



Official Cert Guide

Learn, prepare, and practice for exam success



CCNP

Routing and Switching TSHOOT 300-135

ciscopress.com

RAYMOND LACOSTE
KEVIN WALLACE, CCIE® No. 7945

<https://t.me/learningnets>

CCNP Routing and Switching TSHOOT 300-135

Official Cert Guide

Raymond Lacoste
CCSI/CCNP

Kevin Wallace
CCIE No. 7945

Cisco Press

800 East 96th Street

Indianapolis, IN 46240

<https://t.me/learningnets>

From the Library of Outcast Outcast

CCNP Routing and Switching TSHOOT 300-135 Official Cert Guide

Raymond Lacoste, CCSI/CCNP

Kevin Wallace, CCIE No. 7945

Copyright© 2015 Pearson Education, Inc.

Published by:

Cisco Press

800 East 96th Street

Indianapolis, IN 46240 USA

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

Printed in the United States of America

First Printing December 2014

Library of Congress Control Number: 2014950275

ISBN-10: 1-58720-561-0

ISBN-13: 978-1-58720-561-3

Warning and Disclaimer

This book is designed to provide information about the 300-135 Troubleshooting and Maintaining Cisco IP Networks (TSHOOT) exam for the CCNP Routing and Switching certification. Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied.

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

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

Trademark Acknowledgments

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

Special Sales

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

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

For questions about sales outside the U.S., please contact international@pearsoned.com.

Feedback Information

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

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

We greatly appreciate your assistance.

Publisher: Paul Boger

Associate Publisher: Dave Dusthimer

Business Operation Manager, Cisco Press:
Jan Cornelssen

Executive Editor: Brett Bartow

Managing Editor: Sandra Schroeder

Development Editor: Ellie Bru

Project Editor: Mandie Frank

Copy Editor: Keith Cline

Technical Editors: Ryan Lindfield, Diane Teare

Team Coordinator: Vanessa Evans

Designer: Mark Shirar

Composition: Tricia Bronkella

Indexer: Lisa Stumpf

Proofreader: The WordSmithery LLC

About the Authors

Raymond Lacoste is a Cisco Certified Systems Instructor (CCSI) who has dedicated his IT career to teaching others. Starting out as a mentor at Skillsoft, he helped students with their studies, explaining various Cisco, Microsoft, and industry-related concepts in ways that improved the students understanding. Now he spends his days at Skillsoft teaching the CCNA and CCNP Routing and Switching certification track. He has taught over 300 Cisco classes in addition to the countless practice labs, demonstrations, hands-on labs, and student guides he has developed. However, it is not just about teaching, it is also about learning. To date, Raymond has passed more than 100 IT certification exams as he continues to keep his learning and knowledge up-to-date. His certification wall includes various Cisco certifications, Microsoft certifications, CompTIA certifications, and the ISC2 CISSP (Certified Information Systems Security Professional) designation. He was also awarded the Cisco Sirius Top Quality Instructor award. His next goal is to achieve the CCIE designation in Routing and Switching. Raymond lives in Atlantic, Canada, with his wife, Melanie, and two children.

Kevin Wallace, CCIEx2 (Collaboration and R/S) #7945, CCSI #20061: With Cisco experience dating back to 1989, Kevin has been a network design specialist for the Walt Disney World Resort, an instructor of Cisco courses for Skillsoft, and a network manager for Eastern Kentucky University.

Kevin currently produces video courses and writes books for Cisco Press/Pearson IT Certification (<http://kwtrain.com/books>), and he lives in central Kentucky with his wife (Vivian) and two daughters (Stacie and Sabrina).

Kevin can be followed on these social media platforms.

Blog: <http://kwtrain.com>

Twitter: <http://twitter.com/kwallaceccie>

Facebook: <http://facebook.com/kwallaceccie>

YouTube: <http://youtube.com/kwallaceccie>

LinkedIn: <http://linkedin.com/in/kwallaceccie>

Google+: <http://google.com/+KevinWallace>

About the Technical Reviewers

Ryan Lindfield is an instructor and technical consultant with Stormwind. On a typical day he's broadcasting official Cisco training from a video studio. When not in the virtual classroom, he can be found supporting customer networks. Ryan has nearly 20 years of technical consulting experience, and over a decade in the classroom. He has delivered training for network, security, and data center technologies around the world. Certifications include: CCNP Routing & Switching, CCNP Security, HP Master Accredited Systems Engineer, VMware VCP, CEH, CISSP, SANS GFCA, CISSP, ECSA, CHFI, CPTE, CPTC, OSWP, and many Microsoft and CompTIA certifications. Ryan leads a 150 member Defcon user group in Tampa, FL, and has given presentations for ISC2 and B-Sides computer security events.

Diane Teare, P.Eng, CCNP, CCDP, CCSI, PMP, is a professional in the networking, training, project management, and e-learning fields. She has more than 25 years of experience in designing, implementing, and troubleshooting network hardware and software, and has been involved in teaching, course design, and project management. She has extensive knowledge of network design and routing technologies. Diane is a Cisco Certified Systems Instructor (CCSI), and holds her Cisco Certified Network Professional (CCNP), Cisco Certified Design Professional (CCDP), and Project Management Professional (PMP) certifications. She is an instructor, and the Course Director for the CCNA and CCNP Routing and Switching curriculum, with one of the largest authorized Cisco Learning Partners. She was the director of e-learning for the same company, where she was responsible for planning and supporting all the company's e-learning offerings in Canada, including Cisco courses. Diane has a Bachelor's degree in applied science in electrical engineering and a Master's degree in applied science in management science. She authored or co-authored the following Cisco Press titles: the first and second editions of *Implementing Cisco IP Routing (ROUTE)*; the second edition of *Designing Cisco Network Service Architectures (ARCH)*; *Campus Network Design Fundamentals*; the three editions of *Authorized Self-Study Guide Building Scalable Cisco Internetworks (BSCI)*; and *Building Scalable Cisco Networks*. Diane edited the first two editions of the *Authorized Self-Study Guide Designing for Cisco Internetwork Solutions (DESGN)*, and also edited *Designing Cisco Networks*.

Dedications

This book is dedicated to two very special people who supported me in my early years of IT, without whom this book would not have been possible. I will forever be grateful for the opportunity you gave me so many years ago to pursue my career. Thank you!

Raymond Lacoste

Acknowledgments

A big thank you to my wife for encouraging me to write this book and supporting me over the months that it took to complete it. Great big hugs to my two wonderful children, ages 9 and 5, who had no idea why Daddy was always sitting at the computer; for some strange reason, though, they knew that it was important and supported me in their own mysterious ways. I love you guys!

An equally big thank you to my parents, without whom I would not be where I am or who I am today, and to my sister, Terry-Anne, who always kicked me in the right direction.

Thanks to Dan Young, my mentor and the Director of Live Learning at Skillsoft, for all the support and encouragement you have provided me all these years.

I'd like to thank Ellie Bru, my Development Editor, for organizing and putting into action all the parts needed to develop this book (definitely not an easy task).

Thank you to Mandie Frank, my Production Editor, for putting all the final pieces of this book together so nicely and making sure that it resembles a book.

Thank you to Diane Teare and Ryan Lindfield for reviewing the book and making sure it's technically sound.

Keith Cline, thank you for making sure all i's were "crossed" and t's "dotted" within the book. (HaHaHa) You found some items in this book that I didn't even know existed. Thank you!

Thank you to Brett Bartow, my Executive Editor, for giving me the opportunity to write this detailed book.

A big thank you to Kevin Wallace, the author of the previous edition of TSHOOT and a friend, who passed the torch on to me for this edition. Thank you.

Lastly, thank you to the entire team at Cisco Press, their families and friends, who work extremely hard to produce high-quality training materials.

—Raymond Lacoste

Contents at a Glance

Introduction xxx

Part I Fundamental Troubleshooting and Maintenance Concepts

Chapter 1 Introduction to Troubleshooting and Network Maintenance 3

Chapter 2 Troubleshooting and Maintenance Tools 41

Chapter 3 Troubleshooting Device Performance 93

Part II Troubleshooting Cisco Catalyst Switch Features

Chapter 4 Troubleshooting Layer 2 Trunks, VTP, and VLANs 129

Chapter 5 Troubleshooting STP and Layer 2 EtherChannel 169

Chapter 6 Troubleshooting Inter-VLAN Routing and Layer 3 EtherChannels 209

Chapter 7 Troubleshooting Switch Security Features 247

Chapter 8 Troubleshooting First-Hop Redundancy Protocols 287

Part III Troubleshooting Router Features

Chapter 9 Troubleshooting IPv4 Addressing and Addressing Technologies 335

Chapter 10 Troubleshooting IPv6 Addressing and Addressing Technologies 367

Chapter 11 Troubleshooting IPv4 and IPv6 ACLs and Prefix Lists 397

Chapter 12 Troubleshooting Basic IPv4/IPv6 Routing and GRE Tunnels 423

Chapter 13 Troubleshooting RIPv2 and RIPv6 463

Chapter 14 Troubleshooting EIGRP 513

Chapter 15 Troubleshooting OSPF 587

Chapter 16 Troubleshooting Route Maps and Policy-Based Routing 675

Chapter 17 Troubleshooting Redistribution 697

Chapter 18 Troubleshooting BGP 749

Part IV Troubleshooting Management

Chapter 19 Troubleshooting Management Protocols and Tools 815

Chapter 20 Troubleshooting Management Access 851

Part V Final Preparation

Chapter 21 Additional Trouble Tickets 871

Chapter 22 Final Preparation 943

Part VI Appendixes

Appendix A Answers to the “Do I Know This Already” Quizzes 951

Appendix B TSHOOT Exam Updates 957

Index 960

CD-Only Appendixes and Glossary

Appendix C Memory Tables

Appendix D Memory Tables Answer Key

Appendix E Study Planner

Glossary

Contents

Introduction xxx

Part I Fundamental Troubleshooting and Maintenance Concepts

Chapter 1 Introduction to Troubleshooting and Network Maintenance 3

“Do I Know This Already?” Quiz 3

Foundation Topics 9

Introduction to Troubleshooting 9

Defining Troubleshooting 9

The Value of Structured Troubleshooting 11

A Structured Approach 13

1. Problem Report 13

2. Collect Information 14

3. Examine Collected Information 15

4. Eliminate Potential Causes 16

5. Propose an Hypothesis 17

6. Verify Hypothesis 18

7. Problem Resolution 19

Popular Troubleshooting Methods 20

The Top-Down Method 21

The Bottom-Up Method 21

The Divide-and-Conquer Method 22

The Following the Traffic Path Method 23

The Comparing Configurations Method 23

The Component Swapping Method 24

Practice Exercise: Selecting a Troubleshooting Approach 25

Introduction to Network Maintenance 26

Defining Network Maintenance 26

Proactive Versus Reactive Network Maintenance 27

Well-Known Network Maintenance Models 28

Example of Adapting a Network Maintenance Model 28

Common Maintenance Procedures 29

Routine Maintenance Tasks 29

Scheduled Maintenance 30

Managing Network Changes 30

Maintaining Network Documentation 32

Restoring Operations After a Failure	33
Measuring Network Performance	34
The Troubleshooting and Network Maintenance Relationship	34
Maintaining Current Network Documentation	35
Establishing a Baseline	36
Communication	36
Change Management	37
Exam Preparation Tasks	39
Review All Key Topics	39
Define Key Terms	39
Chapter 2 Troubleshooting and Maintenance Tools	41
“Do I Know This Already?” Quiz	41
Foundation Topics	45
The Troubleshooting and Network Maintenance Toolkit	45
Network Documentation Tools	46
Basic Tools	47
<i>CLI Tools</i>	47
<i>GUI Tools</i>	48
<i>Recovery Tools</i>	48
<i>Logging Tools</i>	53
<i>Network Time Protocol as a Tool</i>	56
Advanced Tools	57
<i>Overview of SNMP and NetFlow</i>	57
<i>Creating a Baseline with SNMP and NetFlow</i>	58
SNMP	58
NetFlow	59
Cisco Support Tools	64
Using Cisco IOS to Verify and Define the Problem	64
Ping	64
Telnet	67
Traceroute	67
Using Cisco IOS to Collect Information	68
Filtering the Output of show Commands	69
Redirecting show Command Output to a File	73
Troubleshooting Hardware	74

	Collecting Information in Transit	75
	Performing Packet Captures	75
	SPAN	76
	RSPAN	78
	Using Tools to Document a Network	80
	Exam Preparation Tasks	85
	Review All Key Topics	85
	Define Key Terms	86
	Complete Tables and Lists from Memory	86
	Command Reference to Check Your Memory	86
Chapter 3	Troubleshooting Device Performance	93
	“Do I Know This Already?” Quiz	93
	Foundation Topics	96
	Troubleshooting Switch Performance Issues	96
	<i>Cisco Catalyst Switch Troubleshooting Targets</i>	96
	<i>TCAM Troubleshooting</i>	101
	<i>High CPU Utilization Troubleshooting on a Switch</i>	105
	Troubleshooting Router Performance Issues	106
	<i>Excessive CPU Utilization</i>	107
	<i>Understanding Packet-Switching Modes (Routers and Multilayer Switches)</i>	113
	<i>Troubleshooting Packet-Switching Modes</i>	116
	<i>Excessive Memory Utilization</i>	121
	Exam Preparation Tasks	124
	Review All Key Topics	124
	Define Key Terms	124
	Complete Tables and Lists from Memory	125
	Command Reference to Check Your Memory	125
Part II	Troubleshooting Cisco Catalyst Switch Features	
Chapter 4	Troubleshooting Layer 2 Trunks, VTP, and VLANs	129
	“Do I Know This Already?” Quiz	129
	Foundation Topics	132
	Frame-Forwarding Process	132
	Troubleshooting Trunks	140
	Encapsulation Mismatch	141
	Incompatible Trunking Modes	143

VTP Domain Name Mismatch	146
Native VLAN Mismatch	146
Allowed VLANs	147
Troubleshooting VTP	148
Domain Name Mismatch	148
Version Mismatch	149
Mode Mismatch	149
Password Mismatch	151
Higher Revision Number	151
Troubleshooting VLANs	152
Incorrect IP Addressing	152
Missing VLAN	153
Incorrect Port Assignment	154
The MAC Address Table	155
Layer 2 Trouble Tickets	157
Trouble Ticket 4-1	158
Trouble Ticket 4-2	160
Exam Preparation Tasks	165
Review All Key Topics	165
Define Key Terms	165
Complete Tables and Lists from Memory	166
Command Reference to Check Your Memory	166
Chapter 5	Troubleshooting STP and Layer 2 EtherChannel
	169
“Do I Know This Already?” Quiz	169
Foundation Topics	172
Spanning Tree Protocol Overview	172
Reviewing STP Operation	173
<i>Determining Root Port</i>	175
<i>Determining Designated Port</i>	176
<i>Determining Nondesignated Port</i>	176
Collecting Information About an STP Topology	177
Gathering STP Information	177
Gathering MSTP Information	179
STP Troubleshooting Issues	180
Corruption of a Switch’s MAC Address Table	180
Broadcast Storms	181

Troubleshooting STP Features	182
PortFast	183
BPDU Guard	184
BPDU Filter	187
Root Guard	189
Loop Guard	190
STP Trouble Tickets	190
Trouble Ticket 5-1	191
Trouble Ticket 5-2	194
Trouble Ticket 5-3	196
Troubleshooting Layer 2 EtherChannel	199
Reviewing Layer 2 EtherChannel	199
EtherChannel Trouble Tickets	200
Trouble Ticket 5-4	201
Trouble Ticket 5-5	204
Exam Preparation Tasks	206
Review All Key Topics	206
Define Key Terms	206
Complete Tables and Lists from Memory	207
Command Reference to Check Your Memory	207
Chapter 6 Troubleshooting Inter-VLAN Routing and Layer 3 EtherChannels	209
“Do I Know This Already?” Quiz	209
Foundation Topics	212
Troubleshooting a Router-on-a-Trunk/Stick	212
Router-on-a-Trunk/Stick Trouble Tickets	213
Trouble Ticket 6-1	214
Trouble Ticket 6-2	218
Troubleshooting Switched Virtual Interfaces	221
Reviewing SVIs	221
Troubleshooting SVIs	223
SVI Trouble Tickets	224
Trouble Ticket 6-3	225
Trouble Ticket 6-4	230
Troubleshooting Routed Ports	233
Routed Ports Trouble Tickets	234
Trouble Ticket 6-5	235

Troubleshooting Layer 3 EtherChannel	237
Layer 3 EtherChannel Trouble Tickets	239
Trouble Ticket 6-6	240
Exam Preparation Tasks	244
Review All Key Topics	244
Define Key Terms	244
Complete Tables and Lists from Memory	245
Show Command Reference to Check Your Memory	245
Chapter 7 Troubleshooting Switch Security Features	247
“Do I Know This Already?” Quiz	247
Foundation Topics	250
Troubleshooting Port Security	250
Common Port Security Issues	250
<i>Port Security Configured but Not Enabled</i>	250
<i>Static MAC Address Not Configured Correctly</i>	251
<i>Maximum Number of MAC Addresses Reached</i>	253
<i>Legitimate Users Being Blocked Because of Violation</i>	254
<i>Running Configuration Not Saved to Startup Configuration</i>	260
Port Security Trouble Tickets	261
Trouble Ticket 7-1	261
Troubleshooting Spoof-Prevention Features	265
DHCP Snooping	265
Dynamic ARP Inspection	267
IP Source Guard	268
Spoof-Prevention Features Trouble Tickets	270
Trouble Ticket 7-2	270
Troubleshooting Access Control	273
Protected Ports	273
Private VLANs	275
VACLs	279
Exam Preparation Tasks	281
Review All Key Topics	281
Define Key Terms	282
Command Reference to Check Your Memory	282

Chapter 8 Troubleshooting First-Hop Redundancy Protocols 287

“Do I Know This Already?” Quiz	287
Foundation Topics	290
Troubleshooting HSRP	290
Reviewing HSRP	290
HSRP Converging After a Failure	291
HSRP Verification and Troubleshooting	292
<i>Virtual Router MAC Address</i>	293
<i>Interface Tracking</i>	293
<i>Verifying First Hop</i>	294
<i>Debug</i>	296
HSRP Trouble Tickets	297
Trouble Ticket 8-1	297
Trouble Ticket 8-2	300
Trouble Ticket 8-3	302
Troubleshooting VRRP	306
Reviewing VRRP	306
VRRP Verification and Troubleshooting	308
<i>Virtual Router MAC Address</i>	309
<i>Object Tracking</i>	309
<i>Verifying First Hop</i>	310
VRRP Trouble Tickets	312
Trouble Ticket 8-4	312
Trouble Ticket 8-5	315
Troubleshooting GLBP	318
Reviewing GLBP	319
GLBP Verification and Troubleshooting	321
<i>Virtual Router MAC Addresses</i>	323
<i>GLBP Object Tracking</i>	323
<i>Verifying GLBP First Hop</i>	325
GLBP Trouble Tickets	326
Trouble Ticket 8-6	327
Trouble Ticket 8-7	329
Comparing HSRP, VRRP, and GLBP	330
Exam Preparation Tasks	332
Review All Key Topics	332

Define Key Terms	333
Complete Tables and Lists from Memory	333
Command Reference to Check Your Memory	333

Part III Troubleshooting Router Features

Chapter 9 Troubleshooting IPv4 Addressing and Addressing Technologies 335

“Do I Know This Already?” Quiz	335
Foundation Topics	338
Troubleshooting IPv4 Addressing	338
IPv4 Addressing Issues	338
Determining IP Addresses Within a Subnet	341
Troubleshooting DHCP for IPv4	342
Reviewing DHCP Operations	342
Potential DHCP Troubleshooting Issues	347
DHCP Troubleshooting Commands	348
Troubleshooting NAT	350
Reviewing NAT	350
NAT Troubleshooting Issues	353
NAT Troubleshooting Commands	354
IPv4 Addressing and Addressing Technologies Trouble Tickets	356
Trouble Ticket 9-1	356
Trouble Ticket 9-2	358
Trouble Ticket 9-3	361
Exam Preparation Tasks	364
Review All Key Topics	364
Define Key Terms	365
Command Reference to Check Your Memory	365

Chapter 10 Troubleshooting IPv6 Addressing and Addressing Technologies 367

“Do I Know This Already?” Quiz	367
Foundation Topics	370
Troubleshooting IPv6 Addressing	370
IPv6 Addressing Review	370
<i>Neighbor Solicitation and Neighbor Advertisement</i>	370
<i>EUI-64</i>	373
Troubleshooting IPv6 Address Assignment	375
Stateless Address Autoconfiguration/SLAAC	375
Stateful DHCPv6	381

Stateless DHCPv6	382
DHCPv6 Operation	384
DHCPv6 Relay Agent	385
IPv6 Addressing Trouble Tickets	386
Trouble Ticket 10-1	386
Trouble Ticket 10-2	389
Exam Preparation Tasks	394
Review All Key Topics	394
Define Key Terms	395
Command Reference to Check Your Memory	395
Chapter 11 Troubleshooting IPv4 and IPv6 ACLs and Prefix Lists	397
“Do I Know This Already?” Quiz	397
Foundation Topics	401
Troubleshooting IPv4 ACLs	401
Reading an IPv4 ACL	401
Using an IPv4 ACL for Filtering	403
Using a Time-Based IPv4 ACL	403
IPv4 ACL Trouble Tickets	405
Trouble Ticket 11-1	405
Troubleshooting IPv6 ACLs	407
Reading an IPv6 ACL	408
Using an IPv6 ACL for Filtering	409
IPv6 ACL Trouble Tickets	410
Trouble Ticket 11-2	410
Troubleshooting Prefix Lists	414
Reading a Prefix List	414
Prefix List Processing	415
Prefix List Trouble Tickets	416
Trouble Ticket 11-3	417
Exam Preparation Tasks	419
Review All Key Topics	419
Define Key Terms	419
Command Reference to Check Your Memory	419

Chapter 12 Troubleshooting Basic IPv4/IPv6 Routing and GRE Tunnels 423

“Do I Know This Already?” Quiz	423
Foundation Topics	427
Packet-Forwarding Process	427
Reviewing Layer 3 Packet-Forwarding Process	427
Troubleshooting the Packet-Forwarding Process	431
Troubleshooting Routing Information Sources	435
Data Structures and the Routing Table	436
Sources of Route Information	436
Troubleshooting Static Routes	438
IPv4 Static Routes	439
IPv6 Static Routes	443
Static Routing Trouble Tickets	445
Trouble Ticket 12-1	445
Trouble Ticket 12-2	448
Troubleshooting GRE Tunnels	450
Exam Preparation Tasks	459
Review All Key Topics	459
Define Key Terms	460
Complete Tables and Lists from Memory	460
Command Reference to Check Your Memory	460

Chapter 13 Troubleshooting RIPv2 and RIPv6 463

“Do I Know This Already?” Quiz	463
Foundation Topics	466
Troubleshooting RIPv2	466
Missing RIPv2 Routes	466
<i>Interface Is Shut Down</i>	469
<i>Wrong Subnet</i>	469
<i>Bad or Missing Network Statement</i>	470
<i>Passive Interface</i>	471
<i>Wrong Version</i>	473
<i>Max Hop Count Exceeded</i>	475
<i>Authentication</i>	477
<i>Route Filtering</i>	479
<i>Split Horizon</i>	480
<i>Autosummarization</i>	482
<i>Better Source of Information</i>	483

<i>ACLs</i>	485
<i>Load Sharing</i>	485
Other RIP Issues	486
<i>Missing Default Route</i>	486
<i>Route Summarization</i>	487
Troubleshooting RIPng	492
RIPv2 and RIPng Trouble Tickets	498
Trouble Ticket 13-1	498
Trouble Ticket 13-2	502
Trouble Ticket 13-3	506
Exam Preparation Tasks	509
Review All Key Topics	509
Define Key Terms	510
Command Reference to Check Your Memory	510
Chapter 14 Troubleshooting EIGRP	513
“Do I Know This Already?” Quiz	513
Foundation Topics	517
Troubleshooting EIGRP for IPv4	517
Troubleshooting EIGRP for IPv4 Neighbor Adjacencies	517
<i>Interface Is Down</i>	518
<i>Mismatched Autonomous System Numbers</i>	518
<i>Incorrect Network Statement</i>	520
<i>Mismatched K Values</i>	522
<i>Passive Interface</i>	523
<i>Different Subnets</i>	524
<i>Authentication</i>	525
<i>ACLs</i>	527
<i>Timers</i>	528
Troubleshooting EIGRP for IPv4 Routes	528
<i>Bad or Missing Network Command</i>	529
<i>Better Source of Information</i>	530
<i>Route Filtering</i>	534
<i>Stub Configuration</i>	535
<i>Interface Is Shut Down</i>	537
<i>Split-horizon</i>	537

Troubleshooting Miscellaneous EIGRP for IPv4 Issues	539
<i>Feasible Successors</i>	539
<i>Discontiguous Networks and Autosummarization</i>	542
<i>Route Summarization</i>	543
<i>Load Balancing</i>	544
EIGRP for IPv4 Trouble Tickets	546
Trouble Ticket 14-1	546
Trouble Ticket 14-2	553
Trouble Ticket 14-3	557
Troubleshooting EIGRP for IPv6	561
Troubleshooting EIGRP for IPv6 Neighbor Issues	561
<i>Interface Is Down</i>	561
<i>Mismatched Autonomous System Numbers</i>	562
<i>Mismatched K Values</i>	562
<i>Passive Interfaces</i>	562
<i>Mismatched Authentication</i>	562
<i>Timers</i>	563
<i>Interface Not Participating in Routing Process</i>	563
<i>ACLs</i>	564
Troubleshooting EIGRP for IPv6 Route	564
<i>Interface Not Participating in Routing Process</i>	564
<i>Better Source of Information</i>	565
<i>Route Filtering</i>	565
<i>Stub Configuration</i>	565
<i>Split-horizon</i>	566
EIGRP for IPv6 Trouble Tickets	567
Trouble Ticket 14-4	568
Troubleshooting Named EIGRP Configurations	572
Named EIGRP Verification Commands	573
Named EIGRP Trouble Tickets	577
Trouble Ticket 14-5	577
Exam Preparation Tasks	582
Review All Key Topics	582
Define Key Terms	583
Command Reference to Check Your Memory	583

Chapter 15 Troubleshooting OSPF 587

“Do I Know This Already?” Quiz	587
Foundation Topics	590
Troubleshooting OSPFv2	590
Troubleshooting OSPFv2 Neighbor Adjacencies	590
<i>Interface Is Down</i>	593
<i>Interface Not Running the OSPF Process</i>	593
<i>Mismatched Timers</i>	594
<i>Mismatched Area Numbers</i>	596
<i>Mismatched Area Type</i>	597
<i>Different Subnets</i>	598
<i>Passive Interface</i>	599
<i>Mismatched Authentication Information</i>	600
ACLs	601
MTU Mismatch	602
Duplicate Router IDs	603
Mismatched Network Types	604
Troubleshooting OSPFv2 Routes	606
<i>Interface Not Running the OSPF Process</i>	606
<i>Better Source of Information</i>	607
Route Filtering	611
Stub Area Configuration	613
<i>Interface Is Shut Down</i>	614
<i>Wrong Designated Router Was Elected</i>	615
Duplicate Router IDs	619
Troubleshooting Miscellaneous OSPFv2 Issues	620
<i>Tracking OSPF Advertisements Through a Network</i>	620
Route Summarization	622
Discontiguous Areas	624
Load Balancing	626
Default Route	627
OSPFv2 Trouble Tickets	627
Trouble Ticket 15-1	628
Trouble Ticket 15-2	635
Trouble Ticket 15-3	639
Troubleshooting OSPFv3 for IPv6	641
OSPFv3 Troubleshooting Commands	641

OSPFv3 Trouble Tickets	647
Trouble Ticket 15-4	647
Trouble Ticket 15-5	650
Troubleshoot OSPFv3 Address Families	655
OSPFv3 Address Family Troubleshooting	655
OSPFv3 AF Trouble Tickets	664
Trouble Ticket 15-6	665
Exam Preparation Tasks	669
Review All Key Topics	669
Define Key Terms	670
Complete Tables and Lists from Memory	670
Command Reference to Check Your Memory	671

Chapter 16 Troubleshooting Route Maps and Policy-Based Routing 675

“Do I Know This Already?” Quiz	675
Foundation Topics	678
Troubleshooting Route Maps	678
How to Read a Route Map	678
Troubleshooting Policy-Based Routing	681
PBR	681
Policy-Based Routing Trouble Tickets	684
Trouble Ticket 16-1	685
Trouble Ticket 16-2	689
Trouble Ticket 16-3	691
Exam Preparation Tasks	693
Review All Key Topics	693
Define Key Terms	693
Command Reference to Check Your Memory	693

Chapter 17 Troubleshooting Redistribution 697

“Do I Know This Already?” Quiz	697
Foundation Topics	700
Troubleshooting IPv4 and IPv6 Redistribution	700
Route Redistribution Overview	700
Troubleshooting Redistribution into RIP	703
Troubleshooting Redistribution into EIGRP	706
Troubleshooting Redistribution into OSPF	710
Troubleshooting Redistribution into BGP	715
Troubleshooting Redistribution with Route Maps	718

Redistribution Trouble Tickets	718
Trouble Ticket 17-1	719
Trouble Ticket 17-2	723
Trouble Ticket 17-3	727
Trouble Ticket 17-4	733
Troubleshooting Advanced Redistribution Issues	737
Troubleshooting Suboptimal Routing Caused by Redistribution	737
Troubleshooting Routing Loops Caused by Redistribution	739
Exam Preparation Tasks	745
Review All Key Topics	745
Define Key Terms	745
Command Reference to Check Your Memory	746
Chapter 18 Troubleshooting BGP	749
“Do I Know This Already?” Quiz	749
Foundation Topics	753
Troubleshooting BGP Neighbor Adjacencies	753
Interface Is Down	754
Layer 3 Connectivity Is Broken	754
Path to Neighbor Is via Default Route	755
Neighbor Does Not Have a Route to the Local Router	756
Incorrect neighbor Statement	757
BGP Packets Sourced from Wrong IP Address	758
ACLs	759
TTL of BGP Packet Expires	761
Mismatched Authentication	763
Misconfigured Peer Groups	764
Timers	765
Troubleshooting BGP Routes	766
Missing or Bad network mask Command	768
Next-Hop Router Not Reachable	770
BGP Split-Horizon Rule	772
Better Source of Information	773
Route Filtering	775
Troubleshooting BGP Path Selection	780
Understanding the Best Path Decision-Making Process	781
Private Autonomous System Numbers	784
Using debug Commands	784

Troubleshooting BGP for IPv6	786
BGP Trouble Tickets	790
Trouble Ticket 18-1	791
Trouble Ticket 18-2	796
Trouble Ticket 18-3	802
MP-BGP Trouble Tickets	807
Trouble Ticket 18-4	807
Exam Preparation Tasks	810
Review All Key Topics	810
Define Key Terms	811
Command Reference to Check Your Memory	811

Part IV Troubleshooting Management

Chapter 19 Troubleshooting Management Protocols and Tools 815

“Do I Know This Already?” Quiz	815
Foundation Topics	818
Management Protocols Troubleshooting	818
NTP Troubleshooting	818
Syslog Troubleshooting	821
SNMP Troubleshooting	823
Management Tools Troubleshooting	826
Cisco IOS IPSLA Troubleshooting	827
<i>Object Tracking Troubleshooting</i>	833
<i>SPAN and RSPAN Troubleshooting</i>	835
Management Protocols and Tools Trouble Tickets	837
Trouble Ticket 19-1	838
Exam Preparation Tasks	845
Review All Key Topics	845
Define Key Terms	846
Command Reference to Check Your Memory	846

Chapter 20 Troubleshooting Management Access 851

“Do I Know This Already?” Quiz	851
Foundation Topics	854
Console and vty Access Troubleshooting	854
Console Access Troubleshooting	854

vty Access Troubleshooting	855
<i>Telnet</i>	855
<i>SSH</i>	857
<i>Password Encryption Levels</i>	858
Cisco IOS AAA Troubleshooting	858
Management Access Trouble Tickets	861
Trouble Ticket 20-1	862
Trouble Ticket 20-2	863
Trouble Ticket 20-3	865
Exam Preparation Tasks	868
Review All Key Topics	868
Define Key Terms	868
Command Reference to Check Your Memory	868

Part V Final Preparation

Chapter 21 Additional Trouble Tickets 871

Introduction	871
Trouble Ticket 1	872
Suggested Solution	875
Trouble Ticket 2	876
Suggested Solution	879
Trouble Ticket 3	880
Suggested Solution	882
Trouble Ticket 4	884
Issue 1: Suggested Solution	891
Issue 2: Suggested Solution	897
Issue 3: Suggested Solution	897
Issue 4: Suggested Solution	898
Trouble Ticket 5	901
Suggested Solution	907
Trouble Ticket 6	910
Suggested Solution	916
Trouble Ticket 7	918
Issue 1: Forgotten Enable Secret Password	919
Issue 1: Suggested Solution	919

Issue 2: An exec-timeout Parameter Set Too Low	921
Issue 2: Suggested Solution	921
Issue 3: ACL Misconfiguration	922
Issue 3: Suggested Solution	922
Trouble Ticket 8	923
Suggested Solution	926
Trouble Ticket 9	926
Issue 1: Adjacency Between Routers R1 and R2	927
Issue 1: Suggested Solution	930
Issue 2: Adjacency Between Routers R2 and BB2	930
Issue 2: Suggested Solution	931
Issue 3: Adjacency Between Routers BB1 and BB2	931
Issue 3: Suggested Solution	933
Trouble Ticket 10	934
Issue 1: Router R2 Not Load Balancing Between Routers BB1 and BB2	937
Issue 1: Suggested Solution	937
Issue 2: Backbone Routes Not Being Suppressed	938
Issue 2: Suggested Solution	939

Chapter 22 Final Preparation 943

Tools for Final Preparation	943
Exam Engine and Questions on the CD	943
Install the Exam Engine	944
Activate and Download the Practice Exam	944
Activating Other Exams	945
Premium Edition	945
The Cisco Learning Network	945
Memory Tables	945
Chapter-Ending Review Tools	946
Suggested Plan for Final Review/Study	946
Step 1: Review Key Topics and DIKTA Questions	947
Step 3: Hands-On Practice	947
Step 5: Subnetting Practice	948
Step 6: Use the Exam Engine	948
Summary	949

Part VI Appendixes

Appendix A Answers to the “Do I Know This Already” Quizzes 951

Appendix B TSHOOT Exam Updates 957

Index 960

CD-Only Appendixes and Glossary

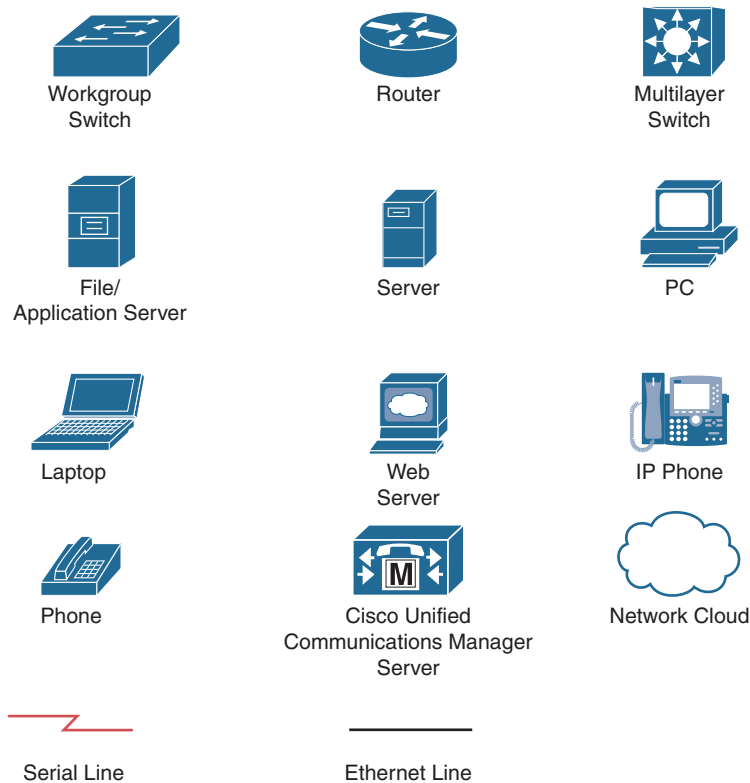
Appendix C Memory Tables

Appendix D Memory Tables Answer Key

Appendix E Study Planner

Glossary

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).
- *Italics* indicate arguments for which you supply actual values.
- Vertical bars (|) separate alternative, mutually exclusive elements.
- Square brackets [] indicate optional elements.
- Braces { } indicate a required choice.
- Braces within brackets [{ }] indicate a required choice within an optional element.

