (path attributes), 599
packet capture, 53–57
Ethanalyzer, 63–71
packet tracer, 71–72
SPAN (Switched Port Analyzer),  
54–57
  configuring, 55–56
ERSPAN, 57–60
filtering traffic, 57
SPAN-on-Drop, 61–62
SPAN-on-Latency (SOL), 60–61
verifying, 56

packet loss  
reasons for, 110
interface errors and drops,  
110–115

platform-specific drops,  
116–124
verifying, 611–613
Packet Manager (PktMgr), 148–160
packet processing filter (PPF),  
575–576
packet tracer, 71–72
packets. See also messages
EIGRP (Enhanced Interior Gateway  
Protocol)
  types of, 399
  verifying, 405–409
FabricPath, 297–300
IS-IS (Intermediate System-to-  
Intermediate System)
  IIH, 513–514, 544–546
  LSPs, 515–516
  types of, 511–512
  verifying, 528–535
LACP. See LACP (link-aggregation  
control packets)
OSPF (Open Shortest Path First)  
  types of, 450
  verifying, 463–467
parentheses ( ) in RegEx, 681–682
partial configuration of route-maps,  
586
passive interfaces  
in EIGRP, 403–405
in IS-IS, 526–528
in OSPF, 461–463
path changed for 10.1.0.24 route  
exmaple, 427
path check after L2 route leaking  
exmaple, 566
path metric calculation in EIGRP,  
396–398
path modification on NX-6 example, 429–430
path selection troubleshooting
EIGRP (Enhanced Interior Gateway Protocol), 419–421
classic metrics versus wide metrics, 433–439
distribute list, 426–427
hop counts, 424–425
interface-based settings, 430
load balancing, 421
offset lists, 427–430
redistribution, 430–432
stub routers, 421–424
IS-IS (Intermediate System-to-Intermediate System), 517–518
OSPF (Open Shortest Path First), 494
external routes, 495–499
inter-area routes, 495
interface link costs, 500–504
intermixed RFC 1583 and RFC 2328 devices, 499–500
intra-area routes, 494
Path-MTU-Discovery (PMTUD), 626–627
PBR (policy-based routing), 591–594
peer flapping (BGP) troubleshooting, 622
bad BGP updates, 622–623
Hold Timer expired, 623–624
Keepalive generation, 624–626
MTU mismatches, 626–630
peer link (vPC), 277
peer-gateway (vPC), 289–291
peering down (BGP) troubleshooting, 609–610
ACL and firewall verification, 613–615
configuration verification, 610–611
debug logfiles, 618–619
notifications, 619–621
OPEN message errors, 617–618
reachability and packet loss verification, 611–613
TCP session verification, 615–617
peer-keepalive link (vPC), 276–277, 282–283
period (.) in RegEx, 682
Persistent Storage Services (PSS), 13–14
pillars of NX-OS, 1–2, 8
PIM (Protocol Independent Multicast), 771–772
(S, G) join events and MROUTE state example, 868
anycast RP configuration on NX-4 example, 840
ASM (any source multicast), 785–787
configuring, 787–788
event-history and MROUTE state verification, 789–795
platform verification, 795–799
verifying, 788–789
auto-RP candidate-RP configuration on NX-1 example, 814–815
BiDIR, 799–803
configuring, 803–804
DF status on NX-4 example, 805–806
event-history on NX-4 example, 806
interface counters on NX-4 example, 807–808
join-prune event-history on NX-1 example, 808
join-prune event-history on NX-4 example, 807
MROUTE entry on NX-1 example, 809
MROUTE entry on NX-2 example, 811
MROUTE entry on NX-4 example, 805
terminology, 800
verifying, 805–811
DF status on NX-1 example, 809
Ethanalyzer capture of PIM hello message example, 784–785
event-history for hello messages example, 784
event-history for RP from NX-4 with BSR example, 827–828
global statistics example, 783
group-to-RP mapping information from NX-2 example, 830
interface and neighbor verification, 780–785
interface parameters on NX-1 example, 782–783
join received from NX-1 on NX-2 example, 793
join sent from NX-1 to NX-2 example, 793
message types
assert message, 778–779
bootstrap message, 777–778
candidate RP advertisement message, 779
DF election message, 779–780
hello message, 775
join-prune message, 776–777
list of, 773–774
register message, 775–776
register-stop message, 776
neighbors on NX-1 example, 781
null register event-history on NX-4 example, 841
RP configuration, 811–812
anycast RP, 830–841
Auto-RP, 813–820
BSR (bootstrap router), 820–830
static RP, 812–813
RPT join from NX-4 to NX-1 example, 792
RPT join received on NX-1 example, 792
SPT joins from NX-2 for vPC-connected sources example, 858
SSM (source specific multicast), 841–843
configuring, 843–845
verifying, 845–848
static RP on NX-3 configuration example example, 812
statistics on NX-4 with BSR example, 828–829
trees, 772–773
vPC (virtual port-channel)
forwarder election on NX-3 and NX-4 example, 866–867
RPF-source cache table on NX-3 and NX-4 example, 856–857
status on NX-4 example, 867
ping test and show ip traffic command output example, 612
ping with DF-bit set example, 629
ping with source interface as loopback example, 611
pipe (|) in RegEx, 681–682
PktMgr (Packet Manager), 148–160
plaintext authentication in OSPF, 478–480
platform FIB verification example, 173–174, 176–178
platform-specific drops, 116–124
plus sign (+) in RegEx, 682
PMTUD (Path-MTU-Discovery), 626–627
port priority
LACP, 268–269
modifying, 232–233
port-channels, 255–258. See also vPC (virtual port-channel)
advanced LACP options, 265–268
advantages of, 255–256
configuring, 259–260
LACP in, 256–258
interface establishment
troubleshooting, 272
system priority, 268–271
verifying packets, 262–265
member interface consistency, 271–272
traffic load-balancing
troubleshooting, 272–274
verifying status, 260–262
Port-Client, 175–179
portfast, 232–235
PPF (packet processing filter), 575–576
prefix advertisement using network command example, 632–633
prefix lists, 580–581, 663–669
prefix matching, 578–579
prefix-list-based route filtering example, 664
primary subnets
EIGRP connectivity, 409–412
IS-IS connectivity, 535–537
OSPF connectivity, 468
process crashes, 108–110
programmability, 950. See also automation; shells and scripting
NX-API, 968–975
NX-SDK, 964–967
Open NX-OS, 950–951
promiscuous PVLANs, 207
community PVLANs and, 212–215
isolated PVLANs and, 208–212
on SVI, 215–217
PSS (Persistent Storage Services), 13–14
PVLANs (private VLANs), 207–208
communication capability between hosts, 208
community PVLANs, 212–215
isolated PVLANs, 208–212
promiscuous PVLANs on SVI, 215–217
trunking between switches, 217–218
PVST (Per-VLAN Spanning Tree), 220
PVST+ (Per-VLAN Spanning Tree Plus), 220
pwd command, 20, 951–952
Python, 960–964
with EEM example, 87
interpreter from CLI and guest shell example, 961
invoking from EEM applet example, 964
printing all interfaces in UP state example, 963–964
python command, 50, 960–961
python interpreter, 50
Q
query modifiers. See RegEx (regular expressions)
question mark (?) in RegEx, 683
queue names (MTS), 146

R
R1 routing table with GRE tunnel example, 139–140
R1’s and NX-2’s IS-IS routing table entries example, 554
R1’s and NX-3’s IS-IS topology table with default metric example, 551
R1’s routing table with 1 gigabit link shutdown example, 502
R1’s routing table with default interface metrics bandwidth example, 550
R1’s routing table with default OSPF auto-cost bandwidth example, 502
RA Guard, 363–364
rate-limiter usage example, 183–184
reachability, verifying, 611–613
redirection, 39
redistribution
  in BGP, 633–634
  in EIGRP, 430–432
  in IS-IS, 566–567
  in OSPF, 487–488
redundancy switchover example, 711–712
RegEx (regular expressions), 676–683
  asterisk (*), 683
  brackets ([]), 680
  caret (^), 679
caret in brackets ([^]), 681
dollar sign ($), 679–680
hyphen (-), 680–681
list of, 676
parentheses (), 681–682
period (.), 682
pipe (|), 681–682
plus sign (+), 682
question mark (?), 683
underscore (_), 677–678
register message (PIM), 775–776, 790
register-stop message (PIM), 776, 791
replication, 744–745
reserved VLAN, 870
resolved and unresolved adjacencies example, 165–166
resource templates (VDC), 131–132
restoring connectivity by allowing BPDUs to process example, 252
reviewing OSPF adjacency event history example, 47
RFC 1583 devices, 499–500
RFC 2328 devices, 499–500
RID (router ID)
  in BGP, 601
  in OSPF, 471, 485–487
rmdir command, 20
Root Bridge Identifier, 220
root bridges, 219
election, 222–224
placement, 228–229
root guard, 229
Root Path Cost, 220
root ports
  identification, 224–225
  modifying location, 229–232
route advertisement in BGP, 631
  with aggregation, 634–635
  with default-information originate command, 636
  with network statement, 631–633
  with redistribution, 633–634
route aggregation in BGP, 634–635
route filtering
  in BGP, 662–663
    communities, 684–686
    with filter lists, 669–673
    looking glass and route servers, 687
    AS-Path access lists, 684
    with prefix lists, 663–669
    regular expressions, 676–683
    with route-maps, 673–676
in OSPF, 487
route leaking in IS-IS, 564–566
route policies in BGP, 662–663
  communities, 684–686
  with filter lists, 669–673
  looking glass and route servers, 687
  AS-Path access lists, 684
  with prefix lists, 663–669
  regular expressions, 676–683
  with route-maps, 673–676
route processing in BGP, 630–631
route propagation in BGP, 630–631
route reflectors in BGP, 657–659
route refresh in BGP, 654–657
route servers, 687
route status, object tracking for, 330–331
route-maps
  attribute modifications (set actions), 586
  in BGP, 673–676
  conditional matching, 582–584
    command options, 583–584
    complex matching, 585–586
    multiple match conditions, 584–585
  explained, 581–582
  partial configuration, 586
  PBR (policy-based routing), 591–594
  RPM (Route Policy Manager), 586–590
  routing loop because of intermixed OSPF devices example, 500
  routing protocol and URIB updates example, 170
  routing protocol states during maintenance mode example, 722–724
routing tables
  with impact example, 422
  of NX-1, NX-2, NX-3, and NX-4 example, 557
  of NX-1 and NX-6 example, 424–425
  of NX-2 and NX-4 example, 486, 548
RP configuration (PIM), 811–812
  anycast RP, 830–841
  Auto-RP, 813–820
  BSR (bootstrap router), 820–830
  static RP, 812–813
RPM (Route Policy Manager), 586–590, 668–669
RSTP (Rapid Spanning Tree Protocol), 220–221
  blocked switch port identification, 225–227
  interface STP cost, 221–222
  root bridge election, 222–224
root port identification, 224–225
tuning, 228–235
  port priority, 232–233
root bridge placement, 228–229
root guard, 229
root port and blocked switch
port locations, 229–232
topology changes and portfast,
232–235
verifying VLANs on trunk links, 227
run bash command, 51, 951–952
runtime diagnostics, 100–107

S
SAFI (subsequent address-family
identifier), 598–599
SAL database info and FIB
verification for IPSG example, 350
sampling
  with NetFlow, 77–78
  with sFlow, 78–80
SAP (service access points), 11, 147
scale factor configuration example,
190, 191–192
scaling BGP (Border Gateway
Protocol), 649–650
maxas-limit command, 662
maximum-prefixes, 659–661
with route reflectors, 657–659
soft reconfiguration inbound versus
route refresh, 654–657
with templates, 653–654
  tuning memory consumption, 650–653
scheduler, 50
scripting. See shells and scripting
secondary IP address to avoid
polarization example, 943
section utility, 42
selective unicast flooding, 918–919
sessions (BGP), 600–601
set actions for route-maps, 586
setting static IS-IS metric on R1 and
R2 example, 552
sFlow, 78–80
  configuring, 79
  statistics, 80
shells and scripting, 951
  bash shell, 951–957
  Guest shell, 957–960
  Python, 960–964
short-lived software releases, 26
show accounting log command,
45–46
show bash-shell command, 951–952
show bfd neighbors command,