Introduction

Professional certifications have been an important part of the computing industry for many years and will continue to become more important. Many reasons exist for these certifications, but the most popularly cited reason is that of credibility. All other considerations held equal, the certified employee/consultant/job candidate is considered more valuable than one who is not.

Goals and Methods

The most important and somewhat obvious goal of this book is to help you pass the 300-135 Troubleshooting and Maintaining Cisco IP Networks (TSHOOT) exam. In fact, if the primary objective of this book were different, the book's title would be misleading; however, the methods used in this book to help you pass the TSHOOT exam are designed to also make you much more knowledgeable about how to do your job. Although this book and the accompanying CD-ROM have many exam preparation tasks and example test questions, the method in which they are used is not to simply make you memorize as many questions and answers as you possibly can.

The methodology of this book helps you discover the exam topics about which you need more review, fully understand and remember exam topic details, and prove to yourself that you have retained your knowledge of those topics. So, this book helps you pass not by memorization, but by helping you truly learn and understand the topics. The TSHOOT exam is typically your final journey in pursuit of the CCNP Routing and Switching certification, and the knowledge contained within is vitally important to consider yourself a truly skilled routing and switching expert or specialist. This book would do you a disservice if it did not attempt to help you learn the material. To that end, the book can help you pass the TSHOOT exam by using the following methods:

- Covering the exam topics and helping you discover which exam topics you have not mastered
- Providing explanations and information to fill in your knowledge gaps
- Supplying multiple troubleshooting case studies with diagrams and diagnostic output that enhance your ability to resolve trouble tickets presented in the exam environment, in addition to real-world troubleshooting issues you might encounter
- Providing practice exercises on exam topics, presented in each chapter and on the enclosed CD-ROM

Who Should Read This Book?

This book is not designed to be a general networking topics book, although it can be used for that purpose. This book is intended to tremendously increase your chances of passing the Cisco TSHOOT exam. Although other objectives can be achieved from using this book, the book is written with one goal in mind: to help you pass the exam. If you want to pass the exam, this book is for you.

Strategies for Exam Preparation

The strategy you use to prepare for the TSHOOT exam might differ slightly from strategies used by other readers, mainly based on the skills, knowledge, and experience you have already obtained. For example, if you have attended a TSHOOT course, you might take a different approach than someone who learned troubleshooting through on-the-job training. Regardless of the strategy you use or the background you have, this book is designed to help you gain the knowledge you need about the issues that can arise with different routing and switching technologies and get you to the point where you can apply that knowledge and pass the exam.

Cisco Certifications and Exams

Cisco offers four levels of routing and switching certification, each with an increasing level of proficiency: Entry, Associate, Professional, and Expert. These are commonly known by their acronyms CCENT (Cisco Certified Entry Networking Technician), CCNA (Cisco Certified Network Associate) Routing and Switching, CCNP (Cisco Certified Network Professional) Routing and Switching, and CCIE (Cisco Certified Internetworking Expert) Routing and Switching.

For the CCNP Routing and Switching certification, you must pass exams on a series of CCNP topics, including the SWITCH, ROUTE, and TSHOOT exams. For most exams, Cisco does not publish the scores needed for passing. You need to take the exam to find that out for yourself.

To see the most current requirements for the CCNP Routing and Switching certification, go to Cisco.com and click **Training and Events**. There you can find out other exam details such as exam topics and how to register for an exam.

How This Book Is Organized

Although this book can be read cover to cover, it is designed to be flexible and enable you to easily move between chapters to cover only the material that you need more work with. The chapters can be covered in any order, although some chapters are related and build upon each other. If you do intend to read them all, the order in the book is an excellent sequence to use.

Each core chapter covers a subset of the topics on the CCNP TSHOOT exam. The chapters are organized into parts, covering the following topics:

- **Chapter 1, “Introduction to Troubleshooting and Network Maintenance:”** This chapter discusses the importance of having a structured troubleshooting approach and a solid network maintenance plan. It identifies many popular models, structures, and tasks that should be considered by all organizations. However, as you will see, there is no “one-stop shop for all your needs” when it comes to troubleshooting and network maintenance. It is more of an art that you will master over time.

- **Chapter 2, “Troubleshooting and Maintenance Tools:”** This chapter introduces you to a sampling of Cisco IOS tools and features designed for network maintenance and troubleshooting. The tools include ping, Telnet, traceroute, NetFlow, SNMP, SPAN, RSPAN, and CDP.
- **Chapter 3, “Troubleshooting Device Performance:”** This chapter discusses common reasons for high CPU and memory utilization on routers and switches in addition to how you can recognize them. You will examine interface statistics, as they can be an initial indication of some type of issue. You will also review the different types of packet switching modes on routers and multilayer switches.
- **Chapter 4, “Troubleshooting Layer 2 Trunks, VTP, and VLANs:”** This chapter begins by reviewing Layer 2 switch operations and builds from there with discussions on how to troubleshoot issues relating to trunks, VTP, and VLANs. You will also discover how important the information in the MAC address table can be while troubleshooting.
- **Chapter 5, “Troubleshooting STP and Layer 2 EtherChannel:”** This chapter reviews the operation of STP and focuses on troubleshooting STP topology issues such as root bridge selection, root port selection, designated port selection, and finally, the blocked port. You will also examine how to troubleshoot STP features such as PortFast, BPDU Guard, BPDU Filter, Root Guard, Loop Guard, and UDLD. In addition, this chapter reviews how you can combine multiple physical Layer 2 switchports into a logical EtherChannel bundle and how you can troubleshoot issues related to them.
- **Chapter 6, “Troubleshooting Inter-VLAN Routing and Layer 3 EtherChannels:”** This chapter focuses on how you can troubleshoot issues related to different inter-VLAN routing implementations (router-on-a-trunk/stick and SVIs), issues related to routed ports, and issues related to Layer 3 EtherChannels.
- **Chapter 7, “Troubleshooting Switch Security Features:”** This chapter is dedicated to troubleshooting issues related to security features that can be implemented on switches. This includes port security, DHCP snooping, dynamic ARP inspection, IP Source Guard, protected ports, PVLANS, and VACLs. Most of the issues you will experience with these features are configuration based. Therefore, you will focus on the configuration requirements for troubleshooting purposes.
- **Chapter 8, “Troubleshooting First-Hop Redundancy Protocols:”** This chapter discusses the issues that might arise when implementing FHRPs such as HSRP, VRRP, and GLBP. It identifies various elements that could cause these FHRPs not to function as expected and that should be considered while you are troubleshooting. It also provides a collection of commands you can use to successfully troubleshoot issues related to each FHRP.
- **Chapter 9, “Troubleshooting IPv4 Addressing and Addressing Technologies:”** This chapter begins by reviewing IPv4 addressing and how you can identify if addressing is the issue. This is extremely important as you do not want to waste your time troubleshooting a service or feature when the issue is related to the device having an inappropriate IPv4 address, subnet mask, or default gateway. The chapter then covers issues and troubleshooting tasks related to DHCPv4 and NAT.

- **Chapter 10, “Troubleshooting IPv6 Addressing and Addressing Technologies:”** This chapter covers how an IPv6-enabled device determines whether the destination is local or remote. You will also learn how MAC addresses are determined for known IPv6 address, and you will explore the various options for address assignment such as SLAAC and DHCPv6, and what to look for while troubleshooting IPv6-related issues.
- **Chapter 11, “Troubleshooting IPv4 and IPv6 ACLs and Prefix Lists:”** This chapter covers the ins and outs of ACLs and prefix lists. You will learn the way they are processed, how they are read, and how you can identify issues related to them. In addition, this chapter explains how you can use ACLs for traffic filtering and how a prefix list can be used for route filtering.
- **Chapter 12, “Troubleshooting Basic IPv4/IPv6 Routing and GRE Tunnels:”** This chapter covers the packet-delivery process and the various commands that enable you to troubleshoot issues related to the process. You will learn how a router chooses which sources of routing information are more believable so that only the best routes are in the routing table. You will also learn how to recognize and troubleshoot issues related to static routing and GRE tunnels.
- **Chapter 13, “Troubleshooting RIPv2 and RIPv6:”** This chapter focuses on the issues that you may have to troubleshoot in a RIPv2 and RIPv6 domain. This includes how you would recognize the issues based on the presented symptoms and the commands you would use to successfully verify the reason why the issue exists.
- **Chapter 14, “Troubleshooting EIGRP:”** This chapter covers troubleshooting of both EIGRP for IPv4 and EIGRP for IPv6. It breaks out the troubleshooting discussions into two different parts: troubleshooting neighbor adjacencies and troubleshooting missing routes. It also covers the troubleshooting of various issues that are not directly related to neighborships or routes that might arise with EIGRP. To wrap up the chapter, named EIGRP troubleshooting is covered.
- **Chapter 15, “Troubleshooting OSPF:”** This chapter covers troubleshooting of both OSPFv2 and OSPFv3. It breaks out the troubleshooting discussions into two different parts: troubleshooting neighbor adjacencies and troubleshooting missing routes. It also covers the troubleshooting of various issues that are not directly related to neighborships or routes that might arise with OSPF. To wrap up the chapter, OSPFv3 address family troubleshooting is covered.
- **Chapter 16, “Troubleshooting Route Maps and Policy-Based Routing:”** This chapter begins by examining route maps. It gives you the opportunity to review how route maps are read and the commands that you can use to verify a route map’s configuration. The rest of the chapter is dedicated to PBR, which allows you to override the router’s default routing behavior. Therefore, you will discover what could cause PBR not to behave as expected and how you can troubleshoot it.

- **Chapter 17, “Troubleshooting Redistribution:”** This chapter explores the differences of redistributing into EIGRP, OSPF, RIP, and BGP for both IPv4 and IPv6. You will learn what to look out for while troubleshooting so that you can quickly solve any issues related to redistribution. In addition, you will examine what could occur in environments that have multiple points of redistribution and how you can identify the issues and solve them.
- **Chapter 18, “Troubleshooting BGP:”** This chapter examines the various issues that you may face when trying to establish an IPv4 and IPv6 eBGP and iBGP neighbor adjacency and how you can identify them and troubleshoot them. You will also examine the issues that may arise when exchanging IPv4 and IPv6 eBGP and iBGP routes and how you can recognize them and troubleshoot them successfully. You also need to be very familiar with the decision-making process that BGP uses to be an efficient troubleshooter. Therefore, you will spend time exploring this process in the chapter as well.
- **Chapter 19, “Troubleshooting Management Protocols and Tools:”** This chapter covers the issues you might encounter with management protocols such as NTP, syslog, and SNMP. It also covers the issues that you might encounter with management tools, such as Cisco IOS IP SLA, Object Tracking, SPAN, and RSPAN.
- **Chapter 20, “Troubleshooting Management Access:”** This chapter examines the different reasons why access to the console and vty lines might fail, and how you can identify them. In addition you will explore the issues that may arise when using Cisco IOS AAA authentication.
- **Chapter 21, “Additional Trouble Tickets:”** This chapter is dedicated to showing you an additional ten trouble tickets and the various approaches that you can take to solve the problems that are presented.
- **Chapter 22, “Final Preparation:”** This chapter identifies tools for final exam preparation and helps you develop an effective study plan.
- **Appendix A, “Answers to the ‘Do I Know This Already?’ Quizzes:”** This appendix has the answers to the “Do I Know This Already” quizzes, and Appendix B, “TSHOOT Exam Updates,” tells you how to find any updates should there be changes to the exam.

Each chapter in the book uses several features to help you make the best use of your time in that chapter. The features are as follows:

- **Assessment:** Each chapter begins with a “Do I Know This Already?” quiz that helps you determine the amount of time you need to spend studying each topic of the chapter. If you intend to read the entire chapter, you can save the quiz for later use. Questions are all multiple-choice, to give a quick assessment of your knowledge.
- **Foundation Topics:** This is the core section of each chapter that explains the protocols, concepts, configuration, and troubleshooting strategies for the topics in the chapter.

- **Exam Preparation Tasks:** At the end of each chapter, this section collects key topics, references to memory table exercises to be completed as memorization practice, key terms to define, and a command reference that summarizes any relevant commands presented in the chapter.

Finally, the companion CD-ROM contains practice CCNP Routing and Switching TSHOOT questions to reinforce your understanding of the book's concepts. Be aware that the TSHOOT exam will primarily be made up of trouble tickets you need to resolve. Mastery of the topics covered by the CD-based questions, however, will help equip you with the tools needed to effectively troubleshoot the trouble tickets presented on the exam.

The CD also contains the Memory Table exercises and answer keys as well as over 60mins of video walking you through an exam strategy.

CCNP TSHOOT Exam Topics

Carefully consider the exam topics Cisco has posted on its website as you study, particularly for clues to how deeply you should know each topic. Also, you can develop a broader knowledge of the subject matter by reading and studying the topics presented in this book. Remember that it is in your best interest to become proficient in each of the CCNP Routing and Switching subjects. When it is time to use what you have learned, being well rounded counts more than being well tested.

Table I-1 shows the official exam topics for the TSHOOT exam, as posted on Cisco.com. Note that Cisco has occasionally changed exam topics without changing the exam number, so do not be alarmed if small changes in the exam topics occur over time. Also, it is possible to receive questions on the exam that are not related to any of the exam topics listed. Cisco indicates this when you view the exam topics on their website. Therefore, to ensure that you are well prepared for the exam, we have covered the exam topics as well as any additional topics that we considered to be necessary for your success. For example, there is no mention of Layer 2 security, inter-VLAN routing, or FHRPs in the exam objectives. However, we have included chapters dedicated to these to make sure that you are well prepared.

Table I-1 *CCNP TSHOOT Exam Topics*

Exam Topics	Chapters Where Exam Topics Are Covered
1.0 Network Principles	
Debug, conditional debug	Chapters 1 and 2
Ping and trace route with extended options	
Diagnose the root cause of networking issues (analyze symptoms, identify and describe root cause)	
Design and implement valid solutions	
Verify and monitor resolution	

Exam Topics	Chapters Where Exam Topics Are Covered
2.0 Layer 2 Technologies	
Troubleshooting switch administration	Chapters 4, 5, 19
Troubleshooting Layer 2 protocols	
Troubleshoot VLANs	
Troubleshoot trunking	
Troubleshoot EtherChannels	
Troubleshoot spanning tree	
Troubleshoot other LAN switching technologies	
Troubleshoot chassis virtualization and aggregation technologies	
3.0 Layer 3 Technologies	
Troubleshooting IPv4 addressing and subnetting	Chapters 9, 10, 12–18
Troubleshoot IPv6 addressing and subnetting	
Troubleshoot static routing	
Troubleshoot default routing	
Troubleshoot administrative distance	
Troubleshoot passive interfaces	
Troubleshoot VRF lite	
Troubleshoot filter with any protocol	
Troubleshoot between any routing protocols or routing sources	
Troubleshoot manual and autosummarization with any routing protocol	
Troubleshoot policy-based routing	
Troubleshoot suboptimal routing	
Troubleshoot loop prevention mechanisms	
Troubleshoot RIPv2	
Troubleshoot EIGRP neighbor relationship and authentication	

Exam Topics	Chapters Where Exam Topics Are Covered
Troubleshoot loop free path selection	Chapters 9, 10, 12–18
Troubleshoot EIGRP operations	
Troubleshoot EIGRP stubs	
Troubleshoot EIGRP load balancing	
Troubleshoot EIGRP metrics	
Troubleshoot OSPF neighbor relationship and authentication	
Troubleshoot network types, area types, and router types	
Troubleshoot OSPF path preference	
Troubleshoot OSPF operations	
Troubleshoot OSPF for IPv6	
Troubleshoot BGP peer relationships and authentication	
Troubleshoot eBGP	
4.0 VPN Technologies	
Troubleshoot GRE	Chapter 12
5.0 Infrastructure Security	
Troubleshoot IOS AAA using local database	Chapters 11 and 20
Troubleshoot device access control	
Troubleshoot router security features	
6.0 Infrastructure Services	
Troubleshoot device Management	Chapters 2, 9, 10, and 19
Troubleshoot SNMP	
Troubleshoot logging	
Troubleshoot Network Time Protocol (NTP)	
Troubleshoot IPv4 and IPv6 DHCP	
Troubleshoot IPv4 Network Address Translation (NAT)	
Troubleshoot SLA architecture	
Troubleshoot tracking objects	



This chapter covers the following topics:

- **Introduction to Troubleshooting:** This section introduces you to troubleshooting and then focuses on a structured troubleshooting approach. It also provides you with some common steps to help you be more efficient.
- **Popular Troubleshooting Methods:** This section introduces you to various troubleshooting methods that can assist in narrowing your focus during your troubleshooting efforts.
- **Introduction to Network Maintenance:** This section introduces you to maintenance tasks and identifies a few well-known network maintenance models that you can adopt.
- **Common Maintenance Procedures:** This section reviews the common network maintenance tasks that all organizations should perform.
- **The Troubleshooting and Network Maintenance Relationship:** This section identifies the importance of aligning maintenance tasks with troubleshooting goals.

Introduction to Troubleshooting and Network Maintenance

Business operations, without a doubt, depend on the reliable operation of data networks (which might also carry voice and video traffic). This statement holds true regardless of the business size. A structured and systematic maintenance approach significantly contributes to the uptime for all networks. In addition, having a sound troubleshooting methodology in place helps ensure that when issues arise you are confident and ready to fix them.

Consider a vehicle as an example. Regular maintenance such as oil changes, joint lubrication, and fluid top-offs are performed on a vehicle to ensure that problems do not arise and the life of that vehicle is maximized. However, if an issue does arise, it is taken to a mechanic so that they may troubleshoot the issue using a structured troubleshooting process and ultimately fix the vehicle. Similarly, the number of issues in a network can be reduced by following a maintenance plan, and troubleshooting can be more effective with a structured approach in place.

This chapter discusses the importance of having a structured troubleshooting approach and a solid network maintenance plan. It identifies many popular models, structures, and tasks that should be considered by all organizations. However, as you will see, there is no “one-stop shop for all your needs” when it comes to troubleshooting and network maintenance. It is more of an art that you will master over time.

“Do I Know This Already?” Quiz

The “Do I Know This Already?” quiz allows you to assess whether you should read this entire chapter thoroughly or jump to the “Exam Preparation Tasks” section. If you are in doubt about your answers to these questions or your own assessment of your knowledge of the topics, read the entire chapter. Table 1-1 lists the major headings in this chapter and their corresponding “Do I Know This Already?” quiz questions. You can find the answers in Appendix A, “Answers to the ‘Do I Know This Already?’ Quizzes.”

Table 1-1 “Do I Know This Already?” Section-to-Question Mapping

Foundation Topics Section	Questions
Introduction to Troubleshooting	1–7
Popular Troubleshooting Methods	8–9
Introduction to Network Maintenance	10–12

Foundation Topics Section	Questions
Identifying Common Maintenance Procedures	13–16
The Troubleshooting and Network Maintenance Relationship	17–20

Caution The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you do not know the answer to a question or are only partially sure of the answer, you should mark that question as wrong for purposes of the self-assessment. Giving yourself credit for an answer that you correctly guess skews your self-assessment results and might provide you with a false sense of security.

1. Identify the three steps in a simplified troubleshooting model.
 - a. Problem replication
 - b. Problem diagnosis
 - c. Problem resolution
 - d. Problem report
2. Which of the following is the best statement to include in a problem report?
 - a. The network is broken.
 - b. User A cannot reach the network.
 - c. User B recently changed his PC's operating system to Microsoft Windows 7.
 - d. User C is unable to attach to an internal share resource of \\10.1.1.1\Budget, although he can print to all network printers, and he can reach the Internet.
3. What troubleshooting step should you perform after a problem has been reported and clearly defined?
 - a. Propose an hypothesis
 - b. Collect information
 - c. Eliminate potential causes
 - d. Examine collected information
4. What are the two primary goals of troubleshooters as they are collecting information?
 - a. Eliminate potential causes from consideration
 - b. Identify indicators pointing to the underlying cause of the problem
 - c. Propose an hypothesis about what is most likely causing the problem
 - d. Find evidence that can be used to eliminate potential causes

5. When performing the “eliminate potential causes” troubleshooting step, which caution should the troubleshooter be aware of?
 - a. The danger of drawing an invalid conclusion from the observed data
 - b. The danger of troubleshooting a network component over which the troubleshooter does not have authority
 - c. The danger of causing disruptions in workflow by implementing the proposed solution
 - d. The danger of creating a new problem by implementing the proposed solution
6. A troubleshooter is hypothesizing a cause for an urgent problem, and her hypothesis involves a network device that she is not authorized to configure. The person who is authorized to configure the network device is unavailable. What should the troubleshooter do?
 - a. Wait for authorized personnel to address the issue.
 - b. Attempt to find a temporary workaround for the issue.
 - c. Override corporate policy, based on the urgency, and configure the network device independently because authorized personnel are not currently available.
 - d. Instruct the user to report the problem to the proper department that is authorized to resolve the issue.
7. Experienced troubleshooters with in-depth comprehension of a particular network might skip the examine information and eliminate potential causes steps in a structured troubleshooting model, instead relying on their own insight to determine the most likely cause of a problem. This illustrates what approach to network troubleshooting?
 - a. Ad hoc
 - b. Shoot from the hip
 - c. Crystal ball
 - d. Independent path
8. Which of the following troubleshooting models requires access to a specific application?
 - a. Bottom-up
 - b. Divide-and-conquer
 - c. Comparing configurations
 - d. Top-down

- 9.** Based on your analysis of a problem report and the data collected, you want to use a troubleshooting model that can quickly eliminate multiple layers of the OSI model as potential sources of the reported problem. Which of the following troubleshooting methods would be most appropriate?
- a.** Following the traffic path
 - b.** Bottom-up
 - c.** Divide-and-conquer
 - d.** Component swapping
- 10.** Which of the following are considered network maintenance tasks? (Choose the three best answers.)
- a.** Troubleshooting problem reports
 - b.** Attending training on emerging network technologies
 - c.** Planning for network expansion
 - d.** Hardware installation
- 11.** Network maintenance tasks can be categorized into one of which two categories?
- a.** Recovery tasks
 - b.** Interrupt-driven tasks
 - c.** Structured tasks
 - d.** Installation tasks
- 12.** Which letter in the FCAPS acronym represents the maintenance area responsible for billing end users?
- a.** F
 - b.** C
 - c.** A
 - d.** P
 - e.** S

- 13.** The lists of tasks required to maintain a network can vary widely, depending on the goals and characteristics of that network. However, some network maintenance tasks are common to most networks. Which of the following would be considered a common task that should be present in any network maintenance model?
- a.** Performing database synchronization for a network's Microsoft Active Directory
 - b.** Making sure that digital certificates used for PKI are renewed in advance of their expiration
 - c.** Using Cisco Prime to dynamically discover network device changes
 - d.** Performing scheduled backups
- 14.** Which of the following statements is true about scheduled maintenance?
- a.** Scheduled maintenance helps ensure that important maintenance tasks are not overlooked.
 - b.** Scheduled maintenance is not recommended for larger networks, because of the diversity of maintenance needs.
 - c.** Maintenance tasks should only be performed based on a scheduled maintenance schedule, to reduce unexpected workflow interruptions.
 - d.** Scheduled maintenance is more of a reactive approach to network maintenance, as opposed to a proactive approach.
- 15.** Which of the following questions are appropriate when defining your change management policies?
- a.** What version of operating system is currently running on the device to be upgraded?
 - b.** What is the return on investment (ROI) of an upgrade?
 - c.** What measureable criteria determine the success or failure of a network change?
 - d.** Who is responsible for authorizing various types of network changes?
- 16.** Which three of the following components would you expect to find in a set of network documentation?
- a.** Logical topology diagram
 - b.** Listing of interconnections
 - c.** Copy of IOS image
 - d.** IP address assignments

- 17.** What is the ideal relationship between network maintenance and troubleshooting?
- a.** Networking maintenance and troubleshooting efforts should be isolated from one another.
 - b.** Networking maintenance and troubleshooting efforts should complement one another.
 - c.** Networking maintenance and troubleshooting efforts should be conducted by different personnel.
 - d.** Networking maintenance is a subset of network troubleshooting.
- 18.** Which three of the following suggestions can best help troubleshooters keep in mind the need to document their steps?
- a.** Require documentation
 - b.** Keep documentation in a hidden folder
 - c.** Schedule documentation checks
 - d.** Automate documentation
- 19.** Which three troubleshooting phases require clear communication with end users?
- a.** Problem report
 - b.** Information collection
 - c.** Hypothesis verification
 - d.** Problem resolution
- 20.** What are two elements of a change management system?
- a.** Determine when changes can be made
 - b.** Determine potential causes for the problem requiring the change
 - c.** Determine who can authorize a change
 - d.** Determine what change should be made

Foundation Topics

Introduction to Troubleshooting

Troubleshooting is a skill, and like all skills, you will get better at it the more you have to perform it. The more troubleshooting situations you are placed in, the more your skills will improve, and as a result of this, the more your confidence will grow. However, don't start wishing for issues to happen in your organization just so that you can get more experience. Although there is no right or wrong way to troubleshoot, there is definitely a more efficient and effective way to troubleshoot that all experienced troubleshooters follow. This section begins by introducing you to troubleshooting. It then focuses on a structured troubleshooting approach that provides you with some common methods to enhance your efficiency.

Defining Troubleshooting

Troubleshooting at its essence is the process of responding to a problem report (sometimes in the form of a trouble ticket), diagnosing the underlying cause of the problem, and resolving the problem. Although you normally think of the troubleshooting process as beginning when a user reports an issue, you need to understand that through effective network monitoring you may detect a situation that could become a troubleshooting issue and resolve that situation before it impacts users.

After an issue is reported, the first step toward resolution is clearly defining the issue. When you have a clearly defined troubleshooting target, you can begin gathering further information related to it. From this information, you should be able to better define the issue. Then based on your diagnosis, you can propose an hypothesis about what is most likely causing the issue. Then the evaluation of these likely causes leads to the identification of the suspected underlying root cause of the issue.

After you identify a suspected underlying cause, you next define approaches to resolving the issue and select what you consider to be the best approach. Sometimes the best approach to resolving an issue cannot be implemented immediately. For example, a piece of equipment might need replacing, or a business's workflow might be disrupted by implementing such an approach during working hours. In such situations, a troubleshooter might use a temporary fix until a permanent fix can be put in place.

Let's look at an example. It is 3:00 p.m. at a luxury hotel in Las Vegas. On this day, the hotel cannot register guests or create the keycards needed for guest rooms. After following the documented troubleshooting procedures, the network team discovers that Spanning Tree Protocol (STP) has failed on a Cisco Catalyst switch, resulting in a Layer 2 topological loop. Thus, the network is being flooded with traffic, preventing registrations and keycards from being completed because the server is not accessible. The network team now has to decide on the best course of action at this point. The permanent fix of replacing the failed equipment immediately would disrupt the network further and take a considerable amount of time, thus delaying the guest registrations further. A temporary

fix would be to disconnect the redundant links involved in the loop so that the Layer 2 loop is broken and guests can be registered at that point. When the impact on guests and guest services is minimal, the network team can implement the permanent fix. Consider Figure 1-1, which depicts a simplified model of the troubleshooting steps previously described.



Figure 1-1 *Simplified Troubleshooting Flow*



This simplified model consists of three steps:

- Step 1.** Problem report
- Step 2.** Problem diagnosis
- Step 3.** Problem resolution

Of these three steps, most of a troubleshooter’s efforts are spent in the problem diagnosis step. For example, your child reports that the toaster won’t work. That is the problem report step. You have it clarified further, and your child indicates that the toaster does not get hot. So, you decide to take a look at the toaster and diagnose it. This is the problem diagnosis step, which is broken up into multiple subcomponents. Table 1-2 describes key components of this problem diagnosis step.



Table 1-2 *Steps to Diagnose a Problem*

Step	Description
Collect information	Because a typical problem report lacks sufficient information to give a troubleshooter insight into a problem’s underlying cause, the troubleshooter should collect additional information, perhaps using network maintenance tools or by interviewing impacted users.
Examine collected information	After collecting sufficient information about a problem, the troubleshooter then examines that information, perhaps comparing the information against previously collected baseline information.
Eliminate potential causes	Based on the troubleshooter’s knowledge of the network and his interrogation of collected information, he can begin to eliminate potential causes for the problem.
Propose an hypothesis	After the troubleshooter eliminates multiple potential causes for the problem, he is left with one or more causes that are more likely to have resulted in the problem. The troubleshooter hypothesizes what he considers to be the most likely cause of the problem.
Verify hypothesis	The troubleshooter then tests his hypothesis to confirm or refute his theory about the problem’s underlying cause.

After collecting, examining, and eliminating, you hypothesize that the power cable for the toaster is not plugged in. You test your hypothesis, and it is correct. Problem solved. This was a simple example, but even with a toaster, you spent the majority of your time diagnosing the problem. Once you determined that there was no electricity to the toaster, you had to figure out whether it was plugged in. If it was plugged in, you then had to consider whether the wall outlet was damaged, or the circuit breaker was off, or the toaster was too old and it broke. All of your effort focused on the problem diagnosis step.

By combining the three main steps with the five substeps, you get the following structured troubleshooting procedure:



- Step 1.** Problem report
- Step 2.** Collect information
- Step 3.** Examine collected information
- Step 4.** Eliminate potential causes
- Step 5.** Propose an hypothesis
- Step 6.** Verify hypothesis
- Step 7.** Problem resolution

The Value of Structured Troubleshooting

Troubleshooting skills vary from administrator to administrator, and as mentioned earlier, your skills as a troubleshooter will get better with experience. However, as a troubleshooter, your primary goal is to be efficient. Being fast comes with experience, but it is not worth much if you are not efficient. To be efficient, you need to follow a structured troubleshooting method. A structured troubleshooting method might look like the approach depicted in Figure 1-2.

If you do not follow a structured approach, you might find yourself moving around troubleshooting tasks in a fairly random way based on instinct. Although in one instance you might be fast at solving the issue, in the next instance you end up taking an unacceptable amount of time. In addition, it can become confusing to remember what you have tried and what you have not. Eventually, you find yourself repeating solutions you have already tried, hoping it works. Also, if another administrator comes to assist you, communicating to that administrator the steps you have already gone through becomes a challenge. Therefore, following a structured troubleshooting approach helps you reduce the possibility of trying the same resolution more than once and inadvertently skipping a task. It also aids in communicating to someone else possibilities that you have already eliminated.

With experience, you will start to see similar issues. In addition, you should have exceptional documentation on past network issues and the steps used to solve them. In such instances, spending time methodically examining information and eliminating potential causes might actually be less efficient than immediately hypothesizing a cause after you collect information about the problem and review past documents. This method, illustrated in Figure 1-3, is often called the *shoot from the hip method*.