694–695, 704–705
show bfd neighbors detail command,
702–703
show bgp command, 606–607,
638–639
show bgp convergence detail
command, 648–649
show bgp convergence detail
command output example,
648–649
show bgp event-history command,
647–648
show bgp event-history detail
command, 642–643, 646,
665–667, 674–675
show bgp ipv4 unicast policy
statistics neighbor command, 675
show bgp policy statistics neighbor
filter-list command, 672
show bgp policy statistics neighbor
prefix-list command, 667–668
show bgp private attr detail command, 652–653
show bgp process command, 607–609
show cli list command, 42–43
show cli list command example, 42–43
show cli syntax command, 43
show cli syntax command example, 43
show clock command, 82
show command output redirection example, 40
show copp diff profile command, 188
show cores command, 29
show cores vdc-all command, 108
show diagnostic bootup level command, 99
show diagnostic content module command, 101–103
show diagnostic content module command output example, 102–103
show diagnostic ondemand setting command, 106–107
show diagnostic result module command, 103–105
show event manager policy internal command, 85–86
show event manager system-policy command, 84–85
show fabricpath conflict all command, 310
show fabricpath isis adjacency command, 304–305
show fabricpath isis interface command, 303–304
show fabricpath isis topology command, 306
show fabricpath isis vlan-range command, 305–306
show fabricpath route command, 307
show fabricpath switch-id command, 303, 315
show fabricpath switch-id command output example, 303
show fabricpath unicast routes vdc command, 308–309
show fex command, 126–128
show file command, 20
show file logflash: command, 24–25
show forwarding distribution ip igmp snooping vlan command, 765
show forwarding distribution ip multicast route group command, 797
show forwarding internal trace v4-adj-history command, 162
show forwarding internal trace v4-pfx-history command, 172–173
show forwarding ipv4 adjacency command, 162–163
show forwarding ipv4 route command, 173–174
show forwarding route command, 173–174
show glbp and show glbp brief command output example, 387–388
show glbp brief command, 386–388
show glbp command, 386–388
show guestshell detail command, 958–959
show hardware capacity interface command, 113
show hardware command, 98
show hardware flow command, 76–77
show hardware internal cpu-mac eobc stats command, 118–119
show hardware internal cpu-mac inband counters command, 123
show hardware internal cpu-mac inband events command, 122–123
show hardware internal cpu-mac inband stats command, 119–122
show hardware internal dev-port-map command, 797–798
show hardware internal errors command, 114, 124
show hardware internal forwarding asic rate-limiter command, 184–185
show hardware internal forwarding instance command, 309
show hardware internal forwarding rate-limiter usage command, 182–184
show hardware internal statistics module pktflow dropped command, 116–118
show hardware mac address-table command, 764
show hardware rate-limiter command, 745–746
show hardware rate-limiters command, 181–182
show hsrp brief command, 373–374
show hsrp detail command, 373–374
show hsrp group detail command, 377–378
show incompatibility-all system command, 713–714
show interface command, 110–112, 193, 194, 203–204
show interface counters errors command, 112–113
show interface port-channel command, 261–262
show interface trunk command, 204–205
show interface trunk command output example, 205
show interface vlan 10 private-vlan mapping command, 216
show ip access-list command, 572–573
show ip adjacency command, 165–166
show ip arp command, 161–162, 796
show ip arp inspection statistics vlan command, 345–346
show ip arp internal event-history command, 163–164
show ip arp internal event-history event command, 92
show ip dhcp relay command, 337–338
show ip dhcp relay statistics command, 337–338
show ip dhcp snooping binding command, 342–343
show ip dhcp snooping command, 342
show ip eigrp command, 404
show ip eigrp interface command, 402, 415–416
show ip eigrp neighbor detail command, 410–411
show ip eigrp topology command, 395, 398
show ip eigrp traffic command, 405
show ip igmp groups command, 845–846
show ip igmp interface command, 853–854
show ip igmp interface vlan command, 768–769
show ip igmp internal event-history debugs command, 769
show ip igmp internal event-history igmp-internal command, 769–770
show ip igmp route command, 769
<table>
<thead>
<tr>
<th>Command</th>
<th>Page Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>show ip igmp snooping groups command</td>
<td>845–846</td>
</tr>
<tr>
<td>show ip igmp snooping groups vlan command</td>
<td>764</td>
</tr>
<tr>
<td>show ip igmp snooping internal event-history vlan command</td>
<td>766</td>
</tr>
<tr>
<td>show ip igmp snooping mrouter command</td>
<td>854–855</td>
</tr>
<tr>
<td>show ip igmp snooping otv groups command</td>
<td>935</td>
</tr>
<tr>
<td>show ip igmp snooping statistics command</td>
<td>864–865</td>
</tr>
<tr>
<td>show ip igmp snooping statistics global command</td>
<td>767</td>
</tr>
<tr>
<td>show ip igmp snooping statistics vlan command</td>
<td>767–768, 934–935</td>
</tr>
<tr>
<td>show ip igmp snooping vlan command</td>
<td>757, 763–764</td>
</tr>
<tr>
<td>show interface command</td>
<td>374</td>
</tr>
<tr>
<td>show ip mroute command</td>
<td>770–771, 794–795, 892–893, 932</td>
</tr>
<tr>
<td>show ip mroute summary command</td>
<td>894</td>
</tr>
<tr>
<td>show ip msdp internal event-history route command</td>
<td>837–838</td>
</tr>
<tr>
<td>show ip msdp internal event-history tcp command</td>
<td>837–838</td>
</tr>
<tr>
<td>show ip msdp peer command</td>
<td>835–836</td>
</tr>
<tr>