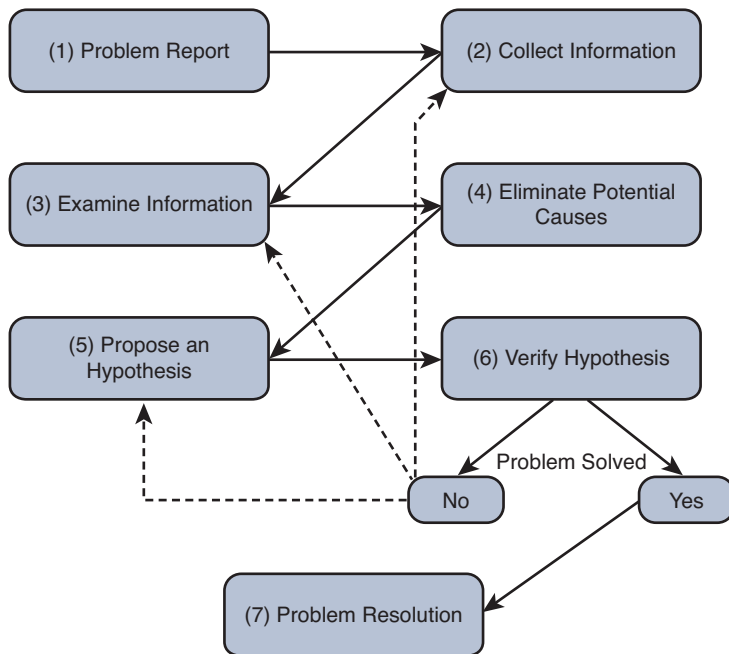


Figure 1-2 *Example of a Structured Troubleshooting Approach*

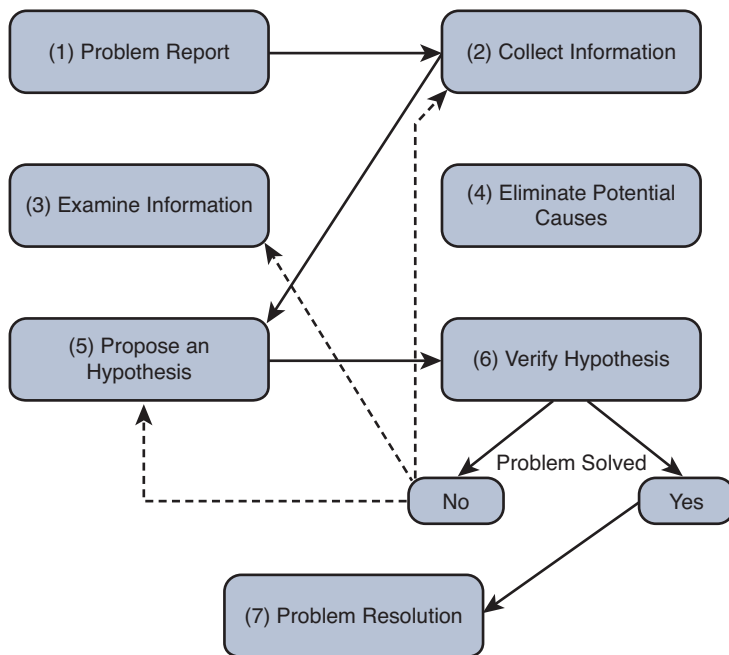


Figure 1-3 *Example of a Shoot from the Hip Troubleshooting Approach*

The danger with the shoot from the hip method is that if your instincts are incorrect, and the problem is not solved, you waste valuable time. Therefore, you need to be able to revert back to the structured troubleshooting approach as needed and examine all collected information.



A Structured Approach

No single collection of troubleshooting procedures is capable of addressing all conceivable network issues because there are too many variables (for example, user actions). However, having a structured troubleshooting approach helps ensure that the organization's troubleshooting efforts are following a similar flow each time an issue arises no matter who is assigned the task. This will allow one troubleshooter to more efficiently take over for or assist another troubleshooter if required.

This section examines each step in a structured approach in more detail as shown in Figure 1-4.

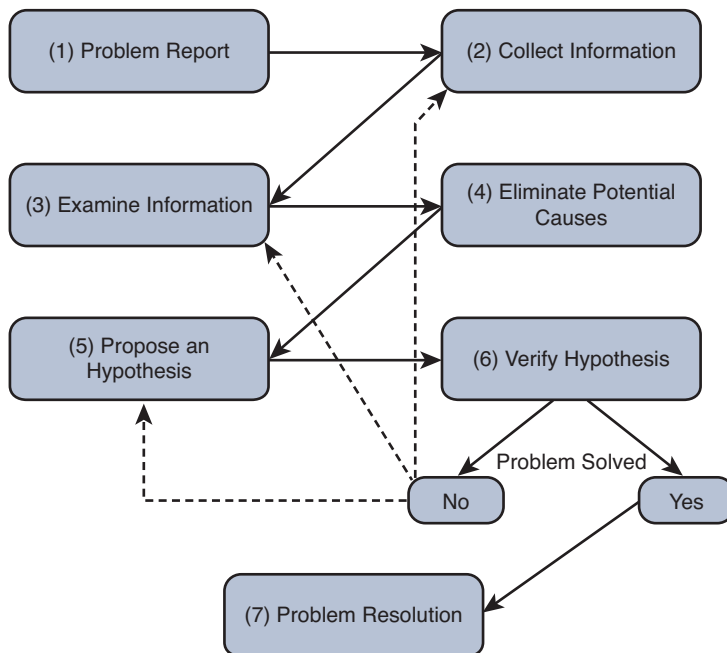


Figure 1-4 A Structured Troubleshooting Approach

1. Problem Report

A problem report from a user often lacks sufficient detail for you to take that problem report and move on to the next troubleshooting process (that is, collect information). For example, a user might report, “The network is broken.” If you receive such a vague report, you probably need to contact the user and ask him exactly what aspect of the network is not functioning correctly.

After your interview with the user, you should be able to construct a more detailed problem report that includes statements such as, when the user does *X*, she observes *Y*. For example, “When the user attempts to connect to a website on the Internet, her browser reports a 404 error. However, the user can successfully navigate to websites on her company’s intranet.” Or, “When the user attempts to connect to an FTP site using a web browser, the web browser reports the page can’t be displayed.”

After you have a clear understanding of the issue, you might need to determine who is responsible for working on the hardware or software associated with that issue. For example, perhaps your organization has one IT group tasked with managing switches and another IT group charged with managing routers. Therefore, as the initial point of contact, you might need to decide whether this issue is one you are authorized to address or if you need to forward the issue to someone else who is authorized. If you are not sure at this point, start collecting information so that the picture can become clearer, and be mindful that you might have to pass this information on to another member of your IT group at some point, so accurate documentation is important.

2. Collect Information

When you are in possession of a clear problem report, the next step is gathering relevant information pertaining to the problem, as shown in Figure 1-5.



Figure 1-5 *A Structured Troubleshooting Approach (Collect Information)*

Efficiently and effectively gathering information involves focusing information gathering efforts on appropriate network entities (for example, routers, servers, switches, or clients) from which information should be collected. Otherwise, the troubleshooter could waste time wading through reams of irrelevant data. For example, to be efficient and effective, the troubleshooter needs to understand what is required to access the resources the end user is unable to access. With our FTP site problem report, the FTP resources are accessible through an FTP client. Troubleshooters not aware of that might spend hours collecting irrelevant data with **debug**, **show**, **ping**, and **tracert** commands, when all they had to do was point the user to the FTP client installed on the client’s computer.

In addition, perhaps a troubleshooter is using a troubleshooting model that follows the path of the affected traffic (as discussed in the “Popular Troubleshooting Methods” section of this chapter), and information needs to be collected from a network device over which the troubleshooter has no access. At that point, the troubleshooter might need to work with appropriate personnel who have access to that device. Alternatively, the troubleshooter might switch troubleshooting models. For example, instead of following the traffic’s path, the troubleshooter might swap components or use a bottom-up troubleshooting model.

3. Examine Collected Information

After collecting information about the problem report (for example, collecting output from **show** or **debug** commands, performing packet captures, using **ping**, or **tracert**), the next structured troubleshooting step is to analyze the collected information as shown in Figure 1-6.

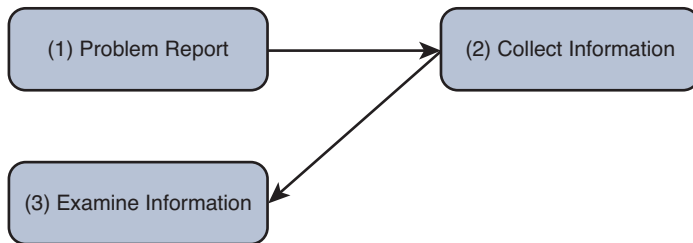


Figure 1-6 A Structured Troubleshooting Approach (Examine Information)

A troubleshooter has two primary goals while examining the collected information:

- Identify indicators pointing to the underlying cause of the problem
- Find evidence that can be used to eliminate potential causes

To achieve these two goals, the troubleshooter attempts to find a balance between two questions:

- What *is* occurring on the network?
- What *should* be occurring on the network?

The delta between the responses to these questions might give the troubleshooter insight into the underlying cause of a reported problem. A challenge, however, is for the troubleshooter to know what currently should be occurring on the network.

If the troubleshooter is experienced with the applications and protocols being examined, the troubleshooter might be able to determine what is occurring on the network and how that differs from what should be occurring. However, if the troubleshooter lacks knowledge of specific protocol behavior, she still might be able to effectively examine the collected information by contrasting that information with baseline data or documentation.

Baseline data might contain, for example, the output of **show** and **debug** commands issued on routers when the network was functioning properly. By contrasting this baseline data with data collected after a problem occurred, even an inexperienced troubleshooter might be able to see the difference between the data sets, thus providing a clue as to the underlying cause of the problem under investigation. This implies that as part of a routine network maintenance plan, baseline data should periodically be collected when the network is functioning properly.

Documentation plays an extremely important role at this point. Accurate and up-to-date documentation can assist a troubleshooter in examining the collected data to determine whether anything has changed in relation to the setup or configuration. Going back to

the FTP example, if the troubleshooter was not aware that an FTP client was required, a quick review of the documentation related to FTP connectivity would indicate so. This would allow the troubleshooter to move on to the next step.

4. Eliminate Potential Causes

Following an examination of collected data, a troubleshooter can start to form conclusions based on that data. Some conclusions might suggest a potential cause for the problem, whereas other conclusions eliminate certain causes from consideration (see Figure 1-7).

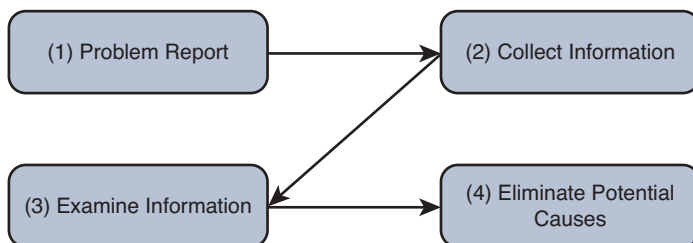


Figure 1-7 *A Structured Troubleshooting Approach (Eliminate Potential Causes)*

It is imperative that you not jump to conclusions at this point. Jumping to conclusions can make you less efficient as a troubleshooter as you start formulating hypotheses based on a small fraction of collected data, which leads to more work and slower overall response times to problems. As an example, a troubleshooter might jump to a conclusion based on the following scenario, which results in wasted time:

A problem report indicates that PC A cannot communicate with server A, as shown in Figure 1-8. The troubleshooter is using a troubleshooting method that follows the path of traffic through the network. The troubleshooter examines output from the **show cdp neighbor** command on routers R1 and R2. Because those routers do not recognize each other as Cisco Discovery Protocol (CDP) neighbors, the troubleshooter leaps to the conclusion that Layer 2 and Layer 1 connectivity is down between R1 and R2. The troubleshooter then runs to the physical routers to verify physical connectivity, only to see that all is fine. Reviewing further output and documentation indicates that CDP is disabled on R1 and R2 interfaces for security reasons. Therefore, the output of **show cdp neighbors** alone is insufficient to conclude that Layer 2 and 1 connectivity was the problem.

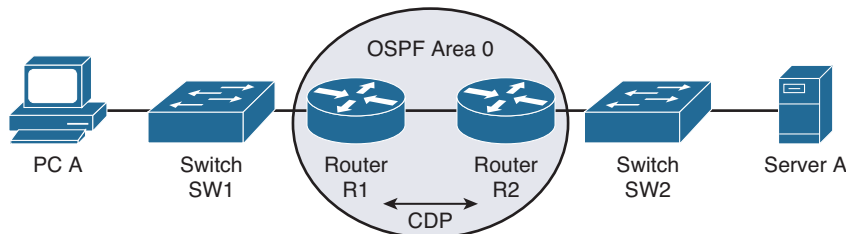


Figure 1-8 *Scenario Topology*

On another note, a caution to be observed when drawing conclusions is not to read more into the data than what is actually there. As an example, a troubleshooter might reach a faulty conclusion based on the following scenario:

A problem report indicates that PC A cannot communicate with server A, as shown in Figure 1-8. The troubleshooter is using a troubleshooting method that follows the path of traffic through the network. The troubleshooter examines output from the **show cdp neighbor** command on routers R1 and R2. Because those routers recognize each other as Cisco Discovery Protocol (CDP) neighbors, the troubleshooter leaps to the conclusion that these two routers see each other as Open Shortest Path First (OSPF) neighbors and have mutually formed OSPF adjacencies. However, the **show cdp neighbor** output is insufficient to conclude that OSPF adjacencies have been formed between routers R1 and R2.

In addition, if time permits, explaining the rationale for your conclusions to a coworker can often help reveal faulty conclusions. As shown by the previous examples, continuing your troubleshooting efforts based on a faulty conclusion can dramatically increase the time required to resolve a problem.

5. Propose an Hypothesis

By eliminating potential causes of a reported problem, as described in the previous process, troubleshooters should be left with one or a few potential causes that they can focus on. At this point, troubleshooters should rank the potential causes from most likely to least likely. Troubleshooters should then focus on the cause they believe is most likely to be the underlying one for the reported problem and propose an hypothesis, as shown in Figure 1-9.

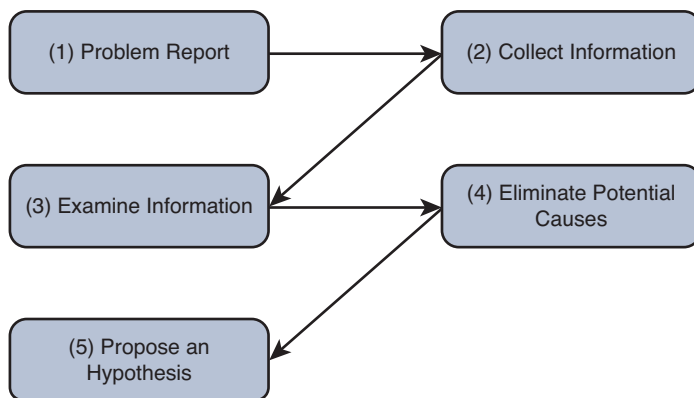


Figure 1-9 *A Structured Troubleshooting Approach (Propose an Hypothesis)*

After proposing an hypothesis, troubleshooters might realize that they are not authorized to access a network device that needs to be accessed to resolve the problem report. In such a situation, a troubleshooter needs to assess whether the problem can wait until authorized personnel have an opportunity to resolve the issue. If the problem is urgent and no authorized administrator is currently available, the troubleshooter might attempt

to at least alleviate the symptoms of the problem by creating a temporary workaround. Although this approach does not solve the underlying cause, it might help business operations continue until the main cause of the problem can be appropriately addressed.

6. Verify Hypothesis

After troubleshooters propose what they believe to be the most likely cause of a problem, they need to develop a plan to address the suspected cause and implement it. Alternatively, if troubleshooters decide to implement a workaround, they need to come up with a plan and implement it while noting that a permanent solution is still needed. However, implementing a plan that resolves a network issue often causes temporary network outages for other users or services. Therefore, the troubleshooter must balance the urgency of the problem with the potential overall loss of productivity, which ultimately affects the financial bottom line. There should be a change management procedure in place that helps the troubleshooter determine the most appropriate time to make changes to the production network and the steps required to do so. If the impact on workflow outweighs the urgency of the problem, the troubleshooter might wait until after business hours to execute the plan.

A key (and you should make it mandatory) component in implementing a problem solution is to have the steps documented. Not only does a documented list of steps help ensure the troubleshooter does not skip any, but such a document can serve as a rollback plan if the implemented solution fails to resolve the problem. Therefore, if the problem is not resolved after the troubleshooter implements the plan, or if the execution of the plan resulted in one or more additional problems, the troubleshooter should execute the rollback plan. After the network is returned to its previous state (that is, the state prior to deploying the proposed solution); the troubleshooter can then reevaluate her hypothesis.

Although the troubleshooter might have successfully identified the underlying cause, perhaps the solution failed to resolve that cause. In that case, the troubleshooter could create a different plan to address that cause. Alternatively, if the troubleshooter had identified other causes and ranked them during the propose an hypothesis step, she can focus her attention on the next most likely cause and create an action plan to resolve that cause and implement it.

This process can be repeated until the troubleshooter has exhausted the list of potential causes or is unable to collect information that can point to other causes, as shown in Figure 1-10. At that point, a troubleshooter might need to gather additional information or enlist the aid of a coworker or the Cisco Technical Assistance Center (TAC).

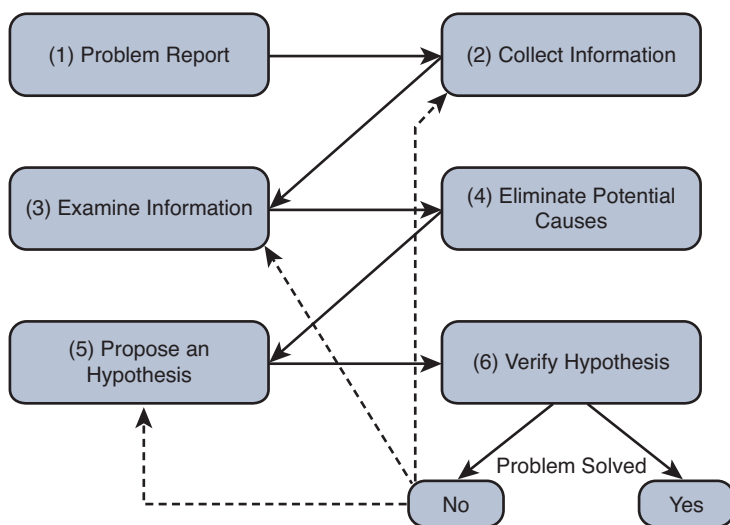


Figure 1-10 *A Structured Troubleshooting Approach (Verify Hypothesis)*

7. Problem Resolution

This is the final step of the structured approach, as shown in Figure 1-11. Although this is one of the most important steps, it is often forgotten or overlooked. After the reported problem is resolved, the troubleshooter should make sure that the solution becomes a documented part of the network. This implies that routine network maintenance will maintain the implemented solution. For example, if the solution involves reconfiguring a Cisco IOS router, a backup of that new configuration should be made part of routine network maintenance practices.

As a final task, the troubleshooter should report the problem resolution to the appropriate party or parties. Beyond simply notifying a user that a problem has been resolved, the troubleshooter should get user confirmation that the observed symptoms are now gone. This task confirms that the troubleshooter resolved the specific issue reported in the problem report, rather than a tangential issue.

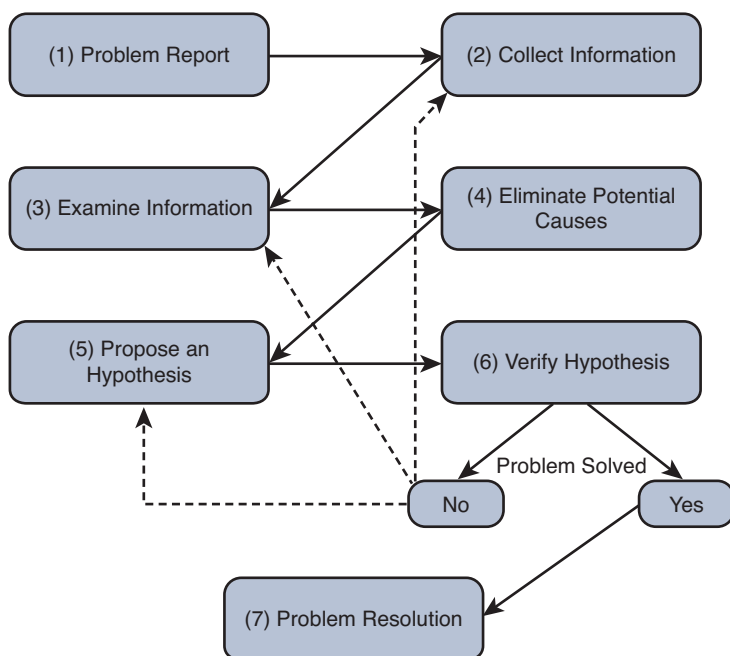


Figure 1-11 *A Structured Troubleshooting Approach (Problem Resolution)*

Popular Troubleshooting Methods

As shown in the structured approach, the elimination of potential causes is a key step. You can use several common troubleshooting methods to narrow the field of potential causes:



- The top-down method
- The bottom-up method
- The divide-and-conquer method
- Following the traffic path
- Comparing configurations
- Component Swapping

This section defines each of these methods in greater detail. However, keep in mind that there is no single best method. Depending on your situation and the issue you are troubleshooting, you may use one or multiple methods.

The Top-Down Method

The top-down troubleshooting method begins at the top layer of the Open Systems Interconnection (OSI) seven-layer model, as shown in Figure 1-12. The top layer is numbered Layer 7 and is named the application layer.

The top-down method first checks the application residing at the application layer and moves down from there. The theory is, when the troubleshooter encounters a layer that is functioning, the assumption can be made that all lower layers are also functioning. For example, if you can ping a remote IP address, because ping uses Internet Control Message Protocol (ICMP), which is a Layer 3 protocol, you can assume that Layers 1–3 are functioning properly. Otherwise, your ping would have failed. A potential downside to this approach is that the troubleshooter needs access to the specific application experiencing a problem to test Layer 7.

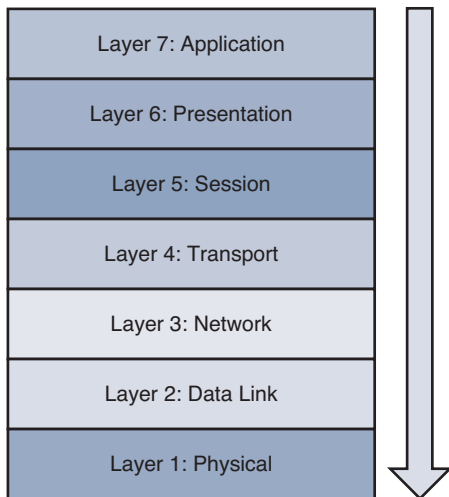


Figure 1-12 *Top-Down Troubleshooting Method*

The Bottom-Up Method

The reciprocal of the top-down method is the bottom-up method, as illustrated in Figure 1-13. The bottom-up method seeks to narrow the field of potential causes by eliminating OSI layers beginning at Layer 1, the physical layer.

Although this is a highly effective method, the bottom-up approach might not be efficient in larger networks because of the time required to fully test lower layers of the OSI model. Therefore, the bottom-up method is often used after employing some other method to narrow the scope of the problem.

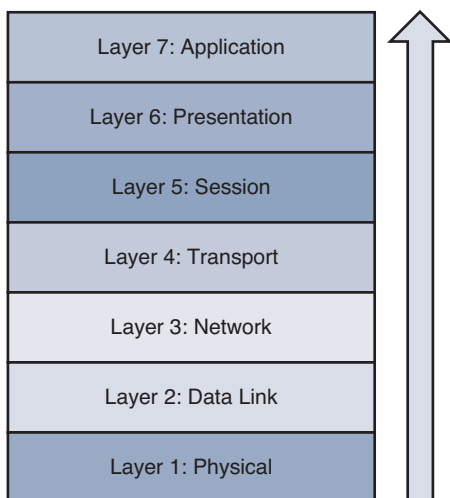


Figure 1-13 *Bottom-Up Troubleshooting Method*

The Divide-and-Conquer Method

After analyzing the information collected for a problem, you might not see a clear indication as to whether the top-down or bottom-up approach would be most effective. In such a situation, you might select the divide-and-conquer approach, which begins in the middle of the OSI stack, as shown in Figure 1-14.

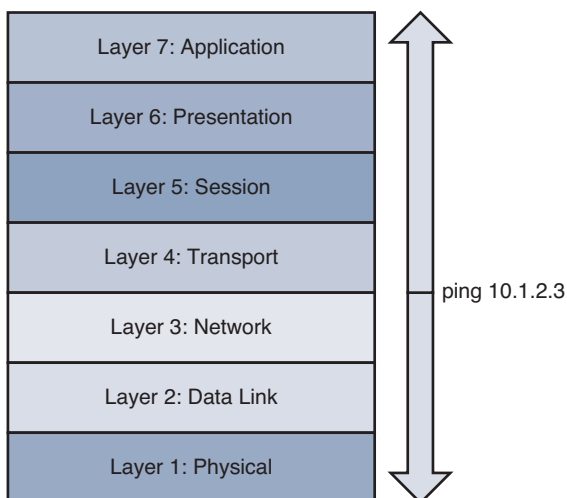


Figure 1-14 *Divide-and-Conquer Troubleshooting Method*

In Figure 1-14, the network administrator issued the **ping 10.1.2.3** command. If the result was successful, the administrator could conclude that Layers 1–3 were operational, and a bottom-up approach could begin from that point. However, if the ping failed, the administrator could begin a top-down approach at Layer 3.

The Following the Traffic Path Method

Another useful troubleshooting approach is to follow the path of the traffic experiencing a problem. For example, if the client depicted in Figure 1-15 is unable to reach its server, you could first check the link between the client and switch SW1. If everything looks good on that link, you could then check the connection between the switch SW1 and router R1. Next, you would check the link between router R1 and switch SW2, and finally the link between switch SW2 and the server.

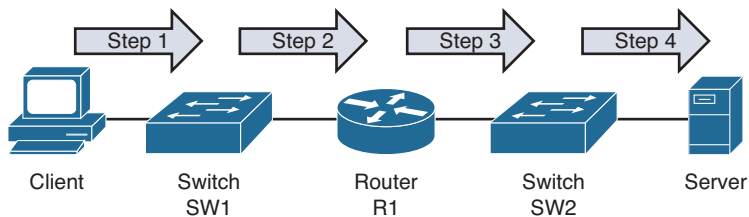


Figure 1-15 *Following the Traffic Path Troubleshooting Method*

The Comparing Configurations Method

Did you ever find yourself looking through a *Highlights* magazine as a child? This magazine often featured two similar pictures, and you were asked to spot the differences. This childhood skill can also prove valuable when troubleshooting some network issues. For example, imagine that you have multiple remote offices, each running the same model of Cisco router. Clients at one of those remote offices cannot obtain an IP address via Dynamic Host Configuration Protocol (DHCP). One troubleshooting approach is to compare that site's router configuration with the router configuration of another remote site that is working properly. You can also look at the configuration stored in a document (Word, Notepad) to see whether it is the same. This methodology is often an appropriate approach for a less-experienced troubleshooter not well versed in the specifics of the network. However, the problem might be resolved without a thorough understanding of what caused the problem. Therefore, the problem is more likely to recur. In addition, what if the documentation is outdated? Now, in addition to the original issue, there are additional issues introduced based on an invalid configuration.

Can you spot the difference in the outputs of Example 1-1a and Example 1-1b?

Example 1-1a show run

```

R1#show run
...OUTPUT OMITTED...
ip dhcp excluded-address 10.8.8.1 10.8.8.10
!
ip dhcp pool POOL-A
    network 10.8.8.0 255.255.255.0
    default-router 10.8.8.11
  
```

```

dns-server 192.168.1.1
netbios-name-server 192.168.1.2
...OUTPUT OMITTED...

```

Example 1-1b `more tftp://10.1.1.10/R1.cfg`

```

R1#more tftp://10.1.1.10/R1.cfg
...OUTPUT OMITTED...
ip dhcp excluded-address 10.8.8.1 10.8.8.10
!
ip dhcp pool POOL-A
    network 10.8.8.0 255.255.255.0
    default-router 10.8.8.1
    dns-server 192.168.1.1
    netbios-name-server 192.168.1.2
...OUTPUT OMITTED...

```

In Example 1-1a, **show run** is displaying the current running configuration. Example 1-1b has the **more tftp://10.1.1.10/R1.cfg** output displaying the archived configuration that was produced as a baseline and stored on a TFTP server. The default router has been changed from 10.8.8.1 to 10.8.8.11.

The Component Swapping Method

Yet another approach to narrowing the field of potential causes of a problem is to physically swap out components. If a problem's symptoms disappear after swapping out a particular component (for example, a cable or a switch), you can conclude that the old component was faulty (either in its hardware or its configuration).

As an example, consider Figure 1-16. A problem report states that the connection between laptop A and switch SW1 is not bringing up a link light on either the laptop or the switch.

As a first step, you might swap out the cable interconnecting these two devices with a known working cable.

If the problem persists, you will want to undo the change you made and then move the cable from switchport 1 to switchport 2. As a next step, you could connect a different laptop to switch SW1. If the problem goes away, you could conclude that the issue is with laptop A. However, if the problem continues, you could swap out switch SW1 with another switch (SW2 in this example). As you test each component and find it is not the problem, undo the change.

Although swapping out components in this fashion might not provide great insight into the specific problem, it could help focus your troubleshooting efforts. For example, if swapping out the switch resolved the issue, you could start to investigate the configuration of the original switch, checking for configuration or hardware issues.

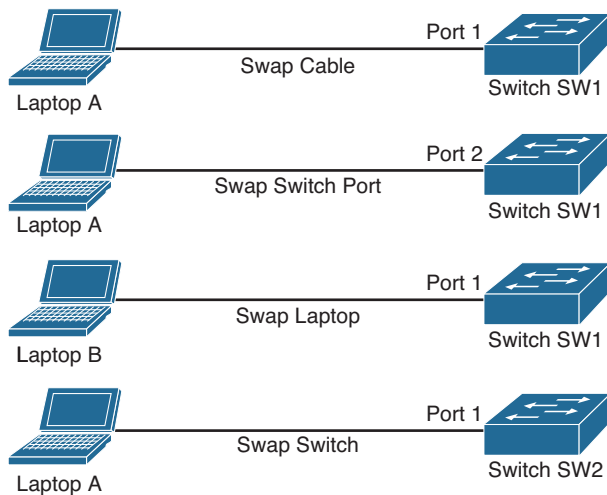


Figure 1-16 *Component Swapping*

Practice Exercise: Selecting a Troubleshooting Approach

As a troubleshooter, you might use one of the previously discussed troubleshooting methods or perhaps a combination of methods to eliminate causes. To illustrate how you might select an appropriate troubleshooting approach, consider the following problem report:

A computer lab at a university contains 48 PCs. Currently, 24 of the PCs cannot access the Internet; the other 24 PCs can. The 24 PCs that cannot currently access the Internet were able to access the Internet yesterday.

Consider which of the previously discussed troubleshooting models might be appropriate for an issue such as the one reported. After you reach your own conclusions about which method or methods would be most appropriate, consider the following rationale:

- **Top-down:** Because the application is working on some PCs in the same location, starting at the application layer will probably not be effective. Although it is possible that 24 of the PCs have some setting in their Internet browser (for example, a proxy configuration) that prevents them from accessing the Internet, these PCs were working yesterday. Therefore, it is unlikely that these 24 PCs were all recently reconfigured with an incorrect application configuration.
- **Bottom-up:** Based on the symptom reported, it is reasonable to guess that there might be an issue with an Ethernet switch (perhaps with a port density of 24). Therefore, a bottom-up approach stands a good chance of isolating the problem quickly.
- **Divide-and-conquer:** The problem seems to be related to a block of PCs, and the problem is probably not application related. Therefore, a divide-and-conquer approach could be useful. Starting at Layer 3 (that is, the network layer), you could

issue a series of pings to determine whether a next-hop gateway is reachable. If the next-hop gateway is not reachable, you could start to troubleshoot Layer 2, checking the Cisco Catalyst switch to which these 24 PCs are attached.

- **Following the traffic path:** The symptom seems to indicate that these 24 PCs might share a common switch. Therefore, following the traffic path to the other end of the cabling (that is, to a switch) could prove useful. Perhaps the switch has lost power resulting in this connectivity issue for the 24 PCs.
- **Comparing configurations:** If a previous troubleshooting method (for example, bottom-up, divide-and-conquer, or following the traffic path) reveals that the 24 PCs that are not working are connected to one Cisco Catalyst switch, and the 24 PCs that are working are connected to another Cisco Catalyst switch, comparing the configuration of those two switches could prove helpful.
- **Component swapping:** Because the 24 PCs are experiencing the same problem within a short time frame (since yesterday), it is unlikely that swapping cables would be useful. However, if these 24 PCs connect to the same Cisco Catalyst switch, swapping out the switch could help isolate the problem.

As you can see from the analysis of the different methods, each has the possibility of providing valuable information that will help you solve this issue. Therefore, you will not usually rely on just one method while you are troubleshooting. You will combine the different methods to produce the most accurate picture possible.

Introduction to Network Maintenance

Network maintenance is an inherent component of a network administrator's responsibilities. However, that network administrator might be performing maintenance tasks in response to a reported problem. This reactive approach is unavoidable, because unforeseen issues do arise. However, the occurrence of these interrupt-driven maintenance tasks can be reduced by proactively performing regularly scheduled maintenance tasks.

You could think of regularly scheduled tasks, such as performing backups and software upgrades, as important but not urgent. Spending more time on the important tasks can help reduce time spent on the urgent tasks (for example, responding to user connectivity issues or troubleshooting a network outage).

This section begins by identifying several common network maintenance tasks that are seen in most organizations. It introduces us to standard network maintenance models; however, these off-the-shelf models might not be a perfect fit for the organization. So, this section discusses how to adapt a well-known model to individual needs. It concludes by discussing several procedures that are a must for maintenance success.

Defining Network Maintenance

Network maintenance, at its essence, is doing whatever is required to keep the network functioning and meeting the business needs of an organization. Therefore, you need to analyze the business needs of the organization and determine which maintenance tasks

are necessary for the success of the business. Time and money need to be spent wisely, and critical business processes need more attention. For example, are you going to back up each PC in the company on a nightly basis or are you going to have all users store resources on a central server and back up the central server?

Some examples of the tasks that fall under the umbrella of network maintenance are as follows:



- Hardware and software installation and configuration
- Troubleshooting problem reports
- Monitoring and tuning network performance
- Planning for network expansion
- Documenting the network and any changes made to the network
- Ensuring compliance with legal regulations and corporate policies
- Securing the network against internal and external threats
- Backing up files and databases

Obviously, this listing is only a sampling of network maintenance tasks. Also, keep in mind that the list of tasks required to maintain your network could differ significantly from the list of tasks required to maintain another network. You need to align your maintenance tasks with your business needs.

Proactive Versus Reactive Network Maintenance

Network maintenance tasks can be categorized as one of the following:

- **Interrupt-driven tasks:** Involve resolving issues as they are reported
- **Structured tasks:** Performed as a predefined plan

Interrupt-driven tasks are not planned. They result from something happening in the network that requires your attention. It may be your immediate attention, or it may be something you can put off until later. Interrupt-driven tasks can never be completely eliminated; however, you can significantly reduce their occurrence when you have a strategic structured approach in place.

Implementing a structured maintenance approach confers many benefits. It reduces total network downtime because you are aware of problems and fix them before they become a major issue. It is more cost-effective because fewer major problems occur, resulting in less resources being consumed for problem resolution. If you do have an unplanned network outage (interrupt-driven), you can resolve it more quickly because a predefined plan is in place to handle that type of outage. In addition, you will also know which tools are required and how to use them to solve the problem. A structured maintenance approach also includes planning for future network capacity; therefore, appropriate hardware and software purchases can be made early on, reducing obsolescence of relatively new purchases.

A structured approach also takes into consideration underlying business goals. Therefore, resources can be allocated that complement business drivers. Security vulnerabilities are more likely to be discovered through ongoing network monitoring, which is another component of a structured maintenance approach, as discussed later in this chapter.

Well-Known Network Maintenance Models

The subtleties of each network should be considered when constructing a structured network maintenance model. However, rather than starting from scratch, you might want to base your maintenance model on one of the well-known maintenance models and make adjustments as appropriate.

The following is a sampling of some of the more well-known maintenance models:



- **FCAPS:** FCAPS (which stands for fault management, configuration management, accounting management, performance management, and security management) is a network maintenance model defined by the International Organization for Standardization (ISO).
- **ITIL:** IT Infrastructure Library (ITIL) defines a collection of best practice recommendations that work together to meet IT business management goals.
- **Cisco Lifecycle Services:** The Cisco Lifecycle Services maintenance model defines distinct phases in the life of a Cisco technology in a network. These phases are prepare, plan, design, implement, operate, and optimize. As a result, the Cisco Lifecycle Services model is often referred to as the PPDIOO model.

Example of Adapting a Network Maintenance Model

The maintenance model you use in your network should reflect business drivers, resources, and expertise unique to your network. Once you choose the model, you must adapt the model to your environment. Suppose, for example, that you have selected the ISO FCAPS model as the foundation for your maintenance model. To adapt the FCAPS model for your environment, you should identify specific tasks to perform on your network for each element of the FCAPS model. Table 1-3 provides a sampling of tasks that might be categorized under each of the FCAPS management areas.

Table 1-3 *FCAPS Management Tasks*

Type of Management	Examples of Management Tasks
Fault management	Use network management software to collect information from routers and switches. Send an e-mail alert when processor utilization or bandwidth utilization exceeds a threshold of 80 percent. Respond to incoming trouble tickets from the help desk.
Configuration management	Require logging of any changes made to network hardware or software configurations. Implement a change management system to alert relevant personnel of planned network changes.

Type of Management	Examples of Management Tasks
Accounting management	Invoice IP telephony users for their long-distance and international calls. Keeping track of what is being done on the network and when it is being done.
Performance management	Monitor network performance metrics for both LAN and WAN links. Deploy appropriate quality of service (QoS) solutions to make the most efficient use of relatively limited WAN bandwidth, while prioritizing mission-critical traffic.
Security management	Deploy firewall, virtual private network (VPN), and intrusion prevention system (IPS) technologies to defend against malicious traffic. Create a security policy dictating rules of acceptable network use. Use an authorization, authentication, and accounting (AAA) server to validate user credentials, assign appropriate user privileges, and log user activity.

By clearly outlining a maintenance methodology and defining actionable and measurable processes you can reduce network downtime and more effectively perform interrupt-driven tasks.

Common Maintenance Procedures

No two network maintenance models will be exactly the same, and no two organizations will implement them in exactly the same way, because of the different business drivers involved. However, there are tasks common to nearly all network maintenance models that will be implemented by all organizations regardless of the business drivers. This section discusses common maintenance tasks that all organizations should be performing.

Routine Maintenance Tasks

Regardless of the organization, there will be maintenance tasks in each organization that occur routinely. This routine can be hourly, daily, weekly, monthly, per quarter, or per year. As you can see, the routine can be frequent or infrequent, but it can also be regular or irregular. For example, adding users or moving users and updating the network based on the user changes is going to be different each time. We cannot have a regular schedule for these types of tasks because they are infrequent and irregular. However, backing up a server on a daily basis at 10:00 p.m. is frequent and regular.

The key with all these tasks is that they are routine regardless of them being frequent, infrequent, regular, or irregular and should be present in a listing of procedures contained in a network maintenance model. Following is a listing of such common maintenance tasks:

- **Configuration changes:** Businesses are dynamic environments, where relocation of users from one office space to another, the addition of temporary staffers, and new hires are commonplace. In response to organizational changes, network administra-

tors need to respond by performing appropriate reconfigurations and additions to network hardware and software. These processes are often referred to as moves, adds, and changes.

- **Replacement of older or failed hardware:** As devices age, their reliability and comparable performance tend to deteriorate. Therefore, a common task is the replacement of older hardware, typically with better performing and more feature-rich devices. Occasionally, production devices fail, thus requiring immediate replacement.
- **Scheduled backups:** Recovery from a major system failure can occur much quicker if network data and device configurations have been regularly backed up. Therefore, a common network maintenance task is to schedule, monitor, and verify backups of selected data and configuration information. These backups can also be useful in recovering important data that was deleted.
- **Updating software:** Updates to operating system software (for servers, clients, and even network devices) are periodically released. The updates often address performance issues and security vulnerabilities. New features are also commonly offered in software upgrades. Therefore, performing routine software updates becomes a key network maintenance task.
- **Monitoring network performance:** The collection and interpretation of traffic statistics, bandwidth utilization statistics, and resource utilization statistics for network devices are common goals of network monitoring. Through effective network monitoring (which might involve the collection and examination of log files or the implementation of a high-end network management server), you can better plan for future expansion (that is, capacity planning), anticipate potential issues before they arise, and better understand the nature of the traffic flowing through your network.

Scheduled Maintenance

Take a moment and define the network maintenance tasks for your network. After doing so, rank them in order of priority. Some tasks will undoubtedly be urgent in nature and need a quick response when things go wrong (for example, replacing a failed router that connects the business to the Internet). Other tasks can be scheduled. For example, you might schedule weekly full backups of your network's file servers, and you might have a monthly maintenance window, during which time you apply software patches.

By having such a schedule for routine maintenance tasks, network administrators are less likely to forget an important task, because they were busy responding to urgent tasks. Also, users can be made aware of when various network services will be unavailable, due to maintenance windows, thus minimizing the impact on workflow.

Managing Network Changes

Making changes to a network often has the side effect of impacting the productivity of users relying on network resources. In addition, a change to one network component might create a problem for another network component. For example, perhaps a firewall

was installed to provide better security for a server farm. However, in addition to common protocols that were allowed to pass through the firewall (for example, DNS, SMTP, POP3, HTTP, HTTPS, and IMAP), one of the servers in the server farm acted as an FTP server, and the firewall configuration did not consider that server. Therefore, the installation of a firewall to better secure a server farm resulted in a troubleshooting issue, where users could no longer reach their FTP server.

The timing of network changes should also be considered. Rather than taking a router down to upgrade its version of Cisco IOS during regular business hours, such an operation should probably be performed during off hours.

Making different organization areas aware of upcoming maintenance operations can also aid in reducing unforeseen problems associated with routine maintenance. For example, suppose that one information technology (IT) department within an organization is responsible for maintaining WAN connections that interconnect various corporate offices, whereas another IT department is charged with performing network backups. If the WAN IT department plans to upgrade the WAN link between a couple of offices at 2:00 a.m. next Tuesday, the IT department in charge of backups should be made aware of that planned upgrade, because a backup of remote data (that is, data accessible over the WAN link to be upgraded) might be scheduled for that same time period.

Some organizations have a formalized change management process, where one department announces online their intention to perform a particular maintenance task during a specified time period. Other departments are then notified of this upcoming change, and determine whether the planned change will conflict with that department's operations. If a conflict is identified, the departments can work together to accommodate one another's needs.

Of course, some network maintenance tasks are urgent (for example, a widespread network outage). Those tasks need timely responses, without going through a formalized change management notification process and allowing time for other departments to respond.

When defining a change management system for your organization, consider the following:



- Who is responsible for authorizing various types of network changes?
- Which tasks should only be performed during scheduled maintenance windows?
- What procedures should be followed prior to making a change (for example, backing up a router's configuration prior to installing a new module in the router)?
- What measurable criteria determine the success or failure of a network change?
- How will a network change be documented, and who is responsible for the documentation?
- How will a rollback plan be created, such that a configuration can be restored to its previous state if the changes resulted in unexpected problems?
- Under what circumstances can formalized change management policies be overridden, and what (if any) authorization is required for an override?

Maintaining Network Documentation

Network documentation typically gets created as part of a network's initial design and installation. However, keeping that documentation current, reflecting all changes made since the network's installation, should be part of any network maintenance model. Keeping documentation current helps more effectively isolate problems when troubleshooting. In addition, accurate documentation can prove to be valuable to designers who want to scale the network.

At a basic level, network documentation could consist of physical and logical network diagrams, in addition to a listing of network components and their configurations. However, network documentation can be much more detailed, including such components as formalized change management procedures, a listing of contact information (for example, for service providers and points of contact in an organization's various IT groups), and the rationale for each network change made.

While the specific components in a set of network documentation can vary, just as the procedures in a network maintenance model vary, the following list outlines common elements found in a set of network documentation:



- **Logical topology diagram:** A logical topology diagram shows the interconnection of network segments, the protocols used, and how end users interface with the network, deployed VLANs, and IP addressing, to name a few. However, this diagram is not concerned with the physical locations of network components.
- **Physical topology diagram:** Unlike a logical topology diagram, a physical topology diagram shows how different geographical areas (for example, floors within a building, buildings, or entire sites) interconnect. The diagram reflects where various network components are physically located.
- **Listing of interconnections:** A listing of interconnections could be, for example, a spreadsheet that lists which ports on which devices are used to interconnect network components or connect out to service provider networks. Circuit IDs for service provider circuits might be included in this documentation.
- **Inventory of network equipment:** An inventory of network equipment would include such information as the equipment's manufacturer, model number, version of software, and modules installed, in addition to information about the licensing of the software, serial number, and an organization's asset tag number.
- **IP address assignments:** An organization might use private IP address space internally and use Network Address Translation (NAT) to translate those private IP address space numbers into publicly routable IP addresses. Alternatively, an organization might have public IP addresses assigned to some or all of their internal devices. A classful IP address space (either public or private) might be subdivided within an organization, resulting in subnets with a nondefault subnet mask. For IPv6 the organization might be manually assigning the interface ID to each device, using EUI-64, or a combination of both. These types of IP addressing specifications would be included in a set of network documentation.

- **Configuration information:** When a configuration change is made, the current configuration should be backed up. With a copy of current configuration information, a device could be replaced quicker, in the event of an outage. Beyond having a backup of current configuration information, some network administrators also maintain archival copies of previous configurations. These older configurations could prove useful when attempting to roll back to a previous configuration state or when trying to duplicate a previous configuration in a new location. It is a good practice to name archival copies of previous configurations based on a certain format that makes sense to you. For example, some companies name their archival copies by date, others by function, and still others by a combination of both.
- **Original design documents:** Documents created during the initial design of a network might provide insight into why certain design decisions were made and how the original designers envisioned future network expansion.

Larger network environments often benefit from having step-by-step guidelines for troubleshooting a given network issue. Such a structured approach to troubleshooting helps ensure that all troubleshooting personnel use a common approach. Although a network issue might be successfully resolved through various means, if different personnel troubleshoot using different approaches, at some point those approaches might conflict with one another, resulting in further issues.

For example, consider one network administrator that configures IEEE 802.1Q trunking on Cisco Catalyst switches by disabling Dynamic Trunking Protocol (DTP) frames and forcing a port to act as a trunk port. Another network administrator within the same company configures 802.1Q trunking by setting a port's trunk state to desirable, which creates a trunk connection only if it receives a DTP frame from the far end of the connection. These two approaches are not compatible, and if each of these two network administrators configured different ends of what they intended to be an 802.1Q trunk, the trunk connection would never come up. This example illustrates the criticality of having clear communication among IT personnel and a set of standardized procedures to ensure consistency in network configuration and troubleshooting practices.

Restoring Operations After a Failure

Although most modern network hardware is very reliable, failures do occur from time to time. Aside from hardware failures, environmental factors could cause a network outage. As a few examples, the failure of an air conditioner unit could cause network equipment to overheat, water leakage due to flooding or plumbing issues could cause hardware failures, and a fire could render the network equipment unusable.

Planning and provisioning hardware and software for such outages before they occur can accelerate recovery time. To efficiently replace a failed (or damaged) device, you should be in possession or have the ability to acquire relatively quickly the following:

- **Duplicate hardware:** The hardware can be stored locally or it can be attainable through a supplier that can get you the device within a certain time based on a service level agreement (SLA).

- **Operating system and application software (along with any applicable licensing) for the device:** Although you can get this from the manufacturer (such as Cisco), it is advisable to have an exact copy of the operating systems and application software stored locally for each device you are using in the organization.
- **Backup of device configuration information:** When a failure happens, you need to restore your device to its last known good configuration. It is ideal to have a backup of the configuration files on a server in the organization. However, if that is not possible, at a minimum have the configurations documented in Notepad somewhere. You do not want to be caught in a situation where you have no information related to the configuration of a device being restored.

Measuring Network Performance

Network monitoring is a proactive approach to network maintenance, enabling you to be alerted to trends and utilization statistics (as a couple of examples). These statistics can forecast future issues, allowing you to be proactive and fix problems before they affect network users. Also, if you work for a service provider, network performance monitoring can ensure that you are providing an appropriate service level to a customer. Conversely, if you are a customer of a service provider, network monitoring can confirm that the service provider is conforming to the SLA for which you are paying.

The Troubleshooting and Network Maintenance Relationship

A structured troubleshooting approach provides step-by-step processes that offer a repeatable consistent plan that makes the troubleshooter more efficient and effective. During our coverage of the structured approach you might have noticed that documentation, baselines, change control, and communication were mentioned. All of these are fundamental assets to your success as a troubleshooter. However, they do not simply appear from the ether, as you have seen from the discussion of network maintenance. For example, documentation and baselines are created at a specific point in time for a device and provide a snapshot of the health and configuration of that device at that point. As a result, we will heavily rely on these resources when issues occur. What happens if someone neglects to update the documentation or baselines based on changes that may have occurred during scheduled maintenance or some past issue? What happens if we have difficulty communicating with others or they withhold information from us? These assets become liabilities as they are unable to address the question: What *should* be occurring in the network?

As you have seen, network maintenance tasks often include troubleshooting tasks, and vice versa. For example, when installing a new network component as part of ongoing network maintenance, an installer is often required to troubleshoot the installation until the new network component is functioning properly. Also, when troubleshooting a network issue, the troubleshooter might use network documentation (for example, a physical topology diagram created as part of a network maintenance task) to help isolate a problem.

This interrelationship between maintenance and troubleshooting suggests that the effectiveness of your troubleshooting efforts is influenced by the effectiveness of your routine network management tasks. Because these tasks are so interrelated, you might want to take proactive measures to ensure your structured maintenance and troubleshooting processes complement one another. For example, both network troubleshooting and maintenance include a documentation component. Therefore, the value of a centralized repository of documentation increases as a result of its use for both maintenance and troubleshooting efforts.

Maintaining Current Network Documentation

A set of maintained network documentation can dramatically improve the efficiency of troubleshooting efforts. For example, if a troubleshooter is following the path that specific traffic takes through a network, physical and logical topology diagrams could help identify the next network component to check.

A danger with relying on documentation is that if the documentation is dated (not maintained), troubleshooters could be led down an incorrect path because of their reliance on that documentation. Such a scenario is often worse than not having documentation at all, because in the absence of documentation, troubleshooters are not led down the wrong path during the troubleshooting process; they have to create their own path.

Although few argue with the criticality of maintaining current documentation, documenting troubleshooting efforts, in practice, often falls by the wayside. The lack of follow-through when it comes to documenting what happened during a troubleshooting scenario is understandable. The troubleshooter's focus is on resolving a reported issue in a timely manner (that is, an urgent task) rather than documenting what they are doing at the time (that is, an important task). Following are a few suggestions to help troubleshooters keep in mind the need to document their steps:



- **Require documentation:** By making documentation a component in the troubleshooting flow, troubleshooters know that before a problem report or a trouble ticket can be closed out, they must generate appropriate documentation. This knowledge often motivates troubleshooters to perform some level of documentation (for example, scribbling notes on the back of a piece of paper) as they are performing their tasks, as opposed to later trying to recall what they did from memory, thus increasing the accuracy of the documentation.
- **Schedule documentation checks:** A structured maintenance plan could include a component that routinely requires verification of network documentation and when it was last updated based on timestamps.
- **Automate documentation:** Because manual checks of documentation might not be feasible in larger environments, automated processes could be used to, for example, compare current and backup copies of device configurations. Any difference in the configurations indicates that someone failed to update the backup configuration of a device after making a configuration change to that device. To assist with the automation of backups, Cisco IOS offers the Configuration Replace and Configuration Rollback feature and the Embedded Event Manager.

Establishing a Baseline

As previously mentioned, troubleshooting involves knowing what should be happening on the network, observing what is currently happening on the network, and determining the difference between the two. To determine what should be happening on the network, a baseline of network performance should be measured as part of a routine maintenance procedure and updated on a regular basis.



For example, a routine network maintenance procedure might require that a **show processes cpu** command be periodically issued on all routers in a network, with the output logged and archived. As shown in Example 1-2, the **show processes cpu** command demonstrates the 5-second, 1-minute, and 5-minute CPU utilization averages. When troubleshooting a performance problem on a router, you could issue this command to determine how a router is currently operating. However, without a baseline as a reference before troubleshooting, you might not be able to draw a meaningful conclusion based on the command output.

Example 1-2 Monitoring Router CPU Utilization

```
R1# show processes cpu
cpu utilization for five seconds: 18%/18%; one minute: 22%; five minutes: 22%
PID Runtime(ms)   Invoked    uSecs      5Sec   1Min   5Min   TTY process
  1         0         1         0      0.00%  0.00%  0.00%    0 chunk Manager
  2         4       167        23      0.00%  0.00%  0.00%    0 Load Meter
  3       821      188     4367      0.00%  0.13%  0.14%    0 Exec
  4         4         1     4000      0.00%  0.13%  0.00%    0 EDDRI_MAIN
  5    43026     2180    19736      0.00%  4.09%  4.03%    0 Check heaps
...OUTPUT OMITTED...
```

Communication

Each of the troubleshooting steps outlined in the structured approach requires clear communication. Table 1-4 describes how communication plays a role in each troubleshooting phase.

Table 1-4 Importance of Clear Communication During Troubleshooting

Troubleshooting Steps	The Role of Communication
Problem report	When a user reports a problem, clear communication with that user helps define the problem. For example, the user can be asked exactly what is not working correctly, if she made any recent changes, and when the problem started.
Collect information	Some information collected might come from other parties (for example, a service provider). Clearly communicating with those other parties helps ensure collection of the proper data.

Troubleshooting Steps	The Role of Communication
Examine collected information	Because a troubleshooter is often not fully aware of all aspects of a network, collaboration with other IT personnel is often necessary.
Eliminate potential causes	The elimination of potential causes might involve consultation with others. This consultation could provide insight leading to the elimination of a potential cause.
Propose an Hypothesis	The consultation a troubleshooter conducts with other IT personnel when eliminating potential causes might also help the troubleshooter more accurately hypothesize a problem's underlying cause.
Verify hypothesis	Temporary network interruptions often occur when verifying an hypothesis; therefore, the nature and reason for an interruption should be communicated to the users impacted.
Problem resolution	After a problem is resolved, the user originally reporting the problem should be informed, and the user should confirm that the problem has truly been resolved.

Also, depending on the severity of an issue, multiple network administrators could be involved in troubleshooting a problem. Because these troubleshooters might be focused on different tasks at different times, it is possible that no single administrator can report on the overall status of the problem. Therefore, when managing a major outage, those involved in troubleshooting the outage should divert user inquiries to a manager who is in frequent contact with the troubleshooting personnel. As a side benefit, being able to quickly divert user requests for status reports to a manager helps minimize interruptions from users.

Change Management

Managing when changes can be made and by whose authority helps minimize network downtime. In fact, these two factors (that is, when a change is allowed and who can authorize it) are the distinguishing factors between making a change as part of a routine maintenance plan and making a change as part of a troubleshooting process.

The process of change management includes using policies that dictate rules regarding how and when a change can be made and how that change is documented. Consider the following scenario, which illustrates how a maintenance change could be a clue while troubleshooting a problem report:

Last week, a network administrator attempted to better secure a Cisco Catalyst switch by administratively shutting down any ports that were in the down/down state (that is, no physical layer connectivity to a device). This morning, a user reported that her PC could not access network resources. After clearly defining the problem, the troubleshooter asked whether anything had changed, as part of the col-

lect information troubleshooting phase. Even though the user was unaware of any changes, she mentioned that she had just returned from vacation, thus leading the troubleshooter to wonder if any network changes had occurred while the user was on vacation. Thanks to the network's change management system, the troubleshooter was able to find in the documentation that last week an administrator had administratively shut down this user's switchport because it was down/down while the user was on vacation and his computer was shut off.

The previous scenario is an excellent example of how following a structured troubleshooting approach, having accurate documentation, and a sound change management policy minimized the total time it took the troubleshooter to solve the problem.

Exam Preparation Tasks

As mentioned in the section “How to Use This Book” in the Introduction, you have a couple of choices for exam preparation: the exercises here; Chapter 22, “Final Preparation;” and the exam simulation questions on the CD-ROM.

Review All Key Topics

Review the most important topics in this chapter, noted with the Key Topic icon in the outer margin of the page. Table 1-5 lists a reference of these key topics and the page numbers on which each is found.



Table 1-5 *Key Topics for Chapter 1*

Key Topic Element	Description	Page Number
List	Outlines the simplified troubleshooting flow	10
Table 1-2	Identifies the five steps used while diagnosing a problem	10
List	Outlines the structured troubleshooting flow	11
Section	Provides details of each step during structured troubleshooting	13
List	Lists the various troubleshooting methods that can be used to narrow the field of potential causes	20
List	Lists examples of network maintenance tasks	27
List	Lists examples of network maintenance models	28
List	Identifies questions that need to be addressed while implementing a change management system	31
List	Outlines various types of documents that should exist and be maintained within an organization	32
List	Examples of how to help troubleshooters remember the importance of documenting their steps	35
Paragraph	Identifies the importance of a baseline	36

Define Key Terms

Define the following key terms from this chapter and check your answers in the glossary:

interrupt-driven task, structured maintenance task, FCAPS, ITIL, Cisco Lifecycle Services, shoot from the hip, top-down method, bottom-up method, divide-and-conquer method, following the traffic path method, comparing configurations method, component swapping method, baseline, change management, documentation



This chapter covers the following topics:

- **The Troubleshooting and Network Maintenance Toolkit:** This section introduces you to the essential tools for troubleshooting and maintenance tasks.
- **Using Cisco IOS to Verify and Define the Problem:** This section reviews the ping, telnet, and traceroute utilities.
- **Using Cisco IOS to Collect Information:** This section focuses on how to use the CLI to collect information for troubleshooting and maintenance.
- **Collecting Information in Transit:** This section identifies how you can configure switches to send copies of frames to packet capturing devices using SPAN and RSPAN.
- **Using CLI Tools to Document a Network:** This section focuses on the steps and commands required to successfully document a network diagram.

Troubleshooting and Maintenance Tools

Collecting network information is an ongoing process. There is no argument that you will be collecting network information when there is an issue. However, if that is the only time you collect network information, you are missing the necessary key element of an efficient and effective troubleshooting process. To be an efficient and effective troubleshooter, you need network information about the good times and the bad times, and you need it now, not later. Therefore, you need to gather baseline data on a regular basis so that you have something to compare your current issue to. In addition, the statistics related to certain network events (for example, processor utilization on a network server exceeding a specified threshold) could trigger the writing of log information (for example, to a syslog server), so you have a snapshot of the device's health at that point in time.

This chapter introduces you to a sampling of Cisco IOS tools and features designed for network maintenance and troubleshooting.

“Do I Know This Already?” Quiz

The “Do I Know This Already?” quiz allows you to assess whether you should read this entire chapter thoroughly or jump to the “Exam Preparation Tasks” section. If you are in doubt about your answers to these questions or your own assessment of your knowledge of the topics, read the entire chapter. Table 2-1 lists the major headings in this chapter and their corresponding “Do I Know This Already?” quiz questions. You can find the answers in Appendix A, “Answers to the ‘Do I Know This Already?’ Quizzes.”

Table 2-1 “Do I Know This Already?” Section-to-Question Mapping

Foundation Topics Section	Questions
The Troubleshooting and Network Maintenance Toolkit	1–6
Using Cisco IOS to Verify and Define the Problem	7–9
Using Cisco IOS to Collect Information	10
Collecting Information in Transit	11
Using CLI Tools to Document a Network	12

Caution The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you do not know the answer to a question or are only partially sure of the answer, you should mark that question as wrong for purposes of the self-assessment. Giving yourself credit for an answer that you correctly guess skews your self-assessment results and might provide you with a false sense of security.

1. Which three of the following are components that would be most useful when recovering from a network equipment outage?
 - a. Backup of device configuration information
 - b. Physical topology
 - c. Duplicate hardware
 - d. Operating system and application software (along with any applicable licensing) for the device
2. The types of information collection used in troubleshooting fall into which three broad categories?
 - a. Troubleshooting information collection
 - b. Baseline information collection
 - c. QoS information collection
 - d. Network event information collection
3. Which of the following would be appropriate for a collaborative web-based documentation solution?
 - a. Blog
 - b. Vlog
 - c. Wiki
 - d. Podcast
4. Which command enables you to view archival copies of a router's startup configuration?
 - a. `show backup`
 - b. `show archive`
 - c. `show flash: | begin backup`
 - d. `show ftp: | begin archive`

5. Which of the following is a Cisco IOS technology that uses a collector to take data from monitored devices and present graphs, charts, and tables to describe network traffic patterns?
- a. NBAR
 - b. NetFlow
 - c. QDM
 - d. IPS
6. Which two of the following are characteristics of the NetFlow feature? (Choose the two best answers.)
- a. Collects detailed information about traffic flows
 - b. Collects detailed information about device statistics
 - c. Uses a pull model
 - d. Uses a push model
7. Which of the following is the ping response to a transmitted ICMP echo datagram that needed to be fragmented when fragmentation was not permitted?
- a. U
 - b. .
 - c. M
 - d. D
8. Which command can be used to determine whether transport layer connectivity is functioning?
- a. telnet
 - b. ping
 - c. traceroute
 - d. arp -a
9. Which command enables you to determine whether a routing loop exists?
- a. telnet
 - b. ping
 - c. traceroute
 - d. arp -a

- 10.** Which of the following commands displays a router's running configuration, starting where the routing protocol configuration begins?
- a.** `show running-config | tee router`
 - b.** `show running-config | begin router`
 - c.** `show running-config | redirect router`
 - d.** `show running-config | append router`
- 11.** What feature available on Cisco Catalyst switches enables you to connect a network monitor to a port on one switch to monitor traffic flowing through a port on a different switch?
- a.** RSTP
 - b.** SPAN
 - c.** RSPAN
 - d.** SPRT
- 12.** What IOS command enables you to discover the Cisco devices that are directly connected to other Cisco devices?
- a.** `show ip interface brief`
 - b.** `show interface status`
 - c.** `show cdp neighbor`
 - d.** `show version`

Foundation Topics

The Troubleshooting and Network Maintenance Toolkit

As previously discussed, troubleshooting and maintenance go hand and hand. A relationship exists between the two. Therefore, the tools we use for troubleshooting and maintenance will be very similar, if not the same.

Chapter 1, “Introduction to Troubleshooting and Network Maintenance,” introduced you to a series of steps that provide a structured troubleshooting process. Several of these steps involve the use of tools that will help gather, examine, and compare information, in addition to fixing and possibly rolling back configurations. Let’s examine four of these steps:

- **Problem report:** By proactively monitoring network devices with specialized reporting tools, you might be alerted to impending performance issues before users are impacted and report it.
- **Collect information:** The collection of information when troubleshooting a problem can often be made more efficient through the use of specialized maintenance and troubleshooting tools. At this point, you are gathering more information that will help paint a clearer picture of the issue at hand.
- **Examine collected information:** As troubleshooters investigate the information they collected during the troubleshooting process, they need to know what normal network behavior looks like. They can then contrast that normal behavior against what they are observing in their collected data. Specialized maintenance tools can be used in a network to collect baseline data on an ongoing basis so that it is available and current when needed.
- **Verify hypothesis:** Specialized maintenance and troubleshooting tools help a troubleshooter implement his fix for an issue; however, he can also help roll back an attempted fix, if that fix proves unsuccessful.

If you look closely, the information that is collected essentially falls into one of three categories:



- **Troubleshooting information collection:** This is the information collected while troubleshooting an issue that was either reported by a user or a network management station (NMS).
- **Baseline information collection:** This is the information collected when the network is operating normally. This information provides a frame of reference against which other data can be compared when we are troubleshooting an issue.

- **Network event information collection:** This is the information collected when our devices automatically generate alerts in response to specific conditions (for example, configured utilization levels on a switch, router, or server being exceeded). These alerts can be simple notification messages or emergency messages. At some point, they will come in handy.

Because such a tight relationship exists between troubleshooting and network maintenance, you should identify the tools required to carry out your maintenance processes based on how well targeted they are toward your specific business processes and tasks, while helping you focus your troubleshooting efforts without having to wade through reams of irrelevant information. This section focuses on tools that are necessary for troubleshooting and maintenance tasks.

Network Documentation Tools

It is fitting that we start this chapter with a discussion on network documentation tools, because without them, all the other tools we use mean nothing if we are not documenting their findings. Chapter 1 discussed the importance of network documentation. However, for this documentation to truly add value and be an asset, it should be easy to retrieve and, more important, be current. To keep the documentation current is a challenge for most people. The big reason is time. However, you can make it less challenging and less time-consuming if it is easy to update with the proper tools.

Many solutions are available on the market. The features you want the tool to provide will determine the overall cost. However, you do not have to purchase the most expensive tool to get the best product. Shop around and communicate with the vendors to see what they have to offer you and your business needs. Get free trials and work with them for a while. That is the only way you will be able to determine whether the product will work for you. A couple of documentation management system examples are as follows:

- **Trouble ticket reporting system:** Several software applications are available for recording, tracking, and archiving trouble reports (that is, trouble tickets). These applications are often referred to as *help desk applications*. However, their usefulness extends beyond the help desk environment.
- **Wiki:** A wiki can act as a web-based collaborative documentation platform. A popular example of a wiki is Wikipedia (<http://www.wikipedia.com>), an Internet-based encyclopedia that can be updated by users. This type of wiki technology can also be used on your local network to maintain a central repository for documentation that is both easy to access and easy to update.

The true power of documentation is seen during the troubleshooting process, and this is especially true when you have a well-organized, searchable repository of information. During the troubleshooting process, if you have a searchable database of past issues that were solved, and guides that can be followed to resolve issues, you can leverage that information and be more efficient and effective. However, do not forget to update the documentation after you solve the ticket. Just because it was reported in the past and already had a resolution does not mean you can skip the documentation process. At

some point, we may need to rely on the number of entries in a ticket reporting system to determine whether some greater issue is lurking in the shadows and causing the reoccurrence of the same minor issues over and over.

Basic Tools

Troubleshooting and network maintenance tools often range in expense from free to tens of thousands of dollars. Similarly, these tools vary in their levels of complexity and usefulness for troubleshooting and maintaining specific issues. You need to select tools that balance your troubleshooting and maintenance needs while meeting your budgetary constraints.

Regardless of budget, all Cisco troubleshooting and network maintenance toolkits will contain the command-line interface (CLI) commands that are executable from a router or switch prompt. In addition, many network devices have a graphical user interface (GUI) to assist network administrators in their configuration and monitoring tasks. External servers (for example, backup servers, logging servers, and time servers) can also collect, store, or provide valuable information for day-to-day network operations and for troubleshooting and maintenance.

CLI Tools

Cisco IOS offers a wealth of CLI commands, which can prove invaluable when troubleshooting a network issue. For example, a **show** command, which displays a static snapshot of information, can display router configuration information and the routes that have been learned by a routing process. The **debug** command can provide real-time information about router or switch processes. The focus of this book is on those **show** and **debug** CLI commands that will assist us in solving trouble tickets. To illustrate, consider Example 2-1, which shows router R2 receiving Open Shortest Path First (OSPF) link-state updates from its OSPF neighbors as those updates occur.