<td>show ip ospf command</td>
<td>461</td>
</tr>
<tr>
<td>show ip ospf event-history command</td>
<td>464–465</td>
</tr>
<tr>
<td>show ip ospf interface command</td>
<td>461, 475–476</td>
</tr>
<tr>
<td>show ip ospf internal event-history adjacency command</td>
<td>47</td>
</tr>
<tr>
<td>show ip ospf internal event-history rib command</td>
<td>169–170</td>
</tr>
<tr>
<td>show ip ospf internal event-history txlist urib command</td>
<td>169</td>
</tr>
<tr>
<td>show ip ospf neighbors command</td>
<td>458–459</td>
</tr>
<tr>
<td>show ip ospf traffic command</td>
<td>463</td>
</tr>
<tr>
<td>show ip pim df command</td>
<td>805–806, 809</td>
</tr>
<tr>
<td>show ip pim group-range command</td>
<td>829–830</td>
</tr>
<tr>
<td>show ip pim interface command</td>
<td>782–783, 852–853</td>
</tr>
<tr>
<td>show ip pim internal event-history bidir command</td>
<td>806</td>
</tr>
<tr>
<td>show ip pim internal event-history data-header-register command</td>
<td>840–841</td>
</tr>
<tr>
<td>show ip pim internal event-history data-register-receive command</td>
<td>790</td>
</tr>
<tr>
<td>show ip pim internal event-history hello command</td>
<td>783–784</td>
</tr>
<tr>
<td>show ip pim internal event-history join-prune command</td>
<td>792–793, 806–807, 808, 846–847, 858, 865</td>
</tr>
<tr>
<td>show ip pim internal event-history null-register command</td>
<td>790, 791, 840–841, 857</td>
</tr>
<tr>
<td>show ip pim internal event-history rp command</td>
<td>819–820, 827–828</td>
</tr>
<tr>
<td>show ip pim internal event-history vpc command</td>
<td>857, 865–867</td>
</tr>
<tr>
<td>show ip pim internal vpc rpf-source command</td>
<td>856–857, 866–867</td>
</tr>
<tr>
<td>show ip pim neighbor command</td>
<td>781</td>
</tr>
<tr>
<td>show ip pim rp command</td>
<td>814–819, 822–827</td>
</tr>
<tr>
<td>show ip pim statistics command</td>
<td>783, 828–829</td>
</tr>
<tr>
<td>show ip prefix-list command</td>
<td>580–581</td>
</tr>
<tr>
<td>show ip route command</td>
<td>171, 419–421</td>
</tr>
</tbody>
</table>
show ip sla configuration command, 324
show ip sla statistics command, 323
show ip traffic command, 154–156, 611–612
show ip verify source interface command, 349–350
show ipv6 dhcp guard policy command, 369–370
show ipv6 dhcp relay statistics command, 358–359
show ipv6 icmp vaddr command, 378–379
show ipv6 interface command, 378–379
show ipv6 nd command, 355–356
show ipv6 nd raguard policy command, 364
show ipv6 neighbor command, 354
show ipv6 snooping policies command, 369–370
show isis adjacency command, 520–523
show isis command, 525–526
show isis database command, 558–560
show isis event-history command, 530–531
show isis interface command, 523–525, 526–527
show isis traffic command, 528–529
show key chain command, 417, 546
show lacp counters command, 262–263
show lacp internal info interface command, 263–264
show lacp neighbor command, 264
show lacp system-identifier command, 264
show logging log command, 88
show logging logfile command, 959
show logging onboard internal kernel command, 148
show logging onboard module 10 status command, 23
show mac address-table command, 198–199
show mac address-table dynamic vlan command, 796, 919–920, 923
show mac address-table multicast command, 764
show mac address-table vlan command, 305–306
show maintenance profile command, 727–728
show maintenance timeout command, 726
show module command, 96–98, 708
show module command output example, 96–97, 708
show monitor session command, 56–57
show ntp peer-status command, 82
show ntp statistics command, 83
show nxapi-server logs command, 973–975
show nxsdk internal event-history command, 967
show nxsdk internal service command, 965–966
show otv adjacency command, 889, 906–907, 910
show otv arp-nd-cache command, 916
show otv data-group command, 931
show otv internal adjacency command, 890
show otv internal event-history arp-nd command, 916–917
show otv isis database command, 899
show otv isis database detail command, 900–902
show otv isis hostname command, 899
show otv isis interface overlay command, 906
show otv isis internal event-history adjacency command, 898
show otv isis internal event-history iih command, 896–897
show otv isis ip redistribute mroute command, 930, 934
show otv isis mac redistribute route command, 903–904
show otv isis redistribute route command, 921–922
show otv isis site command, 895–896
show otv isis site statistics command, 904–905
show otv isis traffic overlay0 command, 904, 906
show otv mroute command, 928, 929
show otv mroute detail command, 929–930, 931, 933
show otv overlay command, 888
show otv route command, 902, 923
show otv route vlan command, 921
show otv site command, 889–890, 895, 911–912
show otv vlan command, 891–892, 920
show policy-map interface command, 114
show policy-map interface control-plane command, 189–190
show policy-map interface control-plane output example, 189–190
show policy-map system type network-qos command, 194–195
show port-channel compatibility-parameters command, 272
show port-channel load-balance command, 273–274
show port-channel summary command, 260–261, 272, 704–705
show port-channel traffic command, 273
show processes log pid command, 29
show processes log vdc-all command, 109–110
show queueing interface command, 114
show queueing interface command, 193, 194
show role command, 952
show routing clients command, 167–168
show routing event-history command, 647–648
show routing internal event-history msgs command, 169–170
show routing ip multicast event-history rib command, 770
show routing ip multicast source-tree detail command, 868–869
show routing memory statistics command, 171
show run aclmgr command, 572
show run all | include glean command, 161
show run copp all command, 186
show run netflow command, 76
show run otv command, 908–909, 917–918
show run pim command, 781
show run sflow command, 79
show run vdc command, 137
show running-config command, 45
show running-config copp command, 188–189
show running-config diff command, 43–44
show running-config diff example, 43–44
show running-config mmode command, 730
show running-config sla sender command, 324
show sflow command, 79–80
show sflow command output example, 80
show sflow statistics command, 80
show snapshots command, 725–726
show sockets client detail command, 157–158
show sockets connection tcp command, 615–616
show sockets connection tcp detail command, 157
show sockets internal event-history events command, 616–617
show sockets internal event-history events command example, 617
show sockets statistics all command, 159
show spanning-tree command, 225–227, 237–238, 281–282
show spanning-tree inconsistentports command, 246, 252
show spanning-tree interface command, 227
show spanning-tree mst command, 238–239
show spanning-tree mst configuration command, 237
show spanning-tree mst interface command, 239–240
show spanning-tree root command, 222–224, 225
show spanning-tree vlan command, 897–898
show system inband queuing statistics command, 150
show system internal access-list input entries detail command, 190
show system internal access-list input statistics command, 340–341, 348–349, 359, 367–368, 700–702
show system internal access-list interface command, 339–340, 367–368, 700–702
show system internal access-list interface e4/2 input statistics module 4 command, 573–574
show system internal aclmgr access-lists policies command, 574–575
show system internal aclmgr ppf node command, 575–576
show system internal adjmgr client command, 164–165
show system internal adjmgr internal event-history events command, 167
show system internal bfd event-history command, 695–699
show system internal bfd transition-history command, 699–700
show system internal copp info command, 191–192
show system internal eltm info interface command, 195
show system internal ethpm info interface command, 175–178, 195
show system internal fabricpath switch-id event-history errors command, 310
show system internal feature-mgr feature action command, 16
show system internal feature-mgr feature bfd current status command, 695
show system internal feature-mgr feature state command, 15
show system internal fex info fport command, 128–130
show system internal fex info sat port command, 128
show system internal flash command, 13–14, 24, 88–89
show system internal forwarding adjacency entry command, 173–174
show system internal forwarding route command, 173–174
show system internal forwarding table command, 350
show system internal mmode logfile command, 731
show system internal mts buffer summary command, 145–146
show system internal mts buffers detail command, 146–147
show system internal mts event-history errors command, 148
show system internal mts sup sap description command, 146–147
show system internal mts sup sap sap-id command, 11–12
show system internal mts sup sap stats command, 147–148
show system internal pixm info ltl command, 765
show system internal pktmgr client command, 151–152
show system internal pktmgr interface command, 152–153
show system internal pktmgr stats command, 153
show system internal port-client event-history port command, 179
show system internal port-client link-event command, 178–179
show system internal qos queueing stats interface command, 114–115
show system internal rpm as-path-access-list command, 672–673
show system internal rpm clients command, 588–589
show system internal rpm event-history rsw command, 588, 672–673
show system internal rpm ip-prefix-list command, 589, 668–669
show system internal sal info database vlan command, 350
show system internal sflow info command, 80
show system internal sup opcodes command, 147
show system internal sysmgr gsync-pending command, 32
show system internal sysmgr service all command, 10, 11, 146
show system internal sysmgr service all command example, 10
show system internal sysmgr service command, 10
show system internal sysmgr service command example, 10
show system internal sysmgr service dependency srvname command, 142–143
show system internal sysmgr state command, 31–32, 710–711
show system internal ufdm event-history debugs command, 171–172
show virtual-service command

show system internal vpcm info
interface command, 318–320
show system mode command, 720–722
show system redundancy ha status
command, 709
show system redundancy status
command, 29–30, 708–709
show system reset-reason command, 29, 110
show tech adjmgr command, 167
show tech arp command, 167
show tech bfd command, 704
show tech bgp command, 687
show tech dhcp command, 362
show tech ethpm command, 179
show tech glbp command, 390
show tech hsrp command, 379
show tech netstack command, 617, 687
show tech nxapi command, 975
show tech nxsdk command, 967
show tech routing ipv4 unicast
command, 687
show tech rpm command, 687
show tech track command, 334
show tech vpc command, 294
show tech vrrp command, 385
show tech vrrpv3 command, 385
show tech-support command, 51, 320, 749–750
show tech-support detail command, 124, 141
show tech-support eem command, 87
show tech-support eltm command, 195
show tech-support ethpm command, 130, 195
show tech-support fabricpath
command, 310
show tech-support fex command, 130
show tech-support ha command, 719
show tech-support issu command, 719
show tech-support mmode command, 731
show tech-support netflow command, 78
show tech-support netstack
command, 160
show tech-support pktmgr command, 160
show tech-support sflow command, 80
show tech-support vdc command, 141
show tunnel internal implicit otv brief