Example 2-1 Sample debug Output

```
R2#debug ip ospf events
OSPF events debugging is on
R2#
*Mar  1 00:06:06.679: OSPF: Rcv LS UPD from 10.4.4.4 on Serial1/0.2 length 124
    LSA count 1
*Mar  1 00:06:06.691: OSPF: Rcv LS UPD from 10.3.3.3 on Serial1/0.1 length 124
    LSA count 1
*Mar  1 00:06:06.999: OSPF: Rcv LS UPD from 10.4.4.4 on Serial1/0.2 length 124
    LSA count 1
*Mar  1 00:06:07.067: OSPF: Rcv LS UPD from 10.3.3.3 on Serial1/0.1 length 156
    LSA count 2
```

This is one of many **show** and **debug** examples you will see throughout this book. Cisco IOS also has a CLI feature that allows a router to monitor events and automatically

respond to a specific event (such as a defined threshold being reached) with a predefined action. This feature is called Cisco IOS Embedded Event Manager (EEM), which we cover in more detail later.

GUI Tools

Although Cisco has a great number of GUI tools, when it comes to router and switch configuration and troubleshooting for the CCNP Routing and Switching track, you will spend all your time in the CLI. Therefore, do not get too comfortable with GUI tools for the Routing and Switching track. However, as an example, you can use the GUI tool known as Cisco Configuration Professional (CCP) to configure and troubleshoot your Integrated Services Routers (ISRs). Figure 2-1 provides a sample of the CCP home page.

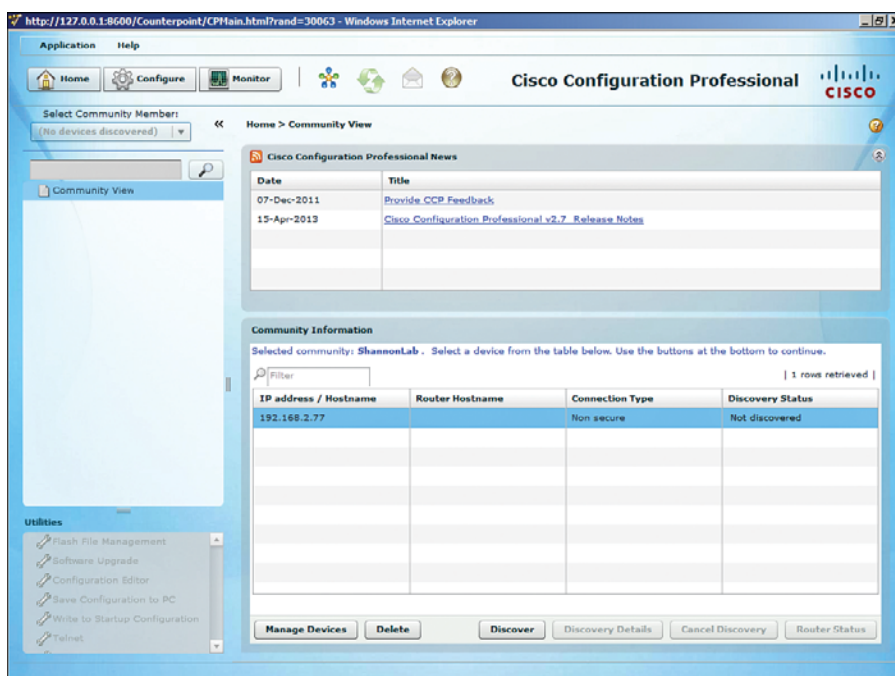


Figure 2-1 *Cisco Configuration Professional*

Recovery Tools

During the recovery process, you need access to duplicate hardware and the IOS. However, you also need a backup of the failed devices configurations. External servers are often used to store archival backups of a device's operating system (for example, a Cisco IOS image) and the configuration information. Depending on your network device, you might be able to back up your operating system and configuration information to a TFTP, FTP, HTTP, or SCP server. To illustrate, consider Example 2-2.


Example 2-2 *Backing Up a Router's Startup Configuration to an FTP Server*

```
R1#copy startup-config ftp://cisco:cisco@192.168.1.74
Address or name of remote host [192.168.1.74]?
Destination filename [r1-config]?
Writing r1-config !
1446 bytes copied in 3.349 secs (432 bytes/sec)
```

In Example 2-2, router R1's startup configuration is being copied to an FTP server with an IP address of 192.168.1.74. Notice that the login credentials (that is, username=cisco and password=cisco) for the FTP server are specified in the **copy** command. In a production environment, the username and password should be stronger and not easily guessed.

If you intend to routinely copy backups to an FTP server, you can avoid specifying the login credentials each time (for security purposes), by adding those credentials to the router's configuration. Example 2-3 shows how to add FTP username and password credentials to the router's configuration, and Example 2-4 shows how the startup configuration can be copied to an FTP server without explicitly specifying those credentials in the **copy** command.

Example 2-3 *Adding FTP Server Login Credentials to a Router's Configuration*

```
R1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#ip ftp username cisco
R1(config)#ip ftp password cisco
R1(config)#end
```

Example 2-4 *Backing Up a Router's Startup Configuration to an FTP Server Without Specifying Login Credentials*

```
R1#copy startup-config ftp://192.168.1.74
Address or name of remote host [192.168.1.74]?
Destination filename [r1-config]?
Writing r1-config !
1446 bytes copied in 3.389 secs (427 bytes/sec)
```

Example 2-5 shows how to add HTTP username and password credentials to the router's configuration. Compare this to the FTP configuration commands and notice the difference.

Example 2-5 *Adding HTTP Server Login Credentials to a Router's Configuration*

```
R1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#ip http client username cisco
R1(config)#ip http client password cisco
R1(config)#end
```

The process of backing up a router's configuration can be automated using an archiving feature, which is part of the Cisco IOS Configuration Replace and Configuration Rollback feature. Specifically, you can configure a Cisco IOS router to periodically (that is, at intervals specified in minutes) back up a copy of the configuration to a specified location (for example, the router's flash, or an FTP server). Also, the archive feature can be configured to create an archive every time you copy a router's running configuration to the startup configuration.

Example 2-6 illustrates a router configured to back up the running configuration every 1440 minutes to an FTP server with an IP address of 192.168.1.74. The login credentials have already been configured in the router's configuration. In addition, the **write-memory** command causes the router to archive a copy of the configuration whenever the router's running configuration is copied to the startup configuration using either the **write-memory** or **copy running-config startup-config** commands.

Example 2-6 Automatic Archive Configuration

```
R1#show run
Building configuration...
...OUTPUT OMITTED...
ip ftp username cisco
ip ftp password cisco
!
archive
  path ftp://192.168.1.74/R1-config
  write-memory
  time-period 1440
...OUTPUT OMITTED...
```

You can view the files stored in a configuration archive by issuing the **show archive** command, as demonstrated in Example 2-7.



Example 2-7 Viewing a Configuration Archive

```
R1#show archive
The maximum archive configurations allowed is 10.
The next archive file will be named ftp://192.168.1.74/R1-config-3

Archive #   Name
1          ftp://192.168.1.74/R1-config-1
2          ftp://192.168.1.74/R1-config-2 <- Most Recent
3
4
5
6
7
8
9
10
```

Example 2-8 shows the execution of the **copy run start** command, which copies a router's running configuration to the router's startup configuration. The **show archive** command is then reissued, and the output confirms that an additional configuration archive (named R1-config-3) has been created on the FTP server because of the **write-memory** command we issued in config-archive configuration mode.

Example 2-8 *Confirming Automated Backups*

```
R1#copy run start
Destination filename [startup-config]?
Building configuration...
[OK]
Writing R1-config-3 !
R1#show archive
The maximum archive configurations allowed is 10.
The next archive file will be named ftp://192.168.1.74/R1-config-4
Archive #  Name
1         ftp://192.168.1.74/R1-config-1
2         ftp://192.168.1.74/R1-config-2
3         ftp://192.168.1.74/R1-config-3 <- Most Recent
4
5
6
7
8
9
10
```

The output of **show archive** indicates that the maximum configurations allowed is ten. This is not entirely true. Because the path is pointing to an FTP server, we are limited only by the amount of storage space on the server. Therefore, the router will continue to create an archive of the running configuration at its scheduled interval. If the archive list on the router fills up (maximum ten), the output of **show archive** will erase the entry for Archive 1, move all entries up the list one spot, and add the new entry to Archive 10, as shown in Example 2-9. Note that this does not delete anything from the FTP server. Only the entry in **show archive** is removed to make space in the list.

Example 2-9 *Confirming Archive Configuration*

```
R1#copy run start
Destination filename [startup-config]?
Building configuration...
[OK]
Writing R1-config-3 !
R1#show archive
The maximum archive configurations allowed is 10.
The next archive file will be named ftp://192.168.1.74/R1-config-4
Archive #  Name
```



```

1      ftp://192.168.1.74/R1-config-7
2      ftp://192.168.1.74/R1-config-8
3      ftp://192.168.1.74/R1-config-9
4      ftp://192.168.1.74/R1-config-10
5      ftp://192.168.1.74/R1-config-11
6      ftp://192.168.1.74/R1-config-12
7      ftp://192.168.1.74/R1-config-13
8      ftp://192.168.1.74/R1-config-14
9      ftp://192.168.1.74/R1-config-15
10     ftp://192.168.1.74/R1-config-16 <- Most Recent

```

However, if you are storing the archive locally in flash as an example, the older files will be deleted to make space, in addition to moving the entries listed in the **show archive** command output. You can change the maximum number of archives with the **maximum** command in config-archive configuration mode.



Restoring a configuration backup requires copying the configuration file from its storage location to the running configuration on the router or switch. The Cisco IOS **copy** command treats this as a merge operation instead of a copy and replace operation. This means that copying anything into the running configuration from any source might not produce the result we desire. We can witness this with the password recovery process on a Cisco router. During this process, after you have loaded the router to factory defaults, you copy the startup configuration into the running configuration, which produces a merge. This merge is easily witnessed with the interfaces. Interfaces that were enabled do not have a **no shutdown** command in the startup configuration, and the factory default setting of a router interface is shutdown and includes a **shutdown** command. This is illustrated in Example 2-10.

Example 2-10 *Comparing the Running Configuration and Startup Configuration Before Issuing the copy Command*

```

R1#show run
...OUTPUT OMITTED...
interface FastEthernet0/0
  no ip address
  shutdown
...OUTPUT OMITTED...
R1#show start
...OUTPUT OMITTED...
interface FastEthernet0/0
  ip address 192.168.1.11 255.255.255.0
...OUTPUT OMITTED...

```

Once the startup configuration is copied to (merged with) the running configuration, the **shutdown** command prevails in the running configuration because there is not a **no shutdown** in the startup configuration that will overwrite that, as shown in Example 2-11. To fix this, after you have copied the startup configuration to the running configuration, you have to issue the **no shutdown** command on all interfaces you want enabled.

Example 2-11 *Witnessing a Configuration Merge*

```

R1#copy start run
Destination filename [running-config]?
1881 bytes copied in 1.444 secs (1303 bytes/sec)

R1#show run
...OUTPUT OMITTED...
interface FastEthernet0/0
  ip address 192.168.1.11 255.255.255.0
  shutdown
...OUTPUT OMITTED...
R1#

```



On the bright side, you can restore a previously archived configuration using the **configure replace** command. Unlike the **copy** command, this does not merge the archived configuration with the running configuration, but rather completely replaces the running configuration with the archived configuration. Example 2-12 shows the restoration of an archived configuration to a router. Notice how the IOS warns you that this is a copy replace function that completely overwrites the current configuration. In this case, there was only one small difference between the running configuration and the archive, as indicated by the statement “Total number of passes: 1.” It was the hostname.

Example 2-12 *Restoring an Archived Configuration*

```

Router#configure replace ftp://192.168.1.74/R1-config-3
This will apply all necessary additions and deletions
to replace the current running configuration with the
contents of the specified configuration file, which is
assumed to be a complete configuration, not a partial
configuration. Enter Y if you are sure you want to proceed. ? [no]: Y
Loading R1-config-3 !
[OK - 3113/4096 bytes]

Total number of passes: 1
Rollback Done

R1#

```

Logging Tools

Device logs offer valuable information when troubleshooting a network issue. Many events that occur on a router are automatically reported to the router’s console. For example, if a router interface goes down or up, a message is written to the console. However, once in production, we are usually not staring at the console output or even connected to the console port. In most cases, we would connect to the device when needed using Telnet or Secure Shell (SSH), and these logging messages are not displayed via Telnet or

SSH by default. If you are connected to a router through Telnet or SSH and want to see console messages, you have to enter the command **terminal monitor** in privilege EXEC mode.

A downside of solely relying on console messages is that those messages can scroll off the screen, or you might close your terminal emulator, after which those messages would no longer be visible as the session is reset. Therefore, a step beyond logging messages to the console is logging messages to a router's buffer (the router's RAM). To cause messages to be written to a router's buffer, you can issue the **logging buffered** command. As part of that command, you can specify how much of the router's RAM can be dedicated to logging. After the buffer fills to capacity, older entries will be deleted to make room for newer entries. You can view the logging messages in the buffer by issuing the **show logging** command. If you need to clear the logging messages in the buffer, issue the **clear logging** command in privilege EXEC mode.

Logging severity levels range from 0 to 7, with corresponding names, as shown in Table 2-2. Notice that lower severity levels are more severe than those with higher levels. By default, the console, vty lines, and buffer will log all messages with a severity level of 7 and lower. However, debugs are logged only when they are turned on with **debug** commands.



Table 2-2 *Severity Levels*

Severity Level	Name
0	Emergencies
1	Alerts
2	Critical
3	Errors
4	Warnings
5	Notifications
6	Informational
7	Debugging

You might want to log messages of one severity level to a router's console and messages of another severity level to the router's buffer. This is possible by using the **logging console severity_level** and **logging buffered severity_level** commands. For example, if you want to log level 6 and lower to the console and level 7 and lower to the buffer, you enter **logging console 6** and **logging buffered 7** in global configuration mode. You can also specify the severity level by name instead of number.

Another logging option is to log messages to an external syslog server. By sending log messages to an external server, you can keep a longer history of logging messages. Depending on the syslog server software, you might be able to schedule automated log archiving, configure advanced script actions, create advanced alerts, and produce statistical graphs. You can direct your router's log output to a syslog server's IP address using

the **logging ip_address** command, and you can specify the severity level that will be sent to the syslog server by using the **logging trap severity_level** command.

Example 2-13 illustrates several of the logging configurations discussed here.



Example 2-13 Logging Configuration

```
R1#show run
...OUTPUT OMITTED...
Building configuration...
!
logging buffered 4096 warnings
logging console warnings
!
logging 192.168.1.50
logging trap 6
...OUTPUT OMITTED...
```

In Example 2-13, events with a severity level of *warning* (that is, 4) or less (that is, 0 to 4) are logged to the router's buffer. This buffer can be viewed with the **show logging** command. The router can use a maximum of 4096 bytes of RAM for the buffered logging. The console is configured for logging events of the same severity level. In addition, the router is configured to log messages with a severity of 6 or lower to a syslog server with an IP address 192.168.1.50. Figure 2-2 shows logging messages being collected by a Kiwi Syslog Server (available from <http://www.kiwisyslog.com>).

Date	Time	Priority	Hostname	Message
07-03-2009	15:20:07	Local7:Warning	192.168.1.11	222: *Mar 3 09:41:12.969: ZCDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet0/0 (not half duplex), with R1 FastEthernet0/0 (half duplex).
07-03-2009	15:27:44	Local7:Notice	192.168.1.11	221: *Mar 3 09:40:50.132: ZSYS-5-CONFIG_I: Configured from console by console
07-03-2009	15:27:26	Local7:Info	192.168.1.11	220: *Mar 3 09:40:32.142: ZSYS-6-LOGGINGHOST_STARTSTOP: Logging to host 192.168.1.50 started - CLI initiated
07-03-2009	15:27:25	Local7:Notice	192.168.1.11	219: *Mar 3 09:40:31.140: ZSYS-5-CONFIG_I: Configured from console by console

Figure 2-2 Syslog Server

Network Time Protocol as a Tool

Picture this scenario. You have just been assigned a trouble ticket. Users are complaining that the network is slow at 5:30 p.m. local time. The problem ticket indicates that this happens every day. You are browsing the logs to see whether anything abnormal is occurring on the network at that time. However, your search will be worthwhile only if the logs have time stamps. If they don't, you will not be able to correlate the log entries to the problem the users are reporting. Therefore, time stamps are useless if they are not accurate. For example, there may be a log entry for 2:25 p.m. that reports high network utilization. Is that really 2:25 p.m. or is it 5:30 p.m.? Time-stamp accuracy is paramount when it comes to troubleshooting. Therefore, you need to make sure the clocks are set correctly on all the devices.

Although you could individually set the clock on each of your devices, those clocks might drift over time and not agree causing variations in the log entries. You might have heard the saying that a man with one watch always knows what time it is, whereas a man with two watches is never quite sure. This implies that devices need to have a common point of reference for their time. Such a reference point is made possible by Network Time Protocol (NTP), which allows network devices to point to a device acting as an NTP server (a time source). However, this must be a reliable time source. For example, the U.S. Naval Observatory in Washington, D.C., is a stratum 1 time source. Stratum 1 time sources are the most reliable and accurate. In addition, because the NTP server might be referenced by devices in different time zones, each device has its own time zone configuration, which indicates how many hours its time zone differs from Greenwich mean time (GMT).



Example 2-14 shows an NTP configuration entered on a router located in the eastern time zone, which is 5 hours behind GMT when daylight savings time is not in effect. The **clock summer-time** command defines when daylight savings time begins and ends. In this example, daylight savings time begins at 2:00 a.m. on the second Sunday in March and ends at 2:00 a.m. on the first Sunday in November. The **ntp server** command is used to point to an NTP server. Note that a configuration can have more than one **ntp server** command, for redundancy. In such cases, NTP will decide based on its protocol which is the most reliable, or you can manually specify which is most reliable by adding the **prefer** option to the **ntp server** command.

Example 2-14 *Configuring a Router to Point to an NTP Server*

```
R1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#clock timezone EST -5
R1(config)#clock summer-time EDT recurring 2 Sun Mar 2:00 1 Sun Nov 2:00
R1(config)#ntp server 192.168.1.150
R1(config)#ntp server 192.168.1.151 prefer
R1(config)#end
```

NTP uses a hierarchy of time servers based on stratum levels from 1 to 15. Stratum 1 is the most reliable. Because it is based on a hierarchy, you may not want all of your devices pointing to the stratum 1 time source that is connected to the Internet. In these instances, you could set up a device or two in your organization to receive their time from the stratum 1 source (making them a stratum 2 source) and then configure the other devices in your organization to receive their time from these local devices in your organization (making them a stratum 3).

Advanced Tools

Keeping an eye on network traffic patterns and performance metrics can help you anticipate problems before they occur. You can then take the necessary measures to address them proactively before they become a major issue. This is in contrast to taking a reactive stance where you continually respond to problem reports as they occur. The saying “If it ain’t broke don’t fix it” does not apply in a proactive network maintenance environment. Your stance in this type of environment should be “If it appears that it will break, fix it.” To be proactive, you need more than just basic **show** and **debug** commands. You need advanced tools to proactively monitor the health of your devices and the health of your network traffic, such as SNMP, NetFlow, and EEM.

Overview of SNMP and NetFlow

Simple Network Management Protocol (SNMP) allows a monitored device (for example, a router or a switch) to run an SNMP agent that collects data such as utilization statistics for processors and memory. An SNMP server can then query the SNMP agent to retrieve those statistics to determine the overall health of that device.

Cisco IOS NetFlow can provide you with tremendous insight into your network traffic patterns. Several companies market NetFlow collectors, which are software applications that can take the NetFlow information reported from a Cisco device and convert that raw data into useful graphs, charts, and tables reflecting traffic patterns. Reasons to monitor network traffic include the following:

- **Ensuring compliance with an SLA:** If you work for a service provider or are a customer of a service provider, you might want to confirm that performance levels to and from the service provider’s cloud are conforming to the agreed-upon service level agreement (SLA).
- **Trend monitoring:** Monitoring resource utilization on your network (for example, bandwidth utilization and router CPU utilization) can help you recognize trends and forecast when upgrades will be required or if users are abusing the network resources.
- **Troubleshooting performance issues:** Performance issues can be difficult to troubleshoot in the absence of a baseline. By routinely monitoring network performance, you have a reference point (that is, a baseline) against which you can compare performance metrics collected after a user reports a performance issue.

Creating a Baseline with SNMP and NetFlow



SNMP and NetFlow are two technologies available on most Cisco IOS platforms that can automate the collection statistics. These statistics can be used, for example, to establish a baseline that can be used in a troubleshooting scenario or in proactive network management and maintenance. Table 2-3 contrasts these two technologies.

Table 2-3 *Comparing SNMP and NetFlow*

Technology	Characteristics
SNMP	Collects device statistics (for example, platform resource utilization, traffic counts, and error counts) Uses a pull model (that is, statistics pulled from a monitored device by a network management station [NMS]) Available on nearly all enterprise network devices
NetFlow	Collects detailed information about traffic flows Uses a push model (that is, statistics pushed from the monitored device to a NetFlow collector) Available on routers and high-end switches

Although both SNMP and NetFlow are useful for statistical data collection, they target different fundamental functions. For example, SNMP is primarily focused on device statistics (the health of a device), whereas NetFlow is primarily focused on traffic statistics (the health of network traffic).

SNMP

A device being managed by SNMP runs a process called an *SNMP agent*, which collects statistics about the device and stores those statistics in a Management Information Base (MIB). A network management system (NMS) can then query the agent for information in the MIB, using the SNMP protocol. SNMP Version 3 (SNMPv3) supports encryption and hashed authentication of SNMP messages. Before SNMPv3, the most popular SNMP version was SNMPv2c, which used *community strings* for authentication. Today, many SNMP deployments are still using version 2c because of its simplicity. Specifically, for an NMS to be allowed to read data from a device running an SNMP agent, the NMS must be configured with a community string that matches the managed device's read-only community string. For the NMS to change the information on the managed device, the NMS must be configured with a community string that matches the managed device's read-write community string. To enhance the security available with SNMPv2c, you can create an access list that determines valid IP addresses or network addresses for NMS servers that are allowed to manage or collect information from the MIB of the device.

Figure 2-3 shows a topology using SNMP. In the topology, router R1 is running an SNMP agent that the NMS server can query.

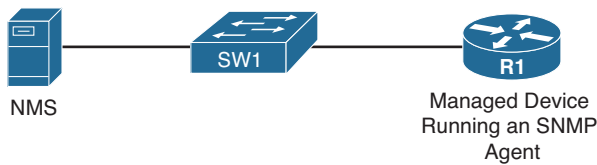


Figure 2-3 *SNMP Sample Topology*

Example 2-15 illustrates the SNMPv2c configuration on router R1. The **snmp-server community string [ro | rw] [access_list_number]** commands specify a read-only (that is, **ro**) community string of **CISCO** and a read-write (that is, **rw**) community string of **PRESS**. Only NMSs permitted in access list 10 and 11 will be able to read, or read/write, respectively, this device using SNMP. Contact and location information for the device is also specified. Finally, notice the **snmp-server ifindex persist** command. This command ensures that the SNMP interface index stays consistent during data collection, even if the device is rebooted. This consistency is important when data is being collected for baselining purposes.

Example 2-15 *SNMP Sample Configuration*

```

R1#configure terminal
R1(config)#snmp-server community CISCO ro 10
R1(config)#snmp-server community PRESS rw 11
R1(config)#snmp-server contact demo@ciscopress.local
R1(config)#snmp-server location 3rd Floor of Lacoste Building
R1(config)#snmp-server ifindex persist
  
```

NetFlow

NetFlow can distinguish between different traffic flows. A *flow* is a series of packets, all of which have shared header information such as source and destination IP addresses, protocol numbers, port numbers, and type of service (TOS) field information. In addition, they are entering the same interface on the device. NetFlow can keep track of the number of packets and bytes observed in each flow. This information is stored in a *flow cache*. Flow information is removed from a flow cache if the flow is terminated, times out, or fills to capacity.

You can use the NetFlow feature as a standalone feature on an individual router. Such a standalone configuration might prove useful for troubleshooting because you can observe flows being created as packets enter a router. However, rather than using just a standalone implementation of NetFlow, you can export the entries in a router's flow cache to a *NetFlow collector*, which is a software application running on a computer/server in your network. After the NetFlow collector has received flow information over a period of time, analysis software running on the NetFlow collector can produce reports detailing traffic statistics.

Figure 2-4 shows a sample topology in which NetFlow is enabled on router R4, and a NetFlow collector is configured on a PC at IP address 192.168.1.50.

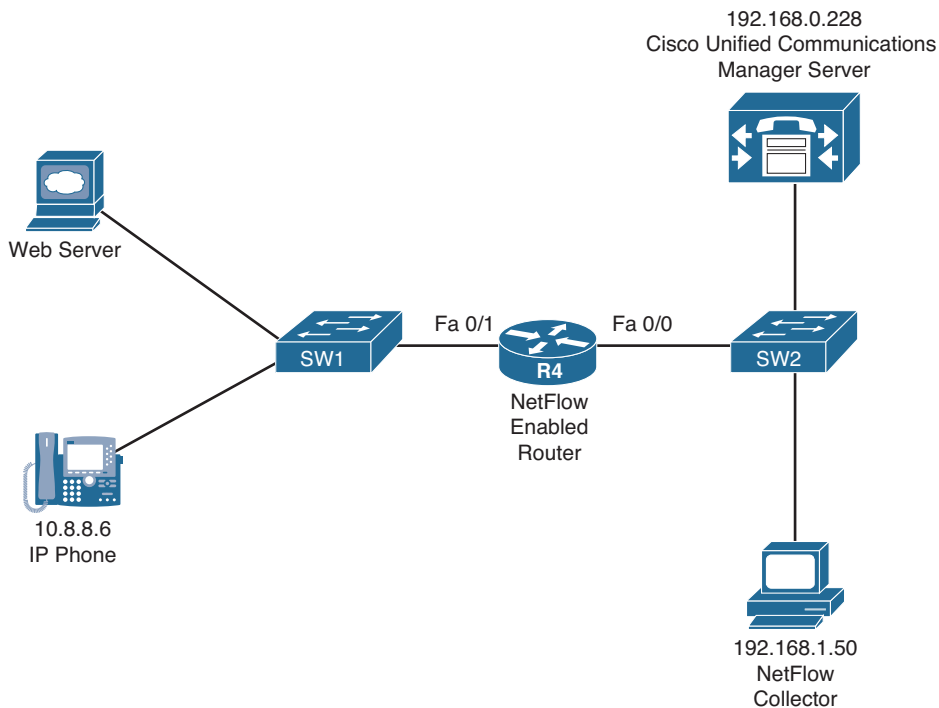


Figure 2-4 *NetFlow Sample Topology*

Example 2-16 illustrates the NetFlow configuration on router R4. Notice that the **ip flow ingress** command is issued for both the Fast Ethernet 0/0 and Fast Ethernet 0/1 interfaces. This ensures that all flows passing through the router, regardless of direction, can be monitored. Although not required, router R4 is configured to report its NetFlow information to a NetFlow collector at IP address 192.168.1.50. The **ip flow-export source lo 0** command indicates that all communication between router R4 and the NetFlow collector will be via interface Loopback 0. A NetFlow Version of 5 was specified. You should check the documentation for your NetFlow collector software to confirm which version to configure. Finally, the **ip flow-export destination 192.168.1.50 5000** command is issued to specify that the NetFlow collector's IP address is 192.168.1.50, and communication to the NetFlow collector should be done over UDP port 5000. Because NetFlow does not have a standardized port number, check your NetFlow collector's documentation when selecting a port.

Example 2-16 *NetFlow Sample Configuration*

```

R4#configure terminal
R4(config)#int fa 0/0
R4(config-if)#ip flow ingress
R4(config-if)#exit
R4(config)#int fa 0/1
R4(config-if)#ip flow ingress
  
```

```

R4(config-if)#exit
R4(config)#ip flow-export source lo 0
R4(config)#ip flow-export version 5
R4(config)#ip flow-export destination 192.168.1.50 5000
R4(config)#end

```

Using your favorite search engine, search for images of “NetFlow collector” (without the quotes) to see various sample images of what a NetFlow collector can provide you. Although an external NetFlow collector is valuable for longer-term flow analysis and can provide detailed graphs and charts, you can issue the **show ip cache flow** command at a router’s CLI prompt to produce a summary of flow information, as shown in Example 2-17. A troubleshooter can look at the output displayed in Example 2-17 and be able to confirm, for example, that traffic is flowing between IP address 10.8.8.6 (a Cisco IP Phone) and 192.168.0.228 (a Cisco Unified Communications Manager server).

Example 2-17 Viewing NetFlow Information

```

R4#show ip cache flow
...OUTPUT OMITTED...

```

Protocol	Total	Flows	Packets	Bytes	Packets	Active (Sec)	Idle (Sec)
-----	Flows	/Sec	/Flow	/Pkt	/Sec	/Flow	/Flow
TCP-Telnet	12	0.0	50	40	0.1	15.7	14.2
TCP-WWW	12	0.0	40	785	0.1	7.1	6.2
TCP-other	536	0.1	1	55	0.2	0.3	10.5
UDP-TFTP	225	0.0	4	59	0.1	11.9	15.4
UDP-other	122	0.0	114	284	3.0	15.9	15.4
ICMP	41	0.0	13	91	0.1	49.9	15.6
IP-other	1	0.0	389	60	0.0	1797.1	3.4
Total:	949	0.2	18	255	3.8	9.4	12.5

SrcIf	SrcIPAddress	DstIf	DstIPAddress	Pr	SrcP	DstP	Pkts
Fa0/0	10.3.3.1	Null	224.0.0.10	58	0000	0000	62
Fa0/1	10.8.8.6	Fa0/0	192.168.0.228	06	C2DB	07D0	2
Fa0/0	192.168.0.228	Fa0/1	10.8.8.6	06	07D0	C2DB	1
Fa0/0	192.168.1.50	Fa0/1	10.8.8.6	11	6002	6BD2	9166
Fa0/1	10.8.8.6	Fa0/0	192.168.1.50	11	6BD2	6002	9166
Fa0/0	10.1.1.2	Local	10.3.3.2	06	38F2	0017	438

Providing Notifications for Network Events

Whereas responding to problem reports from users is a reactive form of troubleshooting, monitoring network devices for significant events and responding to those events is a

proactive form of troubleshooting. For example, before a user loses connectivity with the Internet, a router that is dual-homed to the Internet might report the event of one of its Internet connections going down. The redundant link can then be repaired, in response to the notification, thus resolving the problem without users being impacted.

Both syslog and SNMP are protocols that can report the occurrence of specific events on a network device, and NetFlow can report events related to network traffic flows. Although these protocols by themselves lack a mechanism to alert a network administrator (for example, via e-mail) when a network event is logged, third-party software is available that can selectively alert appropriate personnel when specific events are logged.

Earlier, this section discussed how a network device running an SNMP agent can be queried for information from an NMS. However, a network device running an SNMP agent can also initiate communication with an NMS. If an interface goes down, for example, the SNMP agent on a managed network device can send a message containing information about the interface state change to an NMS, and then the NMS can notify a network administrator via e-mail. These messages, from the agent to the NMS, are called *traps*. These traps require the NMS to interpret them because they are not in an easy, readable format.



Example 2-18 demonstrates how to enable a router to send SNMP traps to an NMS. The `snmp-server host 192.168.1.150 version 2c CISCOPRESS` command points router R4 to an SNMP server (that is, an NMS) at IP address 192.168.1.150. The SNMP server is configured for SNMP **version 2c** and a community string of **CISCOPRESS**; therefore, we include that information on the router for communication purposes with the NMS.

The `snmp-server enable traps` command is used to enable all traps on the router. If you only need to enable specific traps, you may do so by adding the individual trap keyword to the `snmp-server enable traps` command (for example, `snmp-server enable traps bgp`). You can view the enabled traps by using the `show run | include traps` command.

Example 2-18 *Enabling SNMP Traps*

```
R4#configure terminal
R4(config)#snmp-server host 192.168.1.150 version 2c CISCOPRESS
R4(config)#snmp-server enable traps
R4(config)#end
R4#show run | include traps
snmp-server enable traps snmp authentication linkdown linkup coldstart warmstart
snmp-server enable traps vrrp
snmp-server enable traps ds1
snmp-server enable traps gatekeeper
snmp-server enable traps tty
snmp-server enable traps eigrp
snmp-server enable traps xgcp
snmp-server enable traps ds3
...OUTPUT OMITTED...
```

The messages received via syslog and SNMP are predefined within Cisco IOS. Although this is a rather large collection of predefined messages and should accommodate most network management requirements, Cisco IOS also supports a feature called Embedded Event Manager (EEM) that enables you to create your own event definitions and specify custom responses to those events. An event can be defined and triggered based on a syslog message, SNMP trap, and even the issuing of a specific Cisco IOS command, as just a few examples. In response to a defined event, EEM can perform various actions, including sending an SNMP trap to an NMS, writing a log message to a syslog server, executing specified Cisco IOS commands, capturing output of specific **show** commands, sending an e-mail to an appropriate party, or executing a tool command language (Tcl) script. From this short list, you can already see how powerful the EEM can be.



To illustrate the basic configuration steps involved in configuring an EEM applet, consider Example 2-19. The purpose of this configuration is to create a syslog message that will be displayed on the router console when someone clears the router's interface counters using the **clear counters** command. The message reminds the administrator to update the network documentation and lists the rationale for clearing the interface counters.

Example 2-19 EEM Sample Configuration

```
R4#configure terminal
R4 (config)#event manager applet COUNTER-RESET
R4 (config-applet)#event cli pattern "clear counters" sync no skip no occurs 1
R4 (config-applet)#action A syslog priority informational msg "Please update network
documentation to record why the counters were reset."
R4 (config-applet)#end
```

The **event manager applet COUNTER-RESET** command creates an EEM applet named **COUNTER-RESET** and enters applet configuration mode. The **event** command specifies what you are looking for in your custom-defined event. In this example, you are looking for the CLI command **clear counters**. Note that the **clear counters** command would be detected even if a shortcut (for example, **cle co**) were used. The **sync no** parameter says that the EEM policy will run asynchronously with the CLI command. Specifically, the EEM policy will not be executed before the CLI command executes. The **skip no** parameter says that the CLI command will not be skipped (that is, the CLI command will be executed). Finally, the **occurs 1** parameter indicates that the EEM event is triggered by a single occurrence of the **clear counters** command being issued.