command, 890–891
show tunnel internal implicit otv
detail command, 922, 937
show tunnel internal implicit otv
tunnel_num command, 891
show udld command, 247–248
show udld internal event-history
errors command, 248–249
show vdc detail command, 137–138
show vdc detail command output
element, 137–138
show vdc internal event-history
command, 140–141
show vdc membership command, 139–140
show vdc resource detail command, 138–139
show vdc resource detail command
output example, 138–139
show vdc resource template
command, 131–132
show virtual-service command, 959–960
show virtual-service tech-support command, 960
show vlan command, 201–202, 214
show vlan command example, 201–202
show vlan private-vlan command, 210–211
show vpc command, 280–281, 284–285, 314–315
show vpc consistency-parameters command, 285–286
show vpc consistency-parameters command example, 285–286
show vpc consistency-parameters vlan command, 286–287
show vpc consistency-parameters vlan command example, 286–287
show vpc consistency-parameters vpc command, 287
show vpc consistency-parameters vpc-id command example, 287
show vpc orphan-ports command, 288
show vpc peer-keepalive command, 282–283
show vrrp command, 380–381
show vrrp statistics command, 381–382
show vrrpv3 command, 383–384
show vrrpv3 statistics command, 384–385
SIA (stuck in active) queries in EIGRP, 443–446
SIA timers output example, 444, 446
site VLAN for OTV, 882
SM (sparse mode), 772
SMUs (Software Maintenance Upgrades), 27–28
sniffing. See network sniffing
soft reconfiguration inbound in BGP, 654–657
software releases, 25–27
SOL (SPAN-on-Latency), 60–61
source command, 963
SPAN (Switched Port Analyzer), 54–57
configuring, 55–56
ERSPAN, 57–60
filtering traffic, 57
SPAN-on-Drop, 61–62
SPAN-on-Latency (SOL), 60–61
verifying, 56
SPAN-on-Drop, 61–62
SPT switchover on NX-4 example, 793
SSM (source specific multicast), 841–843
configuring, 843–845
verifying, 845–848
SSO (stateful switchover), 707–712
stateful restarts, 29
stateless restarts, 29
static joins, 748
static routes, object tracking with, 334
static RP, configuring, 812–813
status of overlay example, 888
STP (Spanning Tree Protocol), 218–219
forwarding loops
BPDU filter, 244–245
BPDU guard, 243–244
detecting and remediating, 241–242
MAC address notifications, 242–243
unidirectional links, 245–252
IEEE 802.1D standards, 219–220
MST (Multiple Spanning-Tree Protocol), 236
  configuring, 236–237
  tuning, 240–241
  verifying, 237–240
port states, 219
port types, 219
portfast enablement example, 235
RSTP (Rapid Spanning Tree Protocol), 220–221
  blocked switch port identification, 225–227
  interface STP cost, 221–222
  root bridge election, 222–224
  root port identification, 224–225
  tuning, 228–235
  verifying VLANs on trunk links, 227
terminology, 219–220
stub routers, 421–424
subnets in VLANs, 203. See also primary subnets
suboptimal path selection example, 562
suboptimal routing in IS-IS, 562–566
supervisor redundancy, 29–34
suspend individual (LACP), 271
suspending vPC orphan port during vPC failure example, 288
SVI (switched virtual interface), promiscuous PVLANs on, 215–217
switching from maintenance mode to normal mode example, 724–725
syslog
  configuring, 90
  with LSAs with duplicate RIDs example, 486, 487
  with LSPs with duplicate system IDs example, 547
  with neighbors configured, 472
  server, 90
  triggered loop guard example, 246
sysmgr (system manager), 9–11
system component troubleshooting, 142–143
  ARP and Adjacency Manager, 160–175
  EthPM and Port-Client, 175–179
  HWRL, CoPP, system QoS, 179–192
  MTS (Message and Transaction Service), 144–148
  MTU settings, 192–195
  Netstack and Packet Manager, 148–160
system maintenance mode always-use-custom-profile command, 728–730
system manager state information example, 710–711
system mode maintenance command, 720–722
system mode maintenance dont-generate-profile command, 730–731
system mode maintenance on-reload reset-reason command, 726–727
system mode maintenance timeout command, 726
system priority (LACP), 268–271
system QoS (quality of service), 179–192
system redundancy HA status example, 709
system redundancy state example, 709
system switchover command, 711–712
System-ID in IS-IS, 539, 546–549

T

tar append command, 20
tar create command, 20
tar extract command, 20
TCAM (ternary content addressable memory), 573–574
TCN (topology change notification), 232–235
TCP connect probes, 328–329
TCP sessions, verifying, 615–617
TCP socket connections example, 615
TCP socket creation and Netstack example, 157
TCPUDP component (Netstack), 156–160
technical support files, 44–45
telnet to port 179 usage example, 616
templates in BGP, 653–654
test packet-tracer command, 71–72
threshold for track list object example, 333
timers in OSPF, 476–478
TLVs (type, length, value) tuples, 512
in IIH, 514
in LSPs, 516
topologies
after SIA replies example, 445
EIGRP topology table, 395–396
IS-IS topology table, 898–905
verifying in FabricPath, 306
track object with static routes example, 334
track-list state, object tracking for, 332–333
traffic load-balancing (port-channels) troubleshooting, 272–274
trees in PIM, 772–773
trunk ports, 204–205
allowed VLANs, 206
configuring and verifying, 204
native VLANs, 206
PVLANs and, 217–218
verifying VLANs on, 227
tuning
BGP memory consumption, 650–653
MST (Multiple Spanning-Tree Protocol), 240–241
RSTP (Rapid Spanning Tree Protocol), 228–235
port priority, 232–233
root bridge placement, 228–229
root guard, 229
root port and blocked switch port locations, 229–232
topology changes and portfast, 232–235
tunnel depolarization, 942–944
Tx-Rx loop, 249–250
Type 1 vPC consistency-checker errors, 283–284
Type 2 vPC consistency-checker errors, 284