The **action** command is then entered to indicate what should be done in response to the defined event. In Example 2-19, the action is given a locally significant name of **A** and is assigned a syslog priority level of **informational**. The specific action to be taken is producing this informational message saying: **Please update network documentation to record why the counters were reset.**

To verify the operation of the EEM configuration presented in Example 2-19, the **clear counters** command is executed in Example 2-20. Notice that entering the **clear counters** command triggers the custom-defined event, resulting in generation of a syslog message reminding an administrator to document the reason they cleared the interface counters.

Example 2-20 *Testing EEM Configuration*

```
R4#clear counters
Clear "show interface" counters on all interfaces [confirm]
R4#
%HA_EM-6-LOG: COUNTER-RESET: Please update network documentation to record why the
counters were reset.
R4#
```

Cisco Support Tools

Cisco has several other configuration, troubleshooting, and maintenance tools available on its website:

http://www.cisco.com/en/US/support/tsd_most_requested_tools.html

Some of the tools available at this website require login credentials with appropriate privilege levels.

Using Cisco IOS to Verify and Define the Problem

When you receive a trouble ticket, your first couple of tasks should be to verify and define the problem. Some relatively simple tasks can confirm the issue reported and in most cases help to focus your troubleshooting efforts. Three easy-to-use tools built in to the Cisco IOS can help you verify connectivity and further define the problem. They are ping, Telnet, and traceroute. This section discusses how ping, Telnet, and traceroute can verify the problem and help focus our efforts.

Ping

A common command, which you can use to check network connectivity, is the **ping** command. If you recall from Chapter 1, a successful ping indicates that Layer 1, 2, and 3 of the OSI model are functioning, and so you can focus your attention on higher OSI layers. The same holds true in reverse with an unsuccessful ping. If it is unsuccessful, you focus your troubleshooting on the lower layers of the OSI model.

A basic **ping** command sends Internet Control Message Protocol (ICMP) echo messages to a specified destination. For every ICMP echo reply received from that specified destination, an exclamation point appears in the output, as shown in Example 2-21.

Example 2-21 *Basic ping Command*

```
R1#ping 10.4.4.4
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.4.4.4, timeout is 2 seconds:
!!!!
```

The **ping** command does have several options that can prove useful during troubleshooting, including the following:

- **size:** Specifies the number of bytes per datagram (defaults to 100 bytes on Cisco IOS)
- **repeat:** Specifies the number of ICMP echo messages sent (defaults to 5)
- **timeout:** Specifies the number of seconds to wait for an ICMP echo reply (defaults to 2)
- **source:** Specifies the source of the ICMP echo datagrams
- **df-bit:** Sets the do not fragment bit in the ICMP echo datagram

Not only can a **ping** command indicate that a given IP address is reachable, but the response to a **ping** command might provide insight into the nature of a problem. For example, if the ping results indicate alternating failures and successes (that is, !.!.!), a troubleshooter might conclude that traffic is being load balanced between the source and destination IP addresses. Traffic flowing across one path is successful, whereas traffic flowing over the other path is failing.

You can also use the **ping** command to create a load on the network to troubleshoot the network under heavy use. For example, you can specify a datagram size of 1500 bytes, along with a large byte count (repeat value) and a timeout of 0 seconds, as shown in Example 2-22.

Notice that all the pings failed. These failures occurred because of the 0-second timeout. The router did not wait before considering the ping to have failed and sending another ICMP echo message. Remember, in this case, we do not care that it failed; we are doing this for the artificial load generated for testing purposes.

Example 2-22 *Creating a Heavy Load on the Network*

```
R1#ping 10.4.4.4 size 1500 repeat 9999 timeout 0

Type escape sequence to abort.
Sending 9999, 1500-byte ICMP Echos to 10.4.4.4, timeout is 0 seconds:
.....
.....
.....
...OUTPUT OMITTED...
```

Perhaps you suspect that an interface has a nondefault maximum transmission unit (MTU) size, which is commonly seen with Q-n-Q tunnels, generic routing encapsulation (GRE) tunnels, and even Point-to-Point Protocol over Ethernet (PPPoE) interfaces. To verify your suspicion, you could send ICMP echo messages across that interface using the **df-bit** and **size** options of the **ping** command to specify the size of the datagram to be sent. The **df-bit** option instructs a router to drop this datagram rather than fragmenting it if fragmentation is required.

Example 2-23 shows the sending of pings with the do not fragment bit set. Notice the **M** in the ping responses, which indicates that fragmentation was required but could not be performed because the do not fragment bit was set. Therefore, you can conclude that a link between the source and destination is using a nonstandard MTU (that is, an MTU less than 1500 bytes).

Telnet

Key Topic

As you just read, the **ping** command is useful for testing Layer 3 (that is, the network layer) connectivity. The **telnet** command is useful for troubleshooting Layer 4 (that is, the transport layer) and Layer 7 (that is, the application layer). By default, Telnet uses TCP port 23; however, you can specify an alternate port number to see whether a particular TCP Layer 4 service is running at a destination IP address. Such an approach might prove useful if you are using a divide-and-conquer approach, starting at Layer 3 (which was determined to be operational as a result of a successful ping), or a bottom-up approach (which has also confirmed Layer 3 to be operational). At this point, you could use telnet to test the transport layer.

To illustrate, notice the **telnet 192.168.1.50 80** command issued in Example 2-25. This command causes router R1 to attempt a TCP connection with 192.168.1.50 using port 80 (the HTTP port). The response of Open indicates that 192.168.1.50 is indeed running a service on port 80.

Example 2-25 Using Telnet to Test the Transport Layer (Success)

```
R1#telnet 192.168.1.50 80
Trying 192.168.1.50, 80 ... Open
```

Let's consider a situation where users indicate that they are unable to connect to the mail server at 192.168.1.51. The mail server uses SMTP port 25. The result of using Telnet to test the transport layer shows that port 25 is not responding on the mail server as shown in Example 2-26. Therefore, you may want to start by checking whether the server is operational and verifying that no access control lists (ACLs) are denying connectivity to port 25.

Example 2-26 Using Telnet to Test the Transport Layer (Failure)

```
R1#telnet 192.168.1.51 25
Trying 192.168.1.51, 25 ...
% Connection refused by remote host
```

Traceroute

Key Topic

The **traceroute** command provides valuable information during the troubleshooting process. The first is verified connectivity. If the trace completes successfully, we have verified Layer 3 connectivity, which is what the **ping** command provides us. The second valuable piece of information is the path that the trace took through the network. This is something that the **ping** command does not provide. Therefore, if we issue the command **ping 10.4.4.4** and it fails, we could then issue the **traceroute 10.4.4.4** command to get an idea of where the ping is failing. Example 2-27 displays the output of a successful trace to the router that has the IP address 10.4.4.4.

Example 2-27 *Using Traceroute*

```

R1#traceroute 10.4.4.4
Type escape sequence to abort.
Tracing the route to 10.4.4.4
VRF info: (vrf in name/id, vrf out name/id)
 1 10.1.1.2 24 msec 44 msec 28 msec
 2 10.1.2.2 24 msec 64 msec 36 msec
 3 10.1.3.2 64 msec 52 msec 84 msec
 4 10.1.4.4 100 msec * 72 msec

```

Example 2-28 shows an unsuccessful ping from R1 to 10.4.4.4. We then use **traceroute** to get a better picture of where this ping is failing so we can focus our attention around that part of the network.

Example 2-28 *Using Traceroute to Follow The Path*

```

R1#ping 10.4.4.4
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.4.4.4, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
R1#traceroute 10.4.4.4
Type escape sequence to abort.
Tracing the route to 10.4.4.4
VRF info: (vrf in name/id, vrf out name/id)
 1 10.1.1.2 44 msec 36 msec 44 msec
 2 10.1.2.2 68 msec 88 msec 88 msec
 3 * * *
 4 * * *
 5 * * *
 6 * * *
...OUTPUT OMITTED...

```

If you see a repeating pattern of IP addresses in the output of traceroute (for example, 10.1.2.2, 10.1.3.2, 10.1.2.2, 10.1.3.2, 10.1.2.2, 10.1.3.2), you have a routing loop.

Using Cisco IOS to Collect Information

After a problem has been clearly defined, the first step in diagnosing that problem is collecting information, as described in Chapter 1. Because the collection of information can be one of the most time-consuming of the troubleshooting processes, the ability to quickly collect appropriate information becomes a valuable troubleshooting skill. Would you prefer to search for the needle in a haystack by moving one piece of straw at a time, or would you prefer to use the biggest strongest magnet in the world and attract the needle out of the haystack? I choose the magnet. You do not want to spend your time looking for the needle in a haystack. Time is valuable. This section introduces basic Cisco

IOS commands useful in gathering information and discusses the filtering of irrelevant information from the output of those commands. Also included in this section are commands helpful in diagnosing connectivity and hardware issues.

Filtering the Output of show Commands

Cisco IOS offers multiple **show** commands and **debug** commands that are useful for gathering information. Throughout this book, you will be introduced to a considerable number of **show** and **debug** commands. However, many of these commands produce a large quantity of output.

Consider the output shown in Example 2-29. The output from the **show processes cpu** command generated approximately 180 lines of output, making it challenging to pick out a single process.

Example 2-29 *show processes cpu Command Output*

```
R1#show processes cpu
CPU Utilization for five seconds: 0%/0%; one minute: 0%; five minute: 0%
  PID  Runtime(ms)   Invoked    uSecs    5Sec    1Min    5Min  TTY process
    1         4         3      1333    0.00%    0.00% 0.00% 0 Chunk Manager
    2      7245      1802      4020    0.08%    0.08% 0.08% 0 Load Meter
    3        56      2040        27    0.00%    0.00% 0.00% 0 OSPF Hello 1
    4         4         1      4000    0.00%    0.00% 0.00% 0 EDDRI_MAIN
    5     21998      1524     14434    0.00%    0.32% 0.25% 0 Check heaps
    6         0         1         0    0.00%    0.00% 0.00% 0 Pool Manager
    7         0         2         0    0.00%    0.00% 0.00% 0 Timers
    8         0         1         0    0.00%    0.00% 0.00% 0 Crash Writer
    9         0        302         0    0.00%    0.00% 0.00% 0 Environmental mo
   10       731      1880       388    0.00%    0.00% 0.00% 0 APR Input
...OUTPUT OMITTED...
 171         0         1         0    0.00%    0.00% 0.00% 0 lib_off_app
 172         4         2      2000    0.00%    0.00% 0.00% 0 Voice Player
 173         0         1         0    0.00%    0.00% 0.00% 0 Media Record
 174         0         1         0    0.00%    0.00% 0.00% 0 Resource Measure
 175        12         6      2000    0.00%    0.00% 0.00% 0 Session Applicat
 176        12       151        79    0.00%    0.00% 0.00% 0 RTPSPI
 177         4     17599         0    0.00%    0.00% 0.00% 0 IP NAT Ager
 178         0         1         0    0.00%    0.00% 0.00% 0 IP NAT WALN
 179         8       314        25    0.00%    0.00% 0.00% 0 CEF Scanner
```

Perhaps you were only looking for CPU utilization statistics for the Check heaps process. Because you know that the content of the one line you are looking for contains the text **Check heaps**, you could take the output of the **show processes cpu** command and pipe

that output (that is, use the `|` character) to the **include Check heaps** statement. The piping of the output causes the output to be filtered to only include lines that include the text **Check heaps**, as demonstrated in Example 2-30. This type of filtering can help troubleshooters more quickly find the data they are looking for. However, realize the information you are looking for is case sensitive. Therefore, **check heaps** is not the same as **Check heaps**.

Example 2-30 *Filtering the show processes cpu Command Output*

```
R1#show processes cpu | include Check heaps
5          24710          1708          14467  1.14%  0.26%  0.24%  0 Check heaps
```

Example 2-30 gave us some interesting values; but what do they mean? If you go back to Example 2-29, you will notice column headers that were omitted in Example 2-30. Therefore, we have to tweak our command so that we can receive the column headers as shown in Example 2-31. Notice that when specifying the additional pipes (`|`) there is no space because it is an “or” operation.

Example 2-31 *Filtering the show processes cpu Command Output with Column Headers*

```
R1#show processes cpu | include Check heaps|^CPU|^ PID
CPU utilization for five seconds: 3%/100%; one minute: 4%; five minutes: 4%
PID Runtime(ms)          Invoked      uSecs   5Sec   1Min   5Min  TTY Process
5          24710          1708          14467  1.14%  0.26%  0.24%  0 Check heaps
```

In Example 2-31 we modified the **show processes cpu | include Check heaps** command to include **|^CPU|^ PID**. The `^` is a regular expression that represents “begins with.” Therefore, these additions state to include any line that begins with CPU or (space)PID. Now those interesting values have meaning because the column headers are included.

In addition, with the **show processes cpu** command, you can sort by 5-second, 1-minute, and 5-minute utilization with the **sorted** parameter. This allows you to place in descending order those processes that are consuming the most CPU resources.

Similar to piping output to the **include** option, you could alternatively pipe output to the **exclude** option. The **exclude** option can display all lines of the output except lines containing the string you specify. For example, the **show ip interfaces brief** command can display IP addresses and interface status information for interfaces on a router and switch, as shown in Example 2-32.

Example 2-32 *show ip interface brief Command Output*

```
R1#show ip interface brief
```

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	192.168.1.11	YES	NVRAM	up	up
Serial0/0	unassigned	YES	NVRAM	administratively down	down
FastEthernet0/1	192.168.0.11	YES	NVRAM	up	up
Serial0/1	unassigned	YES	NVRAM	administratively down	down
NVI0	unassigned	YES	unset	up	up
Loopback0	10.1.1.1	YES	NVRAM	up	up

Notice in Example 2-32 that some of the interfaces have an IP address of unassigned. If you want to only view information pertaining to interfaces with assigned IP addresses, you can pipe the output of the **show ip interface brief** command to **exclude unassigned**, as illustrated in Example 2-33.

Example 2-33 *Filtering Output from the show ip interface brief Command Using exclude*

```
R1#show ip interface brief | exclude unassigned
```

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	192.168.1.11	YES	NVRAM	up	up
FastEthernet0/1	192.168.0.11	YES	NVRAM	up	up
Loopback0	10.1.1.1	YES	NVRAM	up	up

As another example, you might be troubleshooting an OSPF routing protocol issue and want to see the section of your running configuration where the routing protocol configuration begins. Piping the output of the **show running-config** command to **begin router**, as shown in Example 2-34, skips the initial portion of the **show running-config** output and begins displaying the output where the first instance of router is seen in the running configuration.

Example 2-34 *Filtering Output from the show running-config Command Using begin*

```
R1#show running-config | begin router
router eigrp 100
  network 10.0.0.0
  network 192.168.1.0

router ospf 1
  log-adjacency-changes
  network 0.0.0.0 255.255.255.255 area 0
...OUTPUT OMITTED...
```

However, if the first instance of router appears in the running configuration before the router ospf section (as in Example 2-34), you will still have to sift through the running configuration until you get to the router ospf section. Because we are trying to find a specific section (in this case OSPF) in the running configuration, we can pipe the output to a section. In Example 2-35, we pipe the output of the **show running-config** command to **section router ospf** and only get output from the router ospf section. As stated earlier, when piping, you need to specify the exact case and the exact spacing. For example, **section GigabitEthernet0/1** works, but **section GigabitEthernet 0/1**, **section Gigabitethernet0/1**, and **section Gi0/1** do not work.

Example 2-35 *Filtering Output from the show running-config Command Using section*

```
R1#show running-config | section router ospf
router ospf 1
  log-adjacency-changes
  network 0.0.0.0 255.255.255.255 area 0
...OUTPUT OMITTED...
```

Another command that often generates a lengthy output, especially in larger environments, is the **show ip route** command. Consider, for example, the output of **show ip route** presented in Example 2-36.

Example 2-36 *Sample show ip route Command Output*

```
R1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

    172.16.0.0/30 is subnetted, 2 subnets
O       172.16.1.0 [110/65] via 192.168.0.22, 00:50:57, FastEthernet0/1
O       172.16.2.0 [110/65] via 192.168.0.22, 00:50:57, FastEthernet0/1
    10.0.0.0/8 is variably subnetted, 6 subnets, 3 masks
O       10.2.2.2/32 [110/2] via 192.168.0.22, 00:50:57, FastEthernet0/1
O       10.1.3.0/30 [110/129] via 192.168.0.22, 00:50:57, FastEthernet0/1
O       10.3.3.3/32 [110/66] via 192.168.0.22, 00:50:57, FastEthernet0/1
O       10.1.2.0/24 [110/75] via 192.168.0.22, 00:50:58, FastEthernet0/1
C       10.1.1.1/32 is directly connected, Loopback0
O       10.4.4.4/32 [110/66] via 192.168.0.22, 00:50:58, FastEthernet0/1
C     192.168.0.0/24 is directly connected, FastEthernet0/1
C     192.168.1.0/24 is directly connected, FastEthernet0/0
```

Although the output shown in Example 2-36 is relatively small, some IP routing tables contain hundreds or even thousands of entries. If you want to determine whether a route for network 172.16.1.0 is present in a routing table, for instance, you could issue the command **show ip route 172.16.1.0**, as depicted in Example 2-37.

Example 2-37 *Specifying a Specific Route with the show ip route Command*

```
R1#show ip route 172.16.1.0
Routing entry for 172.16.1.0/30
  Known via "ospf 1", distance 110, metric 65, type intra area
  Last update from 192.168.0.22 on FastEthernet0/1, 00:52:08 ago
  Routing Descriptor Blocks:
    * 192.168.0.22, from 10.2.2.2, 00:52:08 ago, via FastEthernet0/1
      Route metric is 65, traffic share count is 1
```

Perhaps you are looking for all subnets of the 172.16.0.0/16 address space. In that event, you could specify the subnet mask and the **longer-prefixes** argument as part of your command. Such a command, as demonstrated in Example 2-38, shows all subnets of network 172.16.0.0/16.

Example 2-38 *Filtering Output from the show ip route Command with the longer-prefixes Option*

```
R1#show ip route 172.16.0.0 255.255.0.0 longer-prefixes
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

    172.16.0.0/30 is subnetted, 2 subnets
O       172.16.1.0 [110/65] via 192.168.0.22, 00:51:39, FastEthernet0/1
O       172.16.2.0 [110/65] via 192.168.0.22, 00:51:39, FastEthernet0/1
```

Redirecting show Command Output to a File

Imagine that you are working with Cisco Technical Assistance Center (TAC) to troubleshoot an issue, and they want a file containing output from the **show tech-support** command issued on your router. Are you going to issue the command and then copy and paste it from your terminal window to a text editor? That is one option. However, Example 2-39 shows how you can use the **| redirect** option to send output from a **show** command to a file. In this case, it is the **show tech-support** command being sent to a file on a TFTP server.

Notice that directing output to a file suppresses the onscreen output, as shown in Example 2-39. If you want the **show** command to be displayed onscreen and stored to a file, you can pipe the output with the **tee** option, as demonstrated in Example 2-40.

Example 2-39 *Redirecting Output to a TFTP Server*

```
R1#show tech-support | redirect tftp://192.168.1.50/tshoot.txt
!
R1#
```

Example 2-40 *Redirecting Output While Also Displaying the Output Onscreen*

```
R1#show tech-support | tee tftp://192.168.1.50/tac.txt
!

-----show version-----

Cisco IOS Software, C2600 Software (C2600-IPVOICE_IVS-M), Version 12.4(3b), RELEASE
SOFTWARE (fc3)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2005 by Cisco Systems, Inc.
Compiled Thu 08-Dec-05 17:35 by alnguyen
...OUTPUT OMITTED...
```

In situations where you already have an output file created and you want to append the output of another **show** command to your existing file, you can pipe the output of your **show** command with the **append** option. Example 2-41 shows how to use the **append** option to add the output of the **show ip interface brief** command to a file named `baseline.txt` that was created at an earlier time and already contains information. Note that this does not overwrite the existing file; it simply adds the new information to it.

Example 2-41 *Appending Output to an Existing File*

```
R1#show ip interface brief | append tftp://192.168.1.50/baseline.txt
!
R1#
```

Troubleshooting Hardware

In addition to software configurations, a network's underlying hardware often becomes a troubleshooting target. As a reference, Table 2-4 offers a collection of Cisco IOS commands used to investigate hardware performance issues.

**Table 2-4** *Cisco IOS Commands for Hardware Troubleshooting*

Command	Description
show processes cpu	Provides 5-second, 1-minute, and 5-minute CPU utilization statistics, in addition to a listing of processes running on a platform along with each process's utilization statistics
show memory	Displays summary information about processor and I/O memory, followed by a more comprehensive report of memory utilization
show interfaces	Shows Layer 1 and Layer 2 interface status, interface load information, and error statistics including the following: input queue drops: Indicates a router received information faster than the information could be processed by the router output queue drops: Indicates a router is not able to send information out the outgoing interface because of congestion (perhaps because of an input/output speed mismatch) input errors: Indicates frames were not received correctly (for example, a cyclic redundancy check (CRC) error occurred), perhaps indicating a cabling problem or a duplex mismatch output errors: Indicates frames were not transmitted correctly, perhaps due to a duplex mismatch
Note Prior to collecting statistics, interface counters can be reset using the clear counters command.	
show controllers	Displays statistical information about an interface (for example, error statistics), where the information varies for different interface types (for example, the type of connected cable might be displayed for a serial interface and whether it is the DCE side or DTE side of the cable)
show platform	Provides detailed information about a router or switch hardware platform

Collecting Information in Transit

Information you collect while troubleshooting is not always going to be at rest. You will sometimes need to collect information while it is in transit. This section discusses how we can capture packets on the network that are flowing through our switches.

Performing Packet Captures

You can use dedicated appliances or PCs running packet capture software to collect and store packets flowing across a network link. When troubleshooting, analysis of captured

packets can provide insight into how a network is treating traffic flow. For example, a packet capture data file can show whether packets are being dropped or if sessions are being reset. You can also look inside Layer 2, 3, and 4 headers using a packet-capture application. For example, you can view a packet's Layer 3 header to determine that packet's Layer 3 quality of service (QoS) priority marking. An example of a popular and free packet-capture utility you can download is Wireshark (<http://www.wireshark.org>), as shown in Figure 2-5.

Capturing and analyzing packets, however, presents two major obstacles. First, the volume of data collected as part of a packet capture can be so large that finding what you are looking for can be a challenge. Therefore, you should understand how to use your packet capture application's filtering features.

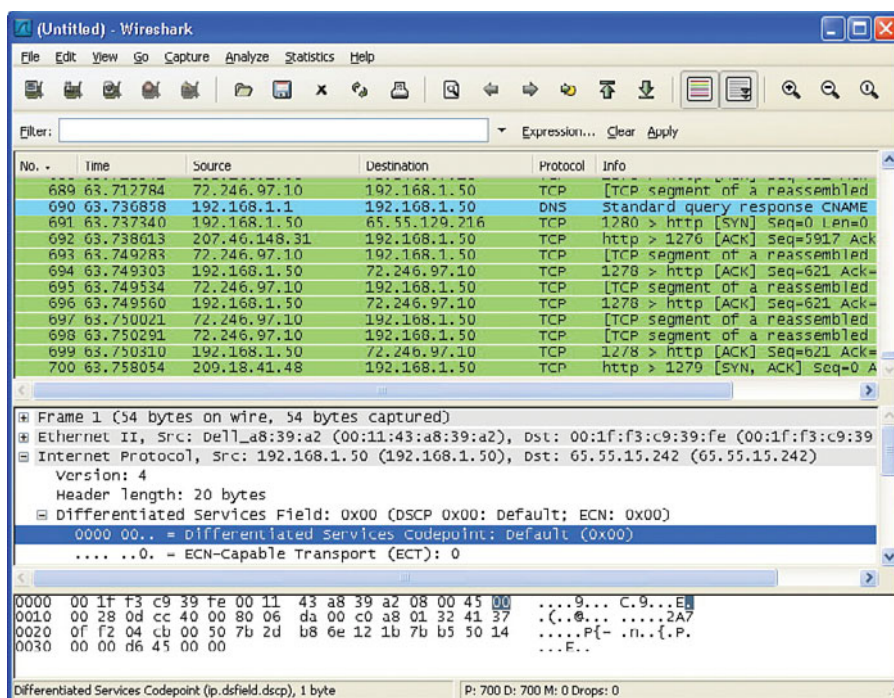


Figure 2-5 Wireshark Packet-Capture Application

SPAN

Key Topic

A second challenge occurs when you want to monitor, for example, traffic flow between two network devices connected to a switch. By default, the packets traveling between those two devices will not be seen by your packet-capturing device. This is because of how the switch is designed to behave. A switch is designed to forward frames based on the destination MAC address of a frame. When a frame is received, the switch looks in the MAC address table to determine which port the frame should be forwarded out based on the destination MAC address. Therefore, if the frame is not destined (based on the

MAC address) for the device with the packet-capturing software, the frame will not be sent out the port connected to that device. This behavior ensures that end-user devices do not see frames that are not intended for them.

Fortunately, Cisco IOS supports a feature known as Switched Port Analyzer (SPAN). SPAN instructs a switch to send copies of packets seen on one port (or one VLAN) to another port where the packet capturing device is connected, as shown in Figure 2-6.

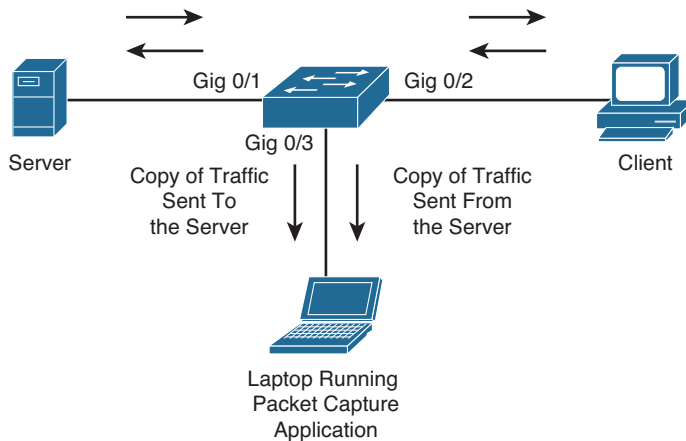


Figure 2-6 Cisco Catalyst Switch Configured for SPAN

Notice that Figure 2-6 depicts a client (connected to Gigabit Ethernet 0/2) communicating with a server (connected to Gigabit Ethernet 0/1). A troubleshooter inserts a packet capture device into Gigabit Ethernet 0/3. However, because the switch's default behavior prevents frames that are flowing between the client and server from being sent out any other port, the laptop running the packet capture application will not see any of these frames. To cause port Gigabit Ethernet 0/3 to receive a copy of all frames sent or received by the server, SPAN is configured on the switch, as shown in Example 2-42.

Notice that Example 2-42 uses the **monitor session id source interface interface_type interface_number** command to indicate that a SPAN monitoring session with a locally significant identifier of 1 will copy packets crossing (that is, entering and exiting) port Gigabit Ethernet 0/1. Then the **monitor session id destination interface interface_type interface_number** command is used to specify port Gigabit Ethernet 0/3 as the destination port for those copied packets. A laptop running packet capture software connected to port Gigabit Ethernet 0/3 will now receive a copy of all traffic the server is sending or receiving.

Example 2-42 SPAN Configuration

```
SW1#conf term
Enter configuration commands, one per line. End with CNTL/Z.
SW1(config)#monitor session 1 source interface gig 0/1
SW1(config)#monitor session 1 destination interface gig 0/3
SW1(config)#end
SW1#show monitor
```

```

Session 1
-----
Type      : Local Session
Source Ports      :
    Both   : Gi0/1
Destination Ports : Gi0/3
Encapsulation   : Native
Ingress         : Disabled

```

RSPAN



In larger environments, a network capture device connected to one switch might need to capture packets flowing through a different switch. Remote SPAN (RSPAN) makes such a scenario possible. Consider Figure 2-7, where a troubleshooter has her laptop running a packet capture application connected to port Fast Ethernet 5/2 on switch SW2. The traffic that needs to be captured is traffic coming from and going to the server connected to port Gigabit Ethernet 0/1 on switch SW1.

A VLAN is configured whose purpose is to carry captured traffic between the switches. Therefore, a trunk exists between switches SW1 and SW2 to carry the SPAN VLAN in addition to a VLAN carrying user data. Example 2-43 shows the configuration on switch SW1 used to create the RSPAN VLAN (that is, VLAN 20) and to specify that RSPAN should monitor port Gigabit Ethernet 0/1 and send packets sent and received on that port out of Gigabit Ethernet 0/3 on VLAN 20. (Note that the **reflector-port** parameter is not required on all switches [for example, a 2960].) The **show monitor** command is then used to verify the RSPAN source and destination. Also, note that by default the **monitor session id source** command monitors both incoming and outgoing traffic on the monitored port.

Example 2-43 RSPAN Configuration on Switch SW1

```

SW1#conf term
SW1(config)#vlan 20
SW1(config-vlan)#name SPAN
SW1(config-vlan)#remote-span
SW1(config-vlan)#exit
SW1(config)#monitor session 1 source interface gig 0/1
SW1(config)#monitor session 1 destination remote vlan 20 reflector-port gig 0/3
SW1(config)#end
SW1#show monitor
Session 1
-----
Type: Remote Source Session

Source Ports:
    Both: Gi0/1

Reflector Port: Gi0/3
Dest RSPAN VLAN: 20

```

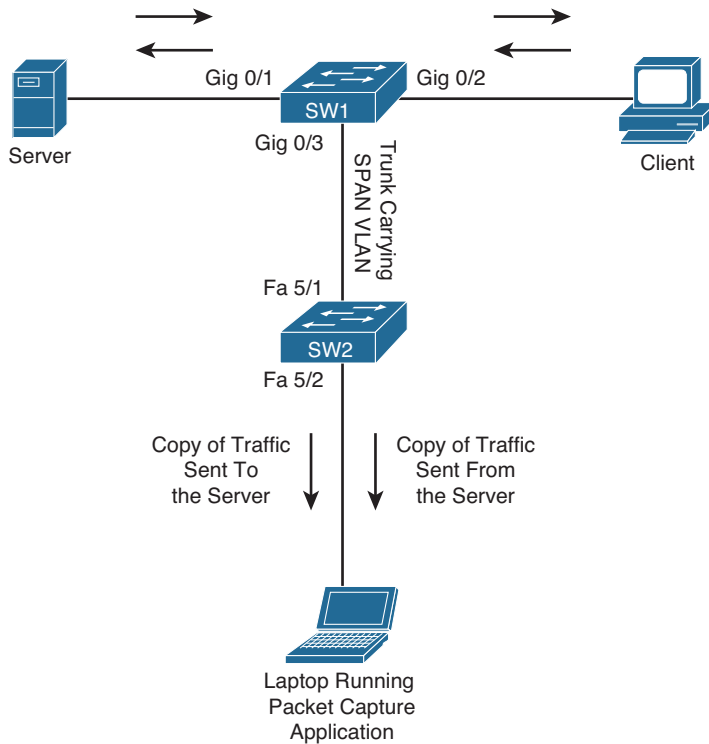


Figure 2-7 Cisco Catalyst Switch Configured for RSPAN

Example 2-44 shows the configuration on switch SW2 used to create the RSPAN VLAN to specify that RSPAN should receive captured traffic from VLAN 20 and send it out port Fast Ethernet 5/2.

Example 2-44 RSPAN Configuration on Switch SW2

```
SW2#conf term
SW2(config)#vlan 20
SW2(config-vlan)#name SPAN
SW2(config-vlan)#remote-span
SW2(config-vlan)#exit
SW2(config)#monitor session 2 source remote vlan 20
SW2(config)#monitor session 2 destination interface fa 5/2
SW2(config)#end
SW2#show monitor
Session 2
-----
Type                : Remote Destination Session
Source RSPAN VLAN   : 20
Destination Ports   : Fa5/2
```

Using Tools to Document a Network

An important undertaking for every network team is documenting the existing network. As stressed throughout this book, accurate documentation is a must. Therefore, this section covers the CLI commands that enable you to build a network diagram.



Your network currently has no network diagram. You are connected to R1 via the console port, as shown in Figure 2-8. Your first task is to find out the types of interfaces that are up/up, and the IP addresses associated with them. To accomplish this, you issue the **show ip interface brief** command, as shown in Example 2-45.

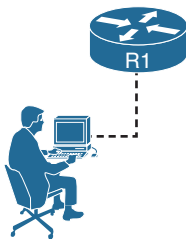


Figure 2-8 Connected to R1 via the Console Port

Example 2-45 Output of show ip interface brief Command on R1

R1#show ip interface brief						
Interface	IP-Address	OK?	Method	Status	Protocol	
FastEthernet0/0	192.168.1.1	YES	manual	up	up	
FastEthernet0/1	unassigned	YES	TFTP	administratively down	down	
Serial0/0/0	172.16.1.1	YES	manual	up	up	
Serial0/0/1	unassigned	YES	NVRAM	administratively down	down	
Serial0/2/0	unassigned	YES	NVRAM	administratively down	down	
Serial0/2/1	unassigned	YES	NVRAM	administratively down	down	

You can gather from the output in Example 2-45 that R1 has Fast Ethernet 0/0 up/up with an IP address of 192.168.1.1. It also has Serial 0/0/0 up/up with an IP address of 172.16.1.1. You can add this information to your diagram, as shown in Figure 2-9.

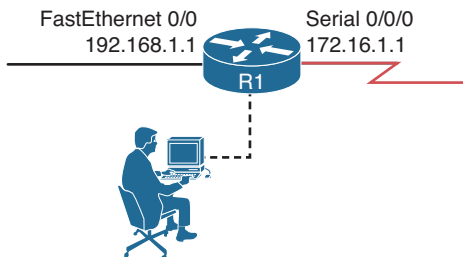


Figure 2-9 Discovered Ethernet and Serial Interfaces on R1

Next, you want to determine which Cisco devices are connected to R1. You accomplish this using the **show cdp neighbors** command, as shown in Example 2-46. You can also use the IEEE standard Link Layer Discovery Protocol (LLDP) to discover neighboring Cisco and Non-Cisco devices if you have enabled it.

Example 2-46 *Output of the show cdp neighbors Command on R1*

```
R1#show cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone,
                  D - Remote, C - CVTA, M - Two-port Mac Relay

Device ID         Local Intrfce   Holdtme    Capability  Platform  Port ID
SW1                Fas 0/0         139        S I         WS-C2960- Fas 0/24
R2                 Ser 0/0/0       133        S I         2811      Ser 0/0/0
```

You observe from the output in Example 2-46 that R1 is connected to a Catalyst 2960 switch named SW1 out Fast Ethernet 0/0. It also indicates that SW1 is using Fast Ethernet 0/24 to connect to R1. You also observe that R1 is connected to a 2811 series router named R2 out Serial 0/0/0 and that R2 is using Serial 0/0/0 to connect to R1. You add this information to the diagram, as shown in Figure 2-10.

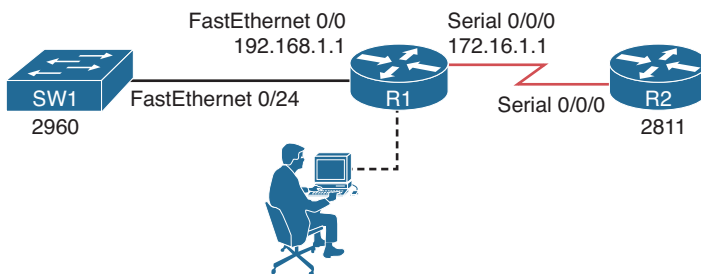


Figure 2-10 *Adding SW1 and R2 to the Diagram*

You need to discover the IP address of Serial 0/0/0 on R2 and the management IP address on SW1. To accomplish this, you use the **show cdp neighbors detail** command, as shown in Example 2-47. You observe from the output that Serial 0/0/0 on R2 has the IP address 172.16.1.2 and that the management IP address on SW1 is 192.168.1.2. You add this information to the diagram, as shown in Figure 2-11. In addition, the **show cdp neighbors detail** command will also provide the Cisco IOS Software version that is running on the neighbor.

Example 2-47 *Output of the show cdp neighbors Command on R1*

```
R1#show cdp neighbors detail
-----
Device ID: SW1
Entry address(es):
  IP address: 192.168.1.2
```

```

Platform: cisco WS-C2960-24TT-L, Capabilities: Switch IGMP
Interface: FastEthernet0/0, Port ID (outgoing port): FastEthernet0/24
Holdtime : 153 sec

Version :
Cisco IOS Software, C2960 Software (C2960-LANBASEK9-M), Version 15.0(2)SE, RELEASE
SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2012 by Cisco Systems, Inc.
Compiled Sat 28-Jul-12 00:29 by prod_rel_team

advertisement version: 2
Protocol Hello: OUI=0x00000C, Protocol ID=0x0112; payload len=27, value=00000000FFF
FFFFFFFF010220FF0000000000000081FF34EB800FF0000
VTP Management Domain: ''
Native VLAN: 1
Duplex: full
-----
Device ID: R2
Entry address(es):
  IP address: 172.16.1.2
Platform: Cisco 2811, Capabilities: Switch IGMP
Interface: Serial0/0/0, Port ID (outgoing port): Serial0/0/0
Holdtime: 127 sec

Version :
Cisco IOS Software, 2800 Software (C2800NM-ADVENTERPRISEK9-M), Version 15.1(4)M5,
RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2012 by Cisco Systems, Inc.
Compiled Tue 04-Sep-12 15:56 by prod_rel_team

advertisement version: 2
VTP Management Domain: ''

```

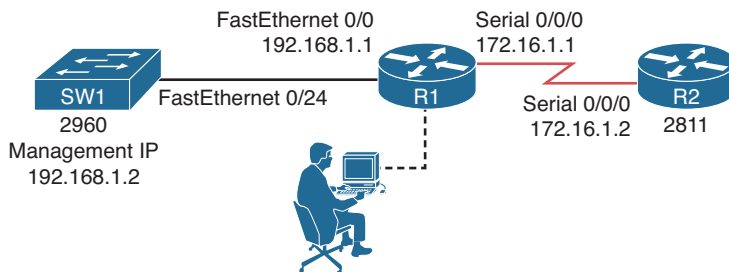


Figure 2-11 Updating IPs in Diagram for SW1 and R2

Finally, you need to include the type of router R1 is. You use the **show version** command, as shown in Example 2-48, which indicates it is also a 2811 series router. You can also verify the Cisco IOS Software version, the system bootstrap version, the number of interfaces, and the configuration register.

Example 2-48 *Output of the show version Command on R1*

```
R1#show version
Cisco IOS Software, 2800 Software (C2800NM-ADVENTERPRISEK9-M), Version 15.1(4)M5,
RELEASE SOFTWARE (fc1)

...output omitted...

ROM: System Bootstrap, Version 12.4(1r) [hqluong 1r], RELEASE SOFTWARE (fc1)

R1 uptime is 14 minutes
System returned to ROM by power-on
System image file is "flash:c2800nm-adventerprisek9-mz.151-4.M5.bin"
Last reload type: Normal Reload

...output omitted...

Cisco 2811 (revision 1.0) with 247808K/14336K bytes of memory.
Processor board ID FTX1023A49D
 2 FastEthernet interfaces
 4 Serial(sync/async) interfaces
 1 Virtual Private Network (VPN) Module
DRAM configuration is 64 bits wide with parity enabled.
239K bytes of non-volatile configuration memory.
125440K bytes of ATA CompactFlash (Read/Write)

...output omitted...

-----
Device#    PID                SN
-----
*0         CISCO2811          ...output omitted...

Configuration register is 0x2102
```

You add the type of router to your diagram as shown in Figure 2-12.

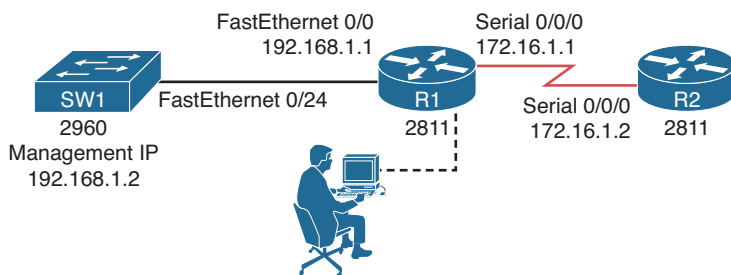


Figure 2-12 *Updating R1's Router Type in the Diagram*

As you can see, you were able to gather quite a bit of information from just four commands: **show ip interface brief**, **show cdp neighbors**, **show cdp neighbors detail**, and **show version**.

Your next step in the process of building your diagram is to connect to SW1 and R2 via their console ports or via Telnet/SSH and issue the same four commands to gather information about the devices connected to them.

Exam Preparation Tasks

As mentioned in the section “How to Use This Book” in the Introduction, you have a couple of choices for exam preparation: the exercises here; Chapter 22, “Final Preparation;” and the exam simulation questions on the CD-ROM.

Review All Key Topics

Review the most important topics in this chapter, noted with the Key Topic icon in the outer margin of the page. Table 2-5 lists a reference of these key topics and the page numbers on which each is found.

Table 2-5 *Key Topics for Chapter 2*

Key Topic Element	Description	Page Number
List	Identifies the three categories that collected information essentially falls into	45
Example 2-2	Backing up a router's startup configuration to an FTP server	49
Example 2-7	Viewing a configuration archive	50
Paragraph	Reviews how copying configurations into RAM is a merge operation	52
Paragraph	Identifies how the configure replace command is used to restore an archived configuration	53
Table 2-2	Severity levels	54
Example 2-13	Logging configuration	55
Paragraph	Identifies the importance of an NTP server and how to configure your device to use one	56
Paragraph	Discusses how you can use SNMP and NetFlow to establish baselines	58
Paragraph	Discusses how to set a device to send SNMP traps to an SNMP server	62
Paragraph	Discusses how you can use EEM to monitor and maintain a device	63
Section	Ping	64
Section	Telnet	67
Section	Traceroute	67
Table 2-4	Cisco IOS commands for hardware troubleshooting	75

Key Topic Element	Description	Page Number
Paragraph	Identifies the need for SPAN when collecting data in transit through a switch	76
Paragraph	Identifies the need for RSPAN when collecting data in transit through multiple switches	78
Paragraph	Discuss the commands and procedures needed to document a network diagram	80

Define Key Terms

Define the following key terms from this chapter and check your answers in the glossary:

CLI, wiki, GUI, TFTP, FTP, HTTP, archive, running configuration, merge, configure replace, syslog, NTP, SNMP, NetFlow, EEM, ping, Telnet, traceroute, Cisco TAC, SPAN, RSPAN, CDP

Complete Tables and Lists from Memory

Print a copy of Appendix C, “Memory Tables,” (found on the disc), or at least the section for this chapter, and complete the tables and lists from memory. Appendix D, “Memory Tables Answer Key,” also on the disc, includes completed tables and lists to check your work.

Command Reference to Check Your Memory

This section includes the most important configuration and EXEC commands covered in this chapter. It might not be necessary to memorize the complete syntax of every command, but you should be able to remember the basic keywords that are needed.

To test your memory of the commands, cover the right side of Tables 2-6 and 2-7 with a piece of paper, read the description on the left side, and then see how much of the command you can remember.

The 300-135 TSHOOT exam focuses on practical, hands-on skills that are used by a networking professional. Therefore, you should be able to identify the commands needed to configure and troubleshoot routers and switches.

Table 2-6 CLI Configuration Commands

Task	Command Syntax
Global configuration mode command, used to enter archive configuration mode	<code>archive</code>
Archive configuration mode command that specifies the IP address of an FTP server and filename prefix a router uses to write its archival configuration files	<code>path ftp://IP_address/filename_prefix</code>

Task	Command Syntax
Archive configuration mode command that causes an archival backup of a router's configuration to be written each time the router's running configuration is copied to its startup configuration	write-memory
Archive configuration mode command that specifies the interval used by a router to automatically back up its configuration	time-period <i>seconds</i>
Global configuration mode command used to specify an FTP username credential, which no longer necessitates the user entering the username	ip ftp username <i>username</i>
Global configuration mode command used to specify an FTP password credential, which no longer necessitates the user entering the password	ip ftp password <i>password</i>
Global configuration mode command used to specify an HTTP username credential, which no longer necessitates the user entering the username	ip http client username <i>username</i>
Global configuration mode command used to specify an HTTP password credential, which no longer necessitates the user entering the password	ip http client password <i>password</i>
Global configuration mode command used to log events to a router's internal buffer, optionally with a maximum number of bytes to be used by the buffer and optionally the minimum severity level of an event to be logged	logging buffered { <i>max_buffer_size</i> } { <i>minimum_severity_level</i> }
Global configuration mode command used to log events to a router's console, optionally with a minimum severity level of an event to be logged	logging console { <i>minimum_severity_level</i> }
Global configuration mode command used to specify the IP address of a syslog server to which a router's log files are written	logging ip_address
Global configuration mode command used to specify a router's local time zone and number of hours the time zone varies from Greenwich mean time (GMT)	clock timezone <i>time_zone_name</i> {+ -} <i>hours</i>

Task	Command Syntax
Global configuration mode command used to specify a router's time zone when daylight savings time is in effect, and when daylight savings time begins and ends	clock summer-time <i>time_zone_name</i> recurring {1-4} <i>beginning_day beginning_month time {1-4} ending_day ending_month time</i>
Global configuration mode command used to specify the IP address of an NTP server	ntp server <i>ip_address</i>
Global configuration mode command that configures SPAN, which specifies the source or destination interface for traffic monitoring	monitor session <i>id</i> { source destination } interface <i>interface_type interface_number</i>
VLAN configuration mode command that indicates a VLAN is to be used as an RSPAN VLAN	remote-span
Global configuration mode command that configures RSPAN on a monitored switch, where the RSPAN VLAN is specified in addition to the port identifier for the port being used to flood the monitored traffic to the monitoring switch	monitor session <i>id</i> destination remote vlan <i>VLAN_id</i> reflector- port <i>port_id</i>
Note The reflector-port parameter is not required on all switches (for example, a 2960).	
Global configuration mode command that configures RSPAN on a monitoring switch, where the RSPAN VLAN is specified	monitor session <i>id</i> source remote vlan <i>VLAN_id</i>
Global configuration mode command that defines an SNMP server read only or read/write community string	snmp-server community <i>community_string</i> { ro rw }
Global configuration mode command that specifies SNMP contact information	snmp-server contact <i>contact_info</i>
Global configuration mode command that specifies SNMP location information	snmp-server location <i>location</i>
Global configuration mode command that forces an SNMP interface index to stay consistent during data collection, even if a device is rebooted	snmp-server ifindex persist
Interface configuration mode command that enables NetFlow for that interface inbound or outbound.	ip flow ingress egress
Global configuration mode command that specifies the source interface used when communicating with an external NetFlow collector	ip flow-export source <i>interface_type interface_number</i>

Task	Command Syntax
Global configuration mode command that specifies the NetFlow version used by a device	ip flow-export version {1 5 9}
Global configuration mode command that specifies the IP address and port number of an external NetFlow collector	ip flow-export destination <i>ip_address</i> <i>port</i>
Global configuration mode command that specifies the IP address, SNMP version, and community string of an NMS	snmp-server host <i>ip_address</i> version {1 2c 3} <i>community_string</i>
Global configuration mode command that enables all possible SNMP traps	snmp-server enable traps
Global configuration mode command that creates an embedded event manager applet and enters applet configuration mode	event manager applet <i>name</i>

Table 2-7 CLI EXEC commands

Task	Command Syntax
Performs a backup of a router's startup configuration to an FTP server at the specified IP address, where the login credentials are provided by the username and password parameters	copy startup-config ftp://username:password@ip_ address
Performs a backup of a router's startup configuration to an FTP server at the specified IP address, where the login credentials have previously been added to the router's configuration	copy startup-config ftp://ip_ address
Displays files contained in a router's configuration archive	show archive
Replaces (as opposed to merges) a router's running configuration with a specified configuration archive	configure replace ftp://ip_address/ filename
Displays 5-second, 1-minute, and 5-minute CPU utilization averages, in addition to a listing of running processes with their CPU utilization	show processes cpu
Shows all subnets within the specified address space in the routing table	show ip route <i>network_address</i> <i>subnet_mask</i> longer-prefixes

Task	Command Syntax
Sends ICMP echo packets to the specified IP address, with options that include size: The number of bytes in the ICMP echo packet repeat: The number of ICMP echo packets sent timeout: The number of seconds the router waits for an ICMP echo reply packet after sending an ICMP echo packet df-bit: Sets the do not fragment bit in the ICMP echo packet	ping <i>ip_address</i> {<i>size bytes</i>} {<i>repeat number</i>} {<i>timeout seconds</i>} {<i>df-bit</i>}
Connects to a remote IP address via Telnet using TCP port 23 by default or optionally through a specified TCP port	telnet <i>ip_address</i> {<i>port</i>}
Displays summary information about processor and I/O memory, followed by a more comprehensive report of memory utilization	show memory
Shows Layer 1 and Layer 2 interface status, interface load information, and error statistics, including input queue drops: Indicates a router received information faster than the information could be processed by the router output queue drops: Indicates a router is not able to send information out the outgoing interface because of congestion (perhaps because of an input/output speed mismatch) input errors: Indicates frames were not received correctly (for example, a CRC error occurred), perhaps indicating a cabling problem or a duplex mismatch output errors: Indicates frames were not transmitted correctly, perhaps due to a duplex mismatch	show interfaces
Note Prior to collecting statistics, interface counters can be reset using the clear counters command.	

Task	Command Syntax
Displays statistical information for an interface (for example, error statistics) where the information varies for different interface types (for example, the type of connected cable might be displayed for a serial interface)	show controllers
Provides detailed information about a router or switch hardware platform	show platform



This chapter covers the following topics:

- **Troubleshooting Switch Performance Issues:** This section identifies common reasons why a switch might not be performing as expected.
- **Troubleshooting Router Performance Issues:** This section identifies common reasons why a router might not be performing as expected.

Troubleshooting Device Performance

Switches and routers consist of many different components. For example, they contain a processor, memory (volatile such as RAM and nonvolatile such as NVRAM and flash), and various interfaces. They are also responsible for performing many different tasks, such as routing, switching, and building all the necessary tables and structures needed to perform various tasks.

The building of the tables and structures is done by the CPU. The storage of these tables and structures is in some form of memory. The routers and switches forward traffic from one interface to another interface based on these tables and structures. Therefore, if a router's or switch's CPU is constantly experiencing high utilization, the memory is overloaded, or the interface buffers are full, these devices will experience performance issues.

This chapter discusses common reasons for high CPU and memory utilization on routers and switches, in addition to how we can recognize them. This chapter also covers interface statistics because they sometimes provide the initial indication of some type of issue.

“Do I Know This Already?” Quiz

The “Do I Know This Already?” quiz allows you to assess whether you should read this entire chapter thoroughly or jump to the “Exam Preparation Tasks” section. If you are in doubt about your answers to these questions or your own assessment of your knowledge of the topics, read the entire chapter. Table 3-1 lists the major headings in this chapter and their corresponding “Do I Know This Already?” quiz questions. You can find the answers in Appendix A, “Answers to the ‘Do I Know This Already?’ Quizzes.”

Table 3-1 “Do I Know This Already?” Section-to-Question Mapping

Foundation Topics Section	Questions
Troubleshooting Switch Performance Issues	1–4
Troubleshooting Router Performance Issues	5–8

Caution The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you do not know the answer to a question or are only partially sure of the answer, you should mark that question as wrong for purposes of the self-assessment. Giving yourself credit for an answer that you correctly guess skews your self-assessment results and might provide you with a false sense of security.

1. What are the components of a switch's control plane? (Choose two.)
 - a. Backplane
 - b. Memory
 - c. CPU
 - d. Forwarding logic
2. What are good indications that you have a duplex mismatch? (Choose two.)
 - a. The half-duplex side of the connection has a high number of FCS errors.
 - b. The full-duplex side of the connection has a high number of FCS errors.
 - c. The half-duplex side of the connection has a high number of late collisions.
 - d. The full-duplex side of the connection has a high number of late collisions.
3. Which of the following are situations when a switch's TCAM would punt a packet to the switch's CPU? (Choose the three best answers.)
 - a. OSPF sends a multicast routing update.
 - b. An administrator telnets to a switch.
 - c. An ACL is applied to a switchport.
 - d. A switch's TCAM has reached capacity.
4. The output of a **show processes cpu** command on a switch displays the following in the first line of the output:


```
CPU utilization for five seconds: 10%/7%; one minute: 12%; five minutes: 6%
```

 Based on the output, what percent of the switch's CPU is being consumed with interrupts?
 - a. 10 percent
 - b. 7 percent
 - c. 12 percent
 - d. 6 percent

5. Which router process is in charge of handling interface state changes?
 - a. TCP Timer process
 - b. IP Background process
 - c. Net Background process
 - d. ARP Input process
6. Which of the following is the least efficient (that is, the most CPU intensive) of a router's packet-switching modes?
 - a. Fast switching
 - b. CEF
 - c. Optimum switching
 - d. Process switching
7. What command is used to display the contents of a router's FIB?
 - a. `show ip cache`
 - b. `show processes cpu`
 - c. `show ip route`
 - d. `show ip cef`
8. Identify common reasons that a router displays a MALLOCFAIL error. (Choose the two best answers.)
 - a. Cisco IOS bug
 - b. Security issue
 - c. QoS issue
 - d. BGP filtering

Foundation Topics

Troubleshooting Switch Performance Issues

Switch performance issues can be tricky to troubleshoot because the problem reported is often subjective. For example, if a user reports that the network is running “slowly,” the user’s perception might mean that the network is slow compared to what he expects. However, network performance might very well be operating at a level that is hampering productivity and at a level that is indeed below its normal level of operation. At that point, as part of the troubleshooting process, you need to determine what network component is responsible for the poor performance. Rather than a switch or a router, the user’s client, server, or application could be the cause of the performance issue.

If you do determine that the network performance is not meeting technical expectations (as opposed to user expectations), you should isolate the source of the problem and diagnose the problem on that device. This section assumes that you have isolated the device causing the performance issue, and that device is a Cisco Catalyst switch.

Cisco Catalyst Switch Troubleshooting Targets

Cisco offers a variety of Catalyst switch platforms, with different port densities, different levels of performance, and different hardware. Therefore, troubleshooting switches will be platform dependent. Many similarities do exist, however. For example, all Cisco Catalyst switches include the following components:



- **Ports:** A switch’s ports physically connect the switch to other network devices. These ports (also known as *interfaces*) allow a switch to receive and transmit traffic.
- **Forwarding logic:** A switch contains hardware that makes forwarding decisions based on different tables in the data plane.
- **Backplane:** A switch’s backplane physically interconnects a switch’s ports. Therefore, depending on the specific switch architecture, frames flowing through a switch enter through a port (that is, the ingress port), flow across the switch’s backplane, and are forwarded out of another port (that is, an egress port).
- **Control plane:** A switch’s CPU and memory reside in the control plane. This control plane is responsible for running the switch’s operating system and building the necessary structures used to make forwarding decisions—for example, the MAC address table and the spanning-tree topology to name a few.

Figure 3-1 depicts these components within a switch. Notice that the control plane does not directly participate in the frame-forwarding process. However, the forwarding logic contained in the forwarding hardware comes from the control plane. Therefore, an indirect relationship exists between frame forwarding and the control plane. As a result, a continuous load on the control plane could, over time, impact the rate at which the switch forwards frames. Also, if the forwarding hardware is operating at maximum capac-

ity, the control plane begins to provide the forwarding logic. So, although the control plane does not architecturally appear to impact switch performance, it should be considered when troubleshooting.

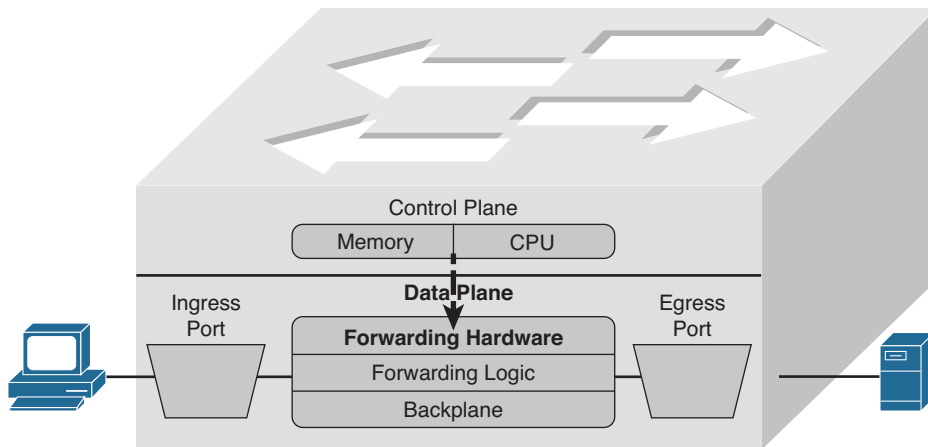


Figure 3-1 Cisco Catalyst Switch Hardware Components

The following are two common troubleshooting targets to consider when diagnosing a suspected switch issue:

- Port errors
- Mismatched duplex settings

The sections that follow evaluate these target areas in greater detail.

Port Errors

When troubleshooting a suspected Cisco Catalyst switch issue, a good first step is to check port statistics. For example, examining port statistics can let a troubleshooter know whether an excessive number of frames are being dropped. If a TCP application is running slowly, the reason might be that TCP flows are going into *TCP slow start*, which causes the window size, and therefore the bandwidth efficiency, of TCP flows to be reduced. A common reason that a TCP flow enters slow start is packet drops. Similarly, packet drops for a UDP flow used for voice or video could result in noticeable quality degradation, because dropped UDP segments are not retransmitted.

Although dropped frames are most often attributed to network congestion, another possibility is that the cabling could be bad. To check port statistics, a troubleshooter could leverage the **show interfaces** command. Consider Example 3-1, which shows the output of the **show interfaces gig 1/0/9 counters** command on a Cisco Catalyst 3750-E switch. Notice that this output shows the number of inbound and outbound frames seen on the specified port.

Example 3-1 show interfaces gig 1/0/9 counters *Command Output*

SW1#show interfaces gig 1/0/9 counters				
Port	InOctets	InUcastPkts	InMcastPkts	InBcastPkts
Gil/0/9	31265148	20003	3179	1
Port	OutOctets	OutUcastPkts	OutMcastPkts	OutBcastPkts
Gil/0/9	18744149	9126	96	6

To view errors that occurred on a port, you could add the keyword of **errors** after the **show interfaces interface_type interface_number counters** command. Example 3-2 illustrates sample output from the **show interfaces gig 1/0/9 counters errors** command.

Example 3-2 show interfaces gig 1/0/9 counters errors *Command Output*

SW1#show interfaces gig 1/0/9 counters errors							
Port	Align-Err	FCS-Err	Xmit-Err	Rcv-Err	UnderSize		
Gil/0/9	0	0	0	0	0		
Port	Single-Col	Multi-Col	Late-Col	Excess-Col	Carri-Sen	Runts	Giants
Gil/0/9	5603	0	5373	0	0	0	0

Table 3-2 provides a reference for the specific errors that might show up in the output of the **show interfaces interface_type interface_number counters errors** command.



Table 3-2 Errors in the show interfaces interface_type interface_number counters errors Command

Error Counter	Description
Align-Err	An alignment error occurs when frames do not end with an even number of octets, while simultaneously having a bad cyclic redundancy check (CRC). An alignment error normally suggests a Layer 1 issue, such as cabling or port (either switchport or network interface card [NIC] port) issues.
FCS-Err	A frame check sequence (FCS) error occurs when a frame has an invalid checksum, although the frame has no framing errors. Like the Align-Err error, an FCS-Err often points to a Layer 1 issue, but it also occurs when there is a duplex mismatch.
Xmit-Err	A transmit error (that is, Xmit-Err) occurs when a port's transmit buffer overflows. A speed mismatch between inbound and outbound links often results in a transmit error.
Rcv-Err	A receive error (that is, Rcv-Err) occurs when a port's receive buffer overflows. Congestion on a switch's backplane could cause the receive buffer on a port to fill to capacity, as frames await access to the switch's backplane. However, most likely, a Rcv-Err is indicating a duplex mismatch.
UnderSize	An undersize frame is a frame with a valid checksum but a size less than 64 bytes. This issue suggests that a connected host is sourcing invalid frame sizes.

Error Counter	Description
Single-Col	A Single-Col error occurs when a single collision occurs before a port successfully transmits a frame. Common reasons for a Single-Col error include high bandwidth utilization on an attached link or a duplex mismatch.
Multi-Col	A Multi-Col error occurs when more than one collision occurs before a port successfully transmits a frame. Similar to the Single-Col error, common reasons for a Multi-Col error include high bandwidth utilization on an attached link or a duplex mismatch.
Late-Col	A late collision is a collision that is not detected until well after the frame has begun to be forwarded. While a Late-Col error could indicate that the connected cable is too long, this is an extremely common error seen in mismatched duplex conditions.
Excess-Col	The Excess-Col error occurs when a frame experiences 16 successive collisions, after which the frame is dropped. This error could result from high bandwidth utilization, a duplex mismatch, or too many devices on a segment.
Carri-Sen	The Carri-Sen counter is incremented when a port wants to send data on a half-duplex link. This is normal and expected on a half-duplex port, because the port is checking the wire to make sure that no traffic is present prior to sending a frame. This operation is the carrier sense procedure described by the carrier sense multiple access with collision detect (CSMA/CD) operation used on half-duplex connections. Full-duplex connections, however, do not use CSMA/CD.
Runts	A runt is a frame that is less than 64 bytes in size and has a bad CRC. A runt could result from a duplex mismatch or a Layer 1 issue.
Giants	A giant is a frame size greater than 1518 bytes (assuming that the frame is not a jumbo frame) that has a bad FCS. Typically, a giant is caused by a problem with the NIC in an attached host. The jumbo frame has a frame size greater than 1518 bytes, but it has a valid FCS.

Mismatched Duplex Settings

As shown in Table 3-2, duplex mismatches can cause a wide variety of port errors. Keep in mind that almost all network devices, other than shared media hubs, can run in full-duplex mode. Therefore, if you have no hubs in your network, all devices should be running in full-duplex mode.

Cisco Catalyst switchports should be configured to autonegotiate both speed and duplex, which is the default setting. Two justifications for this recommendation are as follows:

- If a connected device supports only half-duplex, it is better for a switchport to negotiate down to half-duplex and run properly than to be forced to run full-duplex, which would result in multiple errors.

- The automatic medium-dependent interface crossover (auto-MDIX) feature can automatically detect whether a port needs a crossover or a straight-through cable to interconnect with an attached device and adjust the port to work regardless of which cable type is connected. You can enable this feature in interface configuration mode with the **mdix auto** command on some models of Cisco Catalyst switches. However, the auto-MDIX feature requires that the port autonegotiate both speed and duplex.

In a mismatched duplex configuration, a switchport at one end of a connection is configured for full-duplex, whereas a switchport at the other end of a connection is configured for half-duplex. Among the different errors previously listed in Table 3-2, two of the biggest indicators of a duplex mismatch are a high FCS-Err counter and a high Late-Col counter. Specifically, a high FCS-Err counter is common to find on the full-duplex end of a connection with a mismatched duplex, whereas a high Late-Col counter is common on the half-duplex end of the connection.

To illustrate, examine Examples 3-3 and 3-4, which display output based on the topology depicted in Figure 3-2. Example 3-3 shows the half-duplex end of a connection, and Example 3-4 shows the full-duplex end of a connection. The half-duplex end sends a frame because it thinks it is safe to send based on the CSMA/CD rule. The full-duplex end sends a frame because it is always safe to send and a collision should not occur. When the collision occurs in this example, SW1 will cease to transmit the remainder of the frame (because the port is half-duplex) and will record that a late collision occurred. However, SW2 will continue to send and receive frames. The frames it receives will not be complete because SW1 did not send the entire frame. Therefore, the FCS (mathematical checksum) of the frame does not match, and we have FCS errors on the full-duplex side.



Figure 3-2 *Topology with Duplex Mismatch*

Example 3-3 *Output from the show interfaces gig 1/0/9 counters errors and the show interfaces gig 1/0/9 | include duplex Commands on a Half-Duplex Port*

```
SW1# show interfaces gig 1/0/9 counters errors
Port          Align-Err    FCS-Err    Xmit-Err    Rcv-Err    UnderSize
Gig1/0/9      0            0          0           0          0
Port          Single-Col  Multi-Col  Late-Col    Excess-Col  Carri-Sen   Runts   Giants
Gig1/0/9      5603        0          5373        0           0           0       0
SW1#show interfaces gig 1/0/9  include duplex
Half-duplex, 100Mb/s, link type is auto, media type is 10/100/1000BaseTX
SW1#
```

Example 3-4 *Output from the show interfaces fa 5/47 counters errors and the show interfaces fa 5/47 | include duplex Commands on a Full-Duplex Port*

```
SW2#show interfaces fa 5/47 counters errors
```

Port	Align-Err	FCS-Err	Xmit-Err	Rcv-Err	UnderSize	OutDiscards
Fa5/47	0	5248	0	5603	27	0

Port	Single-Col	Multi-Col	Late-Col	Excess-Col	Carri-Sen	Runts	Giants
Fa5/47	0	0	0	0	0	227	0

Port	SQETest-Err	Deferred-Tx	IntMacTx-Err	IntMacRx-Err	Symbol-Err
Fa5/47	0	0	0	0	0

```
SW2#show interfaces fa 5/47 include duplex
Full-duplex, 100Mb/s
SW2#
```

In your troubleshooting, even if you only have access to one of the switches, if you suspect a duplex mismatch, you could change the duplex settings on the switch over which you do have control. Then, you could clear the interface counters to see whether the errors continue to increment. You could also perform the same activity (for example, performing a file transfer) that the user was performing when he noticed the performance issue. By comparing the current performance to the performance experienced by the user, you might be able to conclude that the problem has been resolved by correcting a mismatched duplex configuration.

TCAM Troubleshooting

As previously mentioned, the two primary components of forwarding hardware are forwarding logic and backplane. A switch's backplane, however, is rarely the cause of a switch performance issue, because most Cisco Catalyst switches have high-capacity backplanes. However, it is conceivable that in a modular switch chassis, the backplane will not have the throughput to support a fully populated chassis, where each card in the chassis supports the highest combination of port densities and port speeds.

The architecture of some switches allows groups of switchports to be handled by separate hardware. Therefore, you might experience a performance gain by simply moving a cable from one switchport to another. However, to strategically take advantage of this design characteristic, you must be very familiar with the architecture of the switch with which you are working.

A multilayer switch's forwarding logic can impact switch performance. A switch's forwarding logic is compiled into a special type of memory called ternary content-addressable memory (TCAM), as illustrated in Figure 3-3. TCAM works with a switch's Cisco Express Forwarding (CEF) feature in the data plane (hardware) to provide extremely fast forwarding decisions. This is accomplished because information from the control plane relating to routing processes such as unicast routing, multicast routing, and policy-based routing, as well as information related to traffic policies such as security and quality of service (QoS) access control lists (ACLs), is populated into the TCAM tables at the data plane (hardware). However, if a switch's TCAM is unable to forward traffic (for

example, the TCAM table is full and does not have the information needed to forward the traffic), that traffic is sent (punted) to the CPU so that it can be forwarded by the switch's CPU, which has a limited forwarding capability.