Type-1 networks, external OSPF routes, 496–497
Type-2 networks, external OSPF routes, 497–499
UDLD (unidirectional link detection), 246–250
configuring, 247
empty echo detection example, 249
event-history example, 248–249
UDP echo probes, 324–325
UDP jitter probes, 325–327
UFDM process, 171–175
UFDM route distribution to IPFIB
and acknowledgment example, 172
underscore (_) in RegEx, 677–678
unicast flooding, 198
with multicast enabled transport, 919–924
in OTV, 877
selective unicast flooding, 918–919
unicast forwarding components, 167
unicast routes from NX-2 for VLAN 215 and VLAN 216 example, 858
unicast RPF configuration and verification example, 351–352
unicast traffic, 734
unicast transport, multicast traffic with, 932–937
unidirectional links, 245
bridge assurance, 250–252
loop guard, 245–246
UDLD (unidirectional link detection), 246–250
unique router-ID in OSPF, 471
unique System-ID in IS-IS, 539
update generation process in BGP, 643–646
UPDATE message, 602
URIB (Unicast Routing Information Base), 167–171
clients, 168
route installation, 647–648
verifying FabricPath, 307
verifying vPC+, 316–317
URPF (Unicast Reverse Path Forwarding), 351–352
UUID (Universally Unique Identifier), 9

V
VDC (Virtual Device Contexts), 35–37, 130–131
configuring, 133–134
initializing, 134–136
internal event history logs example, 140–141
management, 137–142
out-of-band and in-band management, 137
resource templates, 131–132
verifying
access port mode example, 203–204
access-list counters
in hardware example, 574–575
in TCAM example, 573–574
ACLs (access control lists)
on line card for DHCP relay example, 339–340
statistics on line card for DHCP relay example, 340–341
active interfaces, 402–403
AED for VLAN 103 example, 920
anycast RP, 830–841
ARP ACLs, 348–349
ARP ND-Cache example, 916
ASM (any source multicast), 788–789
Auto-RP, 813–820
BFD (bidirectional forwarding detection)
  with echo function, 702–703
  neighbors example, 694–695
  sessions, 693–707
BGP (Border Gateway Protocol), 605–609
  ACLs and firewalls, 613–615
  configuration, 610–611
  reachability and packet loss, 611–613
  TCP sessions, 615–617
BiDIR (Bidirectional), 805–811
BPDU filter example, 245
BSR (bootstrap router), 820–830
community PVLAN configuration example, 214
configuration incompatibilities example, 713–714
connectivity
  after virtual link example, 484–485
  between primary subnets example, 411
  with promiscuous PVLAN SVI example, 216–217
  between PVLANs example, 214–215
contents of logflash: directory example, 24
CoPP (control plane policing)
  EIGRP example, 407–408
  IS-IS example, 532
  NetFlow, 78
  OSPF example, 465–466
current bit-rate of OTV control-group example, 894
DAI (dynamic ARP inspection), 345–346
detailed dynamic tunnel parameters example, 891
DHCP relay, 337–338
DHCPv6 guard configuration and policy, 369–370
EEM (Embedded Event Manager), 85–86
EIGRP (Enhanced Interior Gateway Protocol)
  hello and hold timers example, 415–416
  neighbors, 423
  packets, 405–409
emulated switch-IDs example, 315
ERSPAN session, 59–60
FabricPath, 303–310
  core interfaces, 303–304
  IS-IS adjacency, 304–305
  software table in hardware, 308–309
  switch-IDs, 303, 310
  topologies, 306
  in URIB, 307, 309
VLANs (virtual LANs), 305–306
FEX (Fabric Extender), 126–128
filtering SPAN traffic, 57
forwarding adjacency example, 163
FP core interfaces example, 303–304
FP MAC information in vPCM example, 318–320
hardware forwarding on module 3, 799
hardware rate-limiters on N7k and N9k switches example, 181–182
hardware statistics for IPv6 snooping example, 367–368
HSRP (Hot Standby Routing Protocol), 373–374
HSRPv6 virtual address, 379
IGMP (Internet Group Management Protocol), 761–771
IGMP snooping example, 757
IGMPv3 on NX-4, 846
ingress L3 unicast flow drops example, 62
interface's OSPF network type example, 475–476
I/O module MFIB on module 3, 798
IOS devices after NX-OS metric transition mode example, 556
IS-IS (Intermediate System-to-Intermediate System)
adjacency example, 305
interface, 523–525
interface level type example, 542
metric transition mode, 555
neighbors, 520–523
packets, 528–535
process level type example, 541
protocol, 525–526
system IDs example, 549
isolated PVLANs
communications example, 211–212
configuration example, 210–211
keychains example, 417
LACP (link-aggregation control packets), 262–265
LACP speed state, 270
Layer 3 routing over vPC, 294
local and remote FP routes in URIB example, 316–317
maintenance and normal profile configurations example, 727–728
maximum links, 267
MFDM on NX-2, 797
missing 172.16.1.0/24 network example, 493–494
MROUTE, 789–795
MROUTE in transport network, 932
MROUTE on NX-2, 795
MST (Multiple Spanning-Tree Protocol), 237, 240
MTU
under ELTM process, 195
under ethpm process, 195
multicast routing for OTV control-group example, 893
NET addressing example, 541
network QoS policy, 195
new path after new reference OSPF bandwidth is configured on R1 and R2 example, 503–504
no services pending synchronization example, 32, 34
NX-OS BGP peering, 607
on-reload reset-reason, 726–727
optimal routing example, 493
ORIB entry for host C example, 921
OSPF (Open Shortest Path First)
area settings example, 474
encrypted authentication example, 481
neighbors, 458–460
packets, 463–467
packets using Ethanalyzer example, 467
packets with ACL example, 467
plaintext authentication example, 479
OTV (Overlay Transport Virtualization)
IS-IS adjacencies, 888–898
next-hop adjacency tracking example, 946
site adjacency example, 896
packet tracer, 71–72
PBR-based traffic example, 593
PIM ASM platform, 795–799
PIM interfaces and neighbors, 780–785
platform FIB, 173–174, 176–178
platform LTL index example, 765
port priority impact on spanning tree protocol topology example, 232–233
port-channel status, 260–262
PPF database example, 575–576
promiscuous PVLAN SVI mapping example, 216
PVLAN switchport type example, 211
redistributed networks example, 567
remote area routes
  on NX-1 and NX-4 example, 483
  on NX-2 and NX-3 example, 482–483
RFC1583 compatibility example, 500
root and blocking ports for VLAN example, 226–227
SAL database info and FIB for IPSG, 350
site group to delivery group mapping example, 931
site-ID of OTV IS-IS neighbor example, 890
site-VLAN spanning-tree example, 897–898
size and location of PSS in flash file system example, 13–14
software table in hardware for FP route example, 308–309
SPAN (Switched Port Analyzer), 56 spanning tree protocol root bridge example, 223
SSM (source specific multicast), 845–848
state and available space for logflash: example, 24
suboptimal routing example, 491
sysmgr state on standby supervisor example, 33
total path cost example, 230–231
trunk port, 204
UDLD switch port status example, 247–248
URPF (Unicast Reverse Path Forwarding), 351–352
VLANs on trunk links, 227
vPC (virtual port-channel)
  autorecovery, 289
  autorecovery example, 289
  consistency-checker, 283–287
  domain status, 280–282
  peer-gateway, 291
  peer-gateway example, 291
  peer-keepalive link, 282–283
vPC+, 314–320
  emulated switches, 315
  MAC addresses, 315–316
  show vpc command, 314–315
  in URIB, 316–317
  in vPCM, 318–320
vPC-connected receiver, 861–869
vPC-connected source, 849–861
VRRP (Virtual Router Redundancy Protocol), 380–381
which OTV ED is AED example, 892
viewing
  access port configuration command example, 203
  and changing LACP system priority example, 268
virtual service list and resource utilization example

contents of specific file in logflash: example, 24–25
CoPP policy and creating custom CoPP policy example, 189
detail information for redistribution example, 590
detailed version of spanning-tree state example, 234
EIGRP (Enhanced Interior Gateway Protocol)
authentication on interfaces example, 417
passive interfaces example, 404
retry values for neighbors example, 410–411
routes on NX-1 example, 420–421
IIH authentication example, 545–546
inconsistent ports example, 252
inconsistent spanning tree protocol ports example, 246
interface specific MST settings example, 240
keychain passwords example, 481, 546
LACP (link-aggregation control packets)
neighbor information example, 264
packet counters example, 263
time stamps for transmissions on interface example, 263–264
MAC addresses on Nexus switch example, 199
nondefault OSPF forwarding address example, 492
number of classic and wide EIGRP neighbors example, 438
number of RPM clients per protocol example, 588–589
OSPF (Open Shortest Path First)
password for simple authentication example, 480
RID example, 471
port-channels
hash algorithm example, 273
interface status example, 262
summary status example, 260
RPM (Route Policy Manager)
event-hitory example, 588
perspective example prefix-lists, 589
STP (Spanning Tree Protocol)
behavior changes with vPC example, 281–282
event-hitory example, 234
port priority example, 232
spanning tree protocol type of ports with bridge assurance example, 250–251
traffic load on member interfaces example, 273
VLANs (virtual LANs)
allowed on trunk link example, 206
participating with spanning tree protocol on interface example, 227
vPC (virtual port-channel)
orphan ports example, 288
peer-keepalive status example, 282
status example, 280–281
virtual link configuration example, 484
virtual service list and resource utilization example, 960
virtualization

Virtual Device Contexts (VDCs), 35–37
virtual port channels (vPC), 37–39
Virtual Routing and Forwarding (VRF), 37

VLANs (virtual LANs), 200–201
access ports, 203–204
creating, 201–203
IGMP snooping group membership example, 764
loop-free topologies. See STP (Spanning Tree Protocol)
mapping
  on L2 trunk example, 942
  in OTV, 941–942
  on overlay interface example, 942
multiple subnets in, 203
PVLANs (private VLANs), 207–208
  communication capability between hosts, 208
  community PVLANs, 212–215
  isolated PVLANs, 208–212
  promiscuous PVLANs on SVI, 215–217
  trunking between switches, 217–218
reserved VLAN, 870
site VLAN for OTV, 882
trunk ports, 204–205
  allowed VLANs, 206
  native VLANs, 206
verifying
  in FabricPath, 305–306
  on trunk links, 227
vPC (virtual port-channel), 37–39, 274–275
ARP synchronization, 291–292
autorecovery, 289
backup Layer 3 routing, 292–293
configuring, 278–280
domains, 275–276
IGMP snooping state on NX-4 example, 854–855
Layer 3 routing, 293–294
member links, 277
multicast traffic, 848–849
duplicate packets, 870
receiver configuration and verification, 861–869
reserved VLAN, 870
source configuration and verification, 849–861
operational behavior, 277–278
orphan ports, 288
peer link, 277
peer-gateway, 289–291
peer-keepalive link, 276–277
status with consistency checker error example, 284–285
topology, 275–276
verifying
  consistency-checker, 283–287
  domain status, 280–282
  peer-keepalive link, 282–283
vPC+
configuring, 311–314
verifying, 314–320
  emulated switches, 315
  MAC addresses, 315–316
  show vpc command, 314–315
  in URIB, 316–317
  in vPCM, 318–320
vPCM (vPC Manager), verifying
  vPC+, 318–320
VRF (Virtual Routing and Forwarding), 37
VRRP (Virtual Router Redundancy Protocol), 380–385
  configuring, 380
  state and detail information example, 381
  statistics, 381–382
  verifying, 380–381
VRRPv3, 382–385
VRRPv3, 382–385

wide metrics
  versus classic metrics in EIGRP, 433–439
  on NX-1, NX-2, and NX-3 example, 437–438
  on NX-1, NX-2, NX-3, and NX-6 example, 438–439
  on NX-1 and NX-2 example, 436–437

X
xml utility, 42

Y
yum command, 954