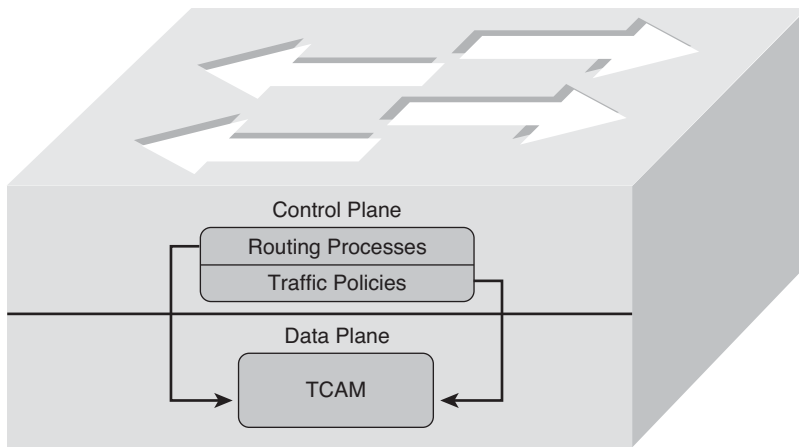


Figure 3-3 *Populating the TCAM*

The process of the TCAM sending packets to a switch's CPU is called *punting*. Consider a few reasons why a packet might be punted from a TCAM to its CPU:



- Routing protocols, in addition to other control plane protocols such as Spanning Tree Protocol (STP), that send multicast or broadcast traffic will have that traffic sent to the CPU for processing.
- Someone connecting to a switch administratively (for example, establishing a Telnet or Secure Shell [SSH] session with the switch) will have his packets sent to the CPU for processing.
- Packets using a feature not supported in hardware (for example, packets traveling over a generic routing encapsulation [GRE] tunnel) are sent to the CPU for processing.
- If a switch's TCAM has reached capacity, additional packets are punted to the CPU. A TCAM might reach capacity if it has too many installed routes or configured access control lists. This is usually the case when you attempt to use a lower-end switch in place of a higher-end switch to save money. This is not generally a good practice.

From the events listed, the event most likely to cause a switch performance issue is a TCAM filling to capacity. Therefore, when troubleshooting switch performance, you might want to investigate the state of the switch's TCAM. TCAM verification commands vary among platforms, so make sure to check the documentation for your switch model.

On most switch platforms, TCAMs cannot be upgraded. Therefore, if you conclude that a switch's TCAM is the source of the performance problems being reported, you could either use a switch with higher-capacity TCAMs or reduce the number of entries

in a switch's TCAM. For example, you could try to optimize your ACLs by being more creative with the entries or leverage route summarization to reduce the number of route entries maintained by a switch's TCAM. Also, some switches (for example, Cisco Catalyst 2960, 3560, or 3750 series switches) enable you to change the amount of TCAM memory allocated to different switch features. This allows you to “borrow” TCAM memory that was reserved for one feature and use it for another feature, optimizing the resources on the switch. This can be accomplished by changing the Switch Database Management (SDM) template on the switch. Refer to Example 3-5, which displays the TCAM resource utilization on a Catalyst 3750E switch. Notice how a finite amount of resources has been reserved for various services and features on the switch. There is a maximum value for unicast MAC addresses, IPv4 unicast and multicast routes, as well as QoS and security access control entries. It appears from this example that SW2 has maxed out the amount of resources that are reserved for IPv4 unicast indirectly connected routes. Therefore, if a packet needs to be forwarded and the needed information is not in the TCAM, it will be punted to the CPU.

Example 3-5 `show platform tcam utilization` *Command Output on a Cisco Catalyst Switch*

```
SW2#show platform tcam utilization
```

CAM Utilization for ASIC# 0	Max Masks/Values	Used Masks/values
Unicast mac addresses:	6364/6364	35/35
IPv4 IGMP groups + multicast routes:	1120/1120	1/1
IPv4 unicast directly-connected routes:	6144/6144	9/9
IPv4 unicast indirectly-connected routes:	2048/2048	2048/2048
IPv4 policy based routing aces:	442/442	12/12
IPv4 qos aces:	512/512	21/21
IPv4 security aces:	954/954	42/42

Note: Allocation of TCAM entries per feature uses a complex algorithm. The above information is meant to provide an abstract view of the current TCAM utilization

To reallocate more resources to IPv4 routing, you can change the SDM template. Using the `show sdm prefer` command on SW2, as shown in Example 3-6, indicates that the current SDM template is “desktop default,” which is the default template on a 3750E Catalyst switch. In this case, more resources need to be reserved for IPv4 routing; therefore, the template needs to be changed.

Example 3-6 `show sdm prefer` *Command Output on a Cisco Catalyst Switch*

```
SW2#show sdm prefer
```

```
The current template is "desktop default" template.
The selected template optimizes the resources in
the switch to support this level of features for
```

8 routed interfaces and 1024 VLANs.

number of unicast mac addresses:	6K
number of IPv4 IGMP groups + multicast routes:	1K
number of IPv4 unicast routes:	8K
number of directly-connected IPv4 hosts:	6K
number of indirect IPv4 routes:	2K
number of IPv4 policy based routing aces:	0
number of IPv4/MAC qos aces:	0.5K
number of IPv4/MAC security aces:	0.875k

Using the global configuration command **sdm prefer**, as shown in Example 3-7, allows you to change the SDM template. In this case, the SDM template is being changed to **routing** so that more resources will be used for IPv4 unicast routing.

Example 3-7 Changing the SDM Template on a Cisco 3750E Catalyst Switch

```
SW2#confi g t
Enter configuration commands, one per line. End with CNTL/Z.
SW2(config)#sdm prefer ?
    access                Access bias
    default               Default bias
    dual-ipv4-and-ipv6    Support both IPv4 and IPv6
    indirect-ipv4-and-ipv6-routing Supports more V4 and V6 Indirect Routes
    lanbase-routing       Supports both IPv4 and IPv6 Static Routing
    routing               Unicast bias
    vlan                  VLAN bias
SW2(config)#sdm prefer routing
Changes to the running SDM preferences have been stored, but cannot take effect
until the next reload.
Use 'show sdm prefer' to see what SDM preference is currently active.
SW2(config)#exit
SW2#reload
System configuration has been modified. Save? [yes/no]: yes
Building configuration...
[OK]
Proceed with reload? [confirm]

%SYS-5-RELOAD: Reload requested by console. Reload Reason: Reload command.
```

After the reload, notice how the SDM template is listed as “desktop routing” in Example 3-8 and that more resources are now dedicated to IPv4 indirect routes. However, also notice that while more resources are allocated to IPv4 unicast routes, fewer resources are allocated to other resources, such as unicast MAC addresses.

Example 3-8 *Verifying That the SDM Template Was Changed After Reload*

```
SW2#show sdm prefer
```

The current template is "desktop routing" template.

The selected template optimizes the resources in the switch to support this level of features for 8 routed interfaces and 1024 VLANs.

```

number of unicast mac addresses:          3K
number of IPv4 IGMP groups + multicast routes:  1K
number of IPv4 unicast routes:             11K
    number of directly-connected IPv4 hosts:    3K
    number of indirect IPv4 routes:           8K
number of IPv4 policy based routing aces:    0.5K
number of IPv4/MAC qos aces:               0.5K
number of IPv4/MAC security aces:          1K

```

In Example 3-9, the output of **show platform tcam utilization** shows that the max masks/values are now 8144/8144 for IPv4 unicast indirectly connected routes; before, they were 2048. In addition, the used masks/values are now 3148, and therefore, the TCAM can forward traffic without having to punt the packets to the CPU.

Example 3-9 *Verifying the tcam utilization on the 3750E Catalyst Switch*

```
SW2#show platform tcam utilization
```

CAM Utilization for ASIC# 0	Max Masks/Values	Used Masks/values
Unicast mac addresses:	3292/3292	35/35
IPv4 IGMP groups + multicast routes:	1120/1120	1/1
IPv4 unicast directly-connected routes:	3072/3072	8/8
IPv4 unicast indirectly-connected routes:	8144/8144	3148/3148
IPv4 policy based routing aces:	490/490	13/13
IPv4 qos aces:	474/474	21/21
IPv4 security aces:	964/964	42/42

Note: Allocation of TCAM entries per feature uses a complex algorithm. The above information is meant to provide an abstract view of the current TCAM utilization

High CPU Utilization Troubleshooting on a Switch

The load on a switch's CPU is often low, even under high utilization, thanks to the TCAM. Because the TCAM maintains a switch's forwarding logic at the data plane, the CPU is rarely tasked to forward traffic. The **show processes cpu** command can be used on a Cisco Catalyst switch to display CPU utilization levels, as demonstrated in Example 3-10.

Example 3-10 *show processes cpu Command Output on a Cisco Catalyst Switch*

```
SW1#show processes cpu
CPU utilization for five seconds: 19%/15%; one minute: 20%; five minutes: 13%
  PID  Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min  TTY Process
    1         0         4          0   0.00%   0.00%   0.00%  0 Chunk Manager
    2         0        610          0   0.00%   0.00%   0.00%  0 Load Meter
    3       128         5      25600   0.00%   0.00%   0.00%  0 crypto sw pk pro
    4      2100        315       6666   0.00%   0.05%   0.05%  0 Check heaps
...OUTPUT OMITTED...
```

Notice in the output in Example 3-10 that the switch is reporting a 19 percent CPU load, with 15 percent of the CPU load used for interrupt processing.

Although such load utilization values might not be unusual for a router, these values might be of concern for a switch. Specifically, a typical CPU load percentage dedicated to interrupt processing is no more than 5 percent. A value as high as 10 percent is considered acceptable. However, the output given in Example 3-10 shows a 15 percent utilization, which is considered high for a Catalyst switch. Such a level implies that the switch's CPU is actively involved in forwarding packets that should normally be handled by the switch's TCAM. Of course, this value might be normal for your organization based on baseline information, even though according to Cisco it is a cause for concern. If the interrupt percent is greater than 10, take time to look into the reason why.

Periodic spikes in processor utilization are also not a major cause for concern if such spikes can be explained. Consider the following reasons that might cause a switch's CPU utilization to spike:

- The CPU is processing routing updates.
- The administrator is issuing a **debug** command (or other processor-intensive commands).
- Simple Network Management Protocol (SNMP) is being used to poll network devices.

If you determine that a switch's high CPU load is primarily the result of interrupts, examine the switch's packet-switching patterns and check the TCAM utilization. If the high CPU utilization is primarily the result of processes, take the time to investigate those specific processes.

A high CPU utilization on a switch might be a result of STP. Recall that an STP failure could lead to a broadcast storm, where Layer 2 broadcast frames endlessly circulate through a network. Therefore, when troubleshooting a performance issue, realize that a switch's high CPU utilization might be a symptom of another issue.

Troubleshooting Router Performance Issues

As you have seen, a Cisco Catalyst switch's performance can be the source of network problems. Similarly, a router performance issue can impact user data flowing through the network.

As an administrator, you might notice a sluggish response to Telnet sessions or SSH sessions that you attempt to establish with a router. Or, you might experience longer-than-normal ping response times from a router. Such symptoms might indicate a router performance issue. In these examples, the router's CPU is so busy it does not have time to respond to your Telnet session or the pings you have sent.

This section investigates three potential router issues, each of which might result in poor router performance

- Excessive CPU utilization
- The packet-switching mode of a router
- Excessive memory utilization

Excessive CPU Utilization

A router's processor (that is, CPU) utilization escalating to a high level but only remaining at that high level for a brief time could represent normal behavior. However, if a router's CPU utilization continually remains at a high level, network performance issues might result. Aside from latency that users and administrators can experience, a router whose CPU is overtaxed might not send routing protocol messages to neighboring routers in a timely fashion. As a result, routing protocol adjacencies can fail, resulting in some networks becoming unreachable.

Processes That Commonly Cause Excessive CPU Utilization

One reason that the CPU of a router might be overloaded is that the router is running a process that is taking up an unusually high percentage of its CPU resources. Following are four such processes that can result in excessive CPU utilization:



- **ARP Input process:** The ARP Input process is in charge of sending Address Resolution Protocol (ARP) requests. This process can consume an inordinate percentage of CPU resources if the router has to send numerous ARP requests. One configuration that can cause such a high number of ARP requests is having a default route configured that points to an Ethernet interface. For example, perhaps a router had the **ip route 0.0.0.0 0.0.0.0 fastethernet 0/1** command entered in global configuration mode so that all packets with no explicit route in the routing table will be forwarded out Fa0/1. At first, this appears harmless; however, such a configuration should be avoided because an ARP Request has to be sent for every destination IP address in every packet that is received by the router and forwarded out Fa0/1. This is because the **ip route** command is stating that all IP addresses (0.0.0.0 0.0.0.0) are reachable through the directly connected interface fastethernet 0/1. Therefore, instead of ARPing for the MAC address of a next-hop IP address, you ARP for the MAC address of the destination IP address in each packet. That will result in an excessive number of ARP requests, which will cause strain on the CPU. In addition, many of the ARP requests will go unanswered and result in dropped packets. The better option is to specify the next-hop IP address because the router will only have

to ARP for the MAC of the next-hop IP address when forwarding the packets out Fa0/1.

- **Net Background process:** An interface has a certain number of buffers available to store packets. These buffers are sometimes referred to as the *queue* of an interface. If an interface needs to store a packet in a buffer but all interface buffers are in use, the interface can pull from a main pool of buffers that the router maintains. The process that allows an interface to allocate one of these globally available buffers is Net Background. If the *throttles*, *ignored*, and *overrun* parameters are incrementing on an interface, the underlying cause might be the Net Background process consuming too many CPU resources.
- **IP Background process:** The IP Background process handles an interface changing its state. A state change might be an interface going from an Up state to a Down state, or vice versa. Another example of state change is an interface's IP address changing. Therefore, anything that can cause repeated state changes, such as bad cabling, might result in the IP Background process consuming a high percentage of CPU resources.
- **TCP Timer process:** The TCP Timer process runs for each TCP router connection. Therefore, many connections can result in high CPU utilization by the TCP Timer process, whether they are established or embryonic. An established TCP connection is one that has successfully completed the three-way handshake. An embryonic connection occurs when the TCP three-way handshake is only two-thirds completed. For example, the client sends the SYN packet to the server, and then the server sends a SYN/ACK back. At this point, the server is in the embryonic state (waiting for an ACK from the client to complete the three-way handshake and establish the connection). However, if the client does not send the ACK back, the server will sit in the embryonic state until it times out. This could be due to connectivity issues or malicious intent.

Cisco IOS Commands Used for Troubleshooting High Processor Utilization

Table 3-3 offers a collection of **show** commands that can be valuable when troubleshooting high CPU utilization on a router.



Table 3-3 *Commands for Troubleshooting High CPU Utilization*

Command	Description
show ip arp	Displays the ARP cache for a router. If several entries are in the Incomplete state, you might suspect a malicious scan (for example, a ping sweep) of a subnet, or you have a route pointing out an Ethernet interface as described in our ARP Input process discussion.

Command	Description
<code>show interface interface_type interface_number</code>	Displays a collection of interface statistics. If the throttles, overruns, or ignored counters continually increment, you might suspect that the Net Background process is attempting to allocate buffer space for an interface from the main buffer pool of the router.
<code>show tcp statistics</code>	Provides information about the number of TCP segments a router sends and receives, including the number of connections initiated, accepted, established, and closed. A high number of connections can explain why the TCP Timer process might be consuming excessive CPU resources. If you see an excessive number of embryonic connections, you might be under a denial-of-service (DoS) attack.
<code>show processes cpu</code>	Displays average CPU utilization over 5-second, 1-minute, and 5-minute intervals, in addition to listing all the router processes and the percentage of CPU resources consumed by each of those processes.
<code>show processes cpu history</code>	Displays a graphical view of CPU utilization over the past 60 seconds, 1 hour, and 3 days. This graphical view can indicate whether an observed high CPU utilization is a temporary spike in utilization or whether the high CPU utilization is an ongoing condition.

Example 3-11 shows sample output from the `show ip arp` command. In the output, only a single instance exists of an Incomplete ARP entry. However, a high number of such entries can suggest the scanning of network resources, which might indicate malicious reconnaissance traffic or that you have a route pointing out an Ethernet interface instead of to a next-hop IP address.

Example 3-11 `show ip arp` Command Output

```
R2#show ip arp
Protocol Address      Age (min)  Hardware Addr  Type   Interface
Internet 10.3.3.2          61  0009.b7fa.d1e0  ARPA   Ethernet0/0
Internet 10.3.3.1           -  00d0.06fe.9ea0  ARPA   Ethernet0/0
Internet 192.168.1.50       0  Incomplete     ARPA
```

Example 3-12 shows sample output from the `show interface interface_type interface_number` command. Note the throttles, overrun, and ignored counters. If these counters continue to increment, the Net Background process might be consuming excessive CPU resources while it allocates buffers from the main buffer pool of the router.

Example 3-12 *show interface interface_type interface_number Command Output*

```

R2#show interface ethernet 0/0
Ethernet0/0 is up, line protocol is up
  Hardware is AmdP2, address is 00d0.06fe.9ea0 (bia 00d0.06fe.9ea0)
  Internet address is 10.3.3.1/24
  MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:02, output 00:00:02, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 1 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
    2156 packets input, 164787 bytes, 0 no buffer
    Received 861 broadcasts, 0 runts, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
    0 input packets with dribble condition detected
    2155 packets output, 212080 bytes, 0 underruns
    0 output errors, 0 collisions, 7 interface resets
    0 babbles, 0 late collision, 0 deferred
    0 lost carrier, 0 no carrier
    0 output buffer failures, 0 output buffers swapped out

```

Example 3-13 shows sample output from the **show tcp statistics** command. If the output indicates numerous connections, the TCP Timer process might be consuming excessive CPU resources while simultaneously maintaining all those connections. If you have a high number of initiated connections with a low number of established connections, it indicates that the three-way handshake is not being completed. This might be due to a DoS attack that is attempting to consume all the TCP connection slots.

Example 3-13 *show tcp statistics Command Output*

```

R2#show tcp statistics
Rcvd: 689 Total, 0 no port
  0 checksum error, 0 bad offset, 0 too short
  474 packets (681 bytes) in sequence
  0 dup packets (0 bytes)
  0 partially dup packets (0 bytes)
  0 out-of-order packets (0 bytes)
  0 packets (0 bytes) with data after window
  0 packets after close
  0 window probe packets, 0 window update packets

```

```

1 dup ack packets, 0 ack packets with unsend data
479 ack packets (14205 bytes)
Sent: 570 Total, 0 urgent packets
1 control packets (including 0 retransmitted)
562 data packets (14206 bytes)
0 data packets (0 bytes) retransmitted
0 data packets (0 bytes) fastretransmitted
7 ack only packets (7 delayed)
0 window probe packets, 0 window update packets
0 Connections initiated, 1 connections accepted, 1 connections established
0 Connections closed (including 0 dropped, 0 embryonic dropped)
0 Total rxmt timeout, 0 connections dropped in rxmt timeout
0 Keepalive timeout, 0 keepalive probe, 0 Connections dropped in keepalive

```

Example 3-14 shows sample output from the **show processes cpu** command. The output in this example indicates a 34 percent CPU utilization in the past 5 seconds, with 13 percent of CPU resources being spent on interrupts. The output also shows the 1-minute CPU utilization average as 36 percent and the 5-minute average as 32 percent. Individual processes running on the router are also shown, along with their CPU utilization levels. Note the ARP Input, Net Background, TCP Timer, and IP Background processes referred to in this section.

Example 3-14 show processes cpu Command Output

```

R2#show processes cpu
CPU utilization for five seconds: 34%/13%; one minute: 36%; five minutes: 32%
PID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min   TTY Process
...OUTPUT OMITTED...
12          4        69         57  0.00%  0.00%  0.00%  0 ARP Input
13          0         1          0  0.00%  0.00%  0.00%  0 HC Counter Timer
14          0         5          0  0.00%  0.00%  0.00%  0 DDR Timers
15         12         2       6000  0.00%  0.00%  0.00%  0 Entity MIB API
16          4         2       2000  0.00%  0.00%  0.00%  0 ATM Idle Timer
17          0         1          0  0.00%  0.00%  0.00%  0 SERIAL A'detect
18          0      3892          0  0.00%  0.00%  0.00%  0 GraphIt
19          0         2          0  0.00%  0.00%  0.00%  0 Dialer event
20          0         1          0  0.00%  0.00%  0.00%  0 Critical Bkgnd
21         132      418       315  0.00%  0.00%  0.00%  0 Net Background
22          0        15          0  0.00%  0.00%  0.00%  0 Logger
...OUTPUT OMITTED...
46          0      521          0  0.00%  0.00%  0.00%  0 SSS Test Client
47         84      711       118  0.00%  0.00%  0.00%  0 TCP Timer
48          4         3     1333  0.00%  0.00%  0.00%  0 TCP Protocols
49          0         1          0  0.00%  0.00%  0.00%  0 Socket Timers
50          0        15          0  0.00%  0.00%  0.00%  0 HTTP CORE
51         12         5     2400  0.00%  0.00%  0.00%  0 PPP IP Route
52          4         5       800  0.00%  0.00%  0.00%  0 PPP IPCP

```

```

53      273      157      1738  0.00%  0.00%  0.00%  0 IP Background
54          0      74          0  0.00%  0.00%  0.00%  0 IP RIB Update
...OUTPUT OMITTED...

```

Example 3-15 shows sample output from the **show processes cpu history** command. The graphical output produced by this command is useful in determining whether a CPU spike is temporary or whether it is an ongoing condition.

Example 3-15 show processes cpu history Command Output

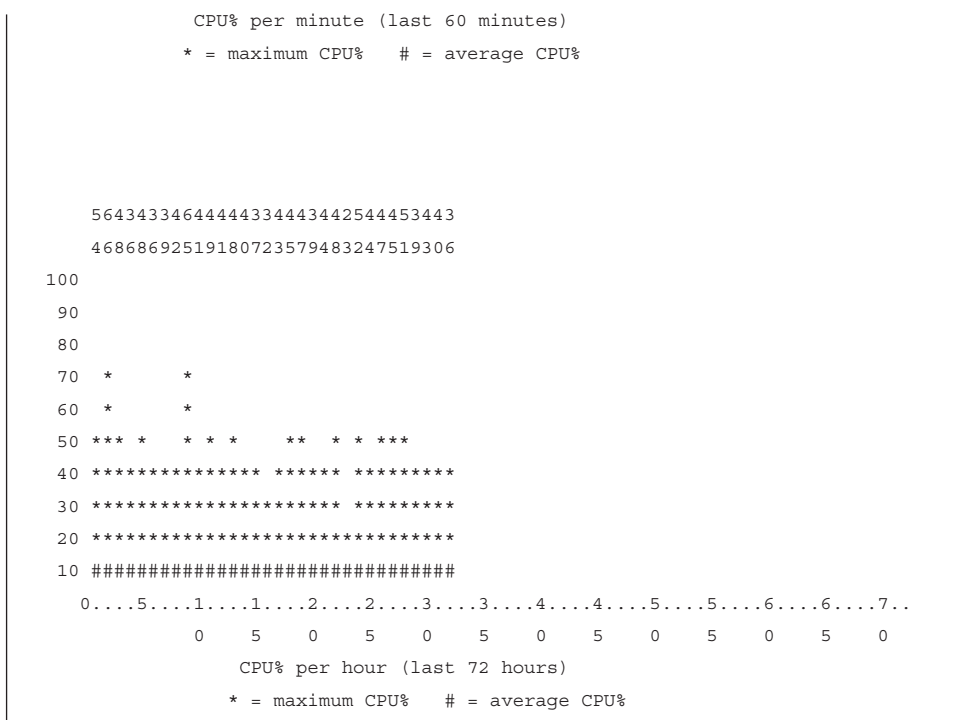
```

R2#show processes cpu history

      4              11111      4444411111      11111
9444444555544444444447777755555888888888877775555577775555
100
 90
 80
 70
 60
 50 *              *****
 40 *              *****
 30 *              *****
 20 *              *****
 10 * *****
    0...5...1...1...2...2...3...3...4...4...5...5...6
      0  5  0  5  0  5  0  5  0  5  0
      CPU% per second (last 60 seconds)

      611111111121112211311111111111121111111111121111111111
376577846281637117756665771573767217674374737664008927775277
100
 90
 80
 70
 60 *
 50 *
 40 *              *
 30 *              *
 20 ***** * * * * ***** * * * * * * * * * * * * * * * *
 10 #####
    0...5...1...1...2...2...3...3...4...4...5...5...6
      0  5  0  5  0  5  0  5  0  5  0

```



Understanding Packet-Switching Modes (Routers and Multilayer Switches)

In addition to the high CPU utilization issues previously discussed, a router's packet-switching mode can impact router performance. Before discussing the most common switching modes, realize that the way a router handles packets (or is capable of handling packets) largely depends on the router's architecture. Therefore, for real-world troubleshooting, consult the documentation for your router to determine how it implements packet switching.

In general, however, Cisco routers and multilayer switches support the following three primary modes of packet switching:



- Process switching
- Fast switching (route caching)
- Cisco Express Forwarding (topology-based switching)

Packet switching involves the router making a decision about how a packet should be forwarded and then forwarding that packet out of the appropriate router interface.

Operation of Process Switching

When a router routes a packet (that is, performs packet switching), the router removes the packet's Layer 2 header, examines the Layer 3 addressing, and decides how to forward

the packet. The Layer 2 header is then rewritten (which involves changing the source and destination MAC addresses and computing a new FCS), and then the packet is forwarded out of the appropriate interface. With process switching, as illustrated in Figure 3-4, the router's CPU becomes directly involved with packet-switching decisions. As a result, the performance of a router configured for process switching can suffer significantly.

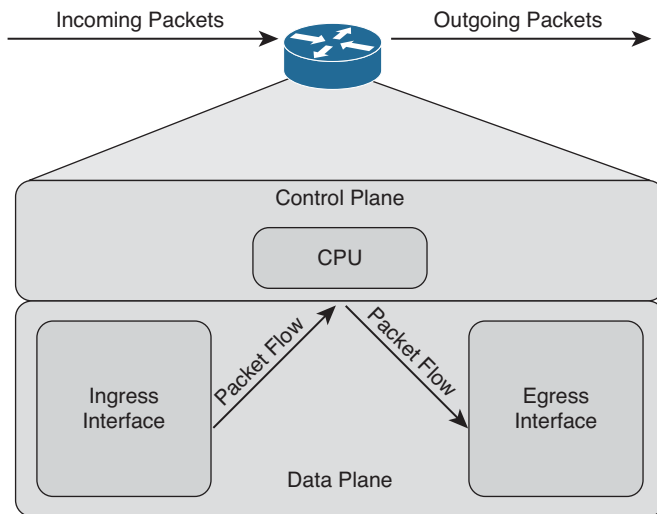


Figure 3-4 *Data Flow with Process Switching*

An interface can be configured for process switching by disabling fast switching and CEF on that interface. The interface configuration mode command used to disable fast switching and CEF at the same time is **no ip route-cache**.

Operation of Fast Switching (Route Caching)

Fast switching uses a fast cache maintained in a router's data plane. The fast cache contains information about how traffic from different data flows should be forwarded. As shown in Figure 3-5, the first packet in a data flow is process-switched by a router's CPU. After the router determines how to forward the first packet of a data flow, that forwarding information is stored in the fast cache. Subsequent packets in that same data flow are forwarded based on information in the fast cache, as opposed to being process-switched. As a result, fast switching reduces a router's CPU utilization when compared to process switching.

You can enable fast switching by turning off CEF in interface configuration mode with the **no ip route-cache cef** command.

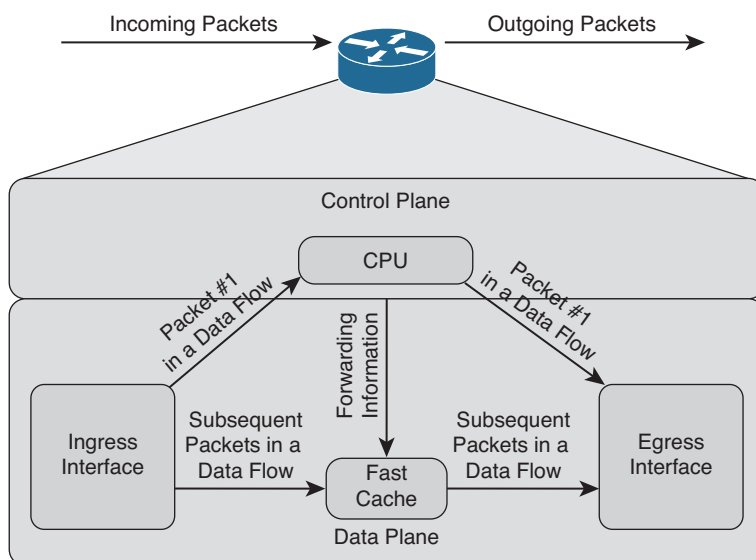


Figure 3-5 *Data Flow with Fast Switching*

Operation of Cisco Express Forwarding (Topology-Based Switching)

Cisco Express Forwarding (CEF) maintains two tables in the data plane. Specifically, the Forwarding Information Base (FIB) maintains Layer 3 forwarding information, whereas the Adjacency Table maintains Layer 2 information for next hops listed in the FIB.

Using these tables, populated from a router's IP routing table and ARP cache, CEF can efficiently make forwarding decisions. Unlike fast switching, CEF does not require the first packet of a data flow to be process-switched. Rather, an entire data flow can be forwarded at the data plane, as shown in Figure 3-6.

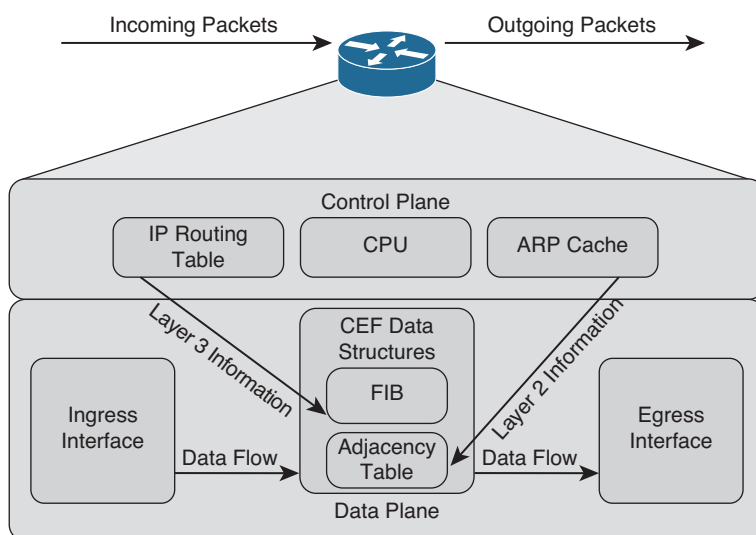


Figure 3-6 *Data Flow with Cisco Express Forwarding*

On many router platforms, CEF is enabled by default. If it is not, you can globally enable it with the **ip cef** command. Alternatively, you can enable CEF for a specific interface with the interface configuration mode command **ip route-cache cef**.

Date Night Example of Process-Switching Modes

Let's pretend that my wife and I are going out to dinner and we are leaving our two children with a babysitter. If we are "Process Switching" with the babysitter, every time our children ask the babysitter for a cookie, she has to call us to ask for permission to give the children a cookie. If the children ask ten times, she has to call us ten times. If we are "Fast Switching" with the babysitter, the first time she calls us, we say yes and then create a "route cache" for the babysitter that states, "if the kids want more, just give them more without calling us." Finally, if we are using "CEF" with the babysitter, before we leave for dinner, we take out the cookie jar, place it on the counter, and tell her to have an awesome evening with the kids. As you can see from this example, date night is better when we use CEF.

Troubleshooting Packet-Switching Modes

Table 3-4 provides a selection of commands that you can use when troubleshooting the packet-switching modes of a router.



Table 3-4 *Commands for Troubleshooting a Router's Packet-Switching Modes*

Command	Description
show ip interface <i>interface_type interface_number</i>	Displays multiple interface statistics, including information about the packet-switching mode of an interface.
show ip cache	Displays the contents of the route cache from a router if fast switching is enabled.
show processes cpu include IP Input	Displays information about the IP input process on a router. The CPU utilization for this process might show a high value if the CPU of a router is actively engaged in process-switching traffic because you turned off fast switching and CEF.
show ip cef	Displays the contents of a router's FIB.
show ip cef adjacency <i>egress_interface_id next_hop_ip_address detail</i>	Displays destinations reachable through the combination of the specified egress interface and next-hop IP address.
show adjacency detail	Provides information contained in the adjacency table of a router, including protocol and timer information.

Example 3-16 shows sample output from the **show ip interface *interface_type interface_number*** command. The output indicates that fast switching and CEF switching are enabled on interface Fast Ethernet 0/0. The reference to flow switching being disabled refers to the Cisco IOS NetFlow feature, which you can use to collect traffic statistics.

Example 3-16 **show ip interface *interface_type interface_number* Command Output**

```
R4#show ip interface fastethernet 0/0
FastEthernet0/0 is up, line protocol is up
...OUTPUT OMITTED...
  ICMP mask replies are never sent
  IP fast switching is enabled
  IP fast switching on the same interface is disabled
  IP Flow switching is disabled
  IP CEF switching is enabled
  IP CEF Fast switching turbo vector
  IP multicast fast switching is enabled
  IP multicast distributed fast switching is disabled
  IP route-cache flags are Fast, CEF
...OUTPUT OMITTED...
```

Example 3-17 shows sample output from the **show ip cache** command. If fast switching is enabled and CEF is disabled, a router begins to populate its route cache. This command shows the contents of a router's route cache.

Example 3-17 **show ip cache Command Output**

```
R4#show ip cache
IP routing cache 3 entries, 588 bytes
  12 adds, 9 invalidates, 0 refcounts
Minimum invalidation interval 2 seconds, maximum interval 5 seconds,
  quiet interval 3 seconds, threshold 0 requests
Invalidation rate 0 in last second, 0 in last 3 seconds
Last full cache invalidation occurred 04:13:57 ago

Prefix/Length      Age           Interface     Next Hop
10.8.8.4/32        00:00:07     FastEthernet0/1 10.8.8.4
10.8.8.6/32        00:00:10     FastEthernet0/1 10.8.8.6
192.168.0.0/24     00:00:10     FastEthernet0/0 10.3.3.1
```

Example 3-18 shows sample output from the **show processes cpu | include IP Input** command. In the output, the IP input process was using only 0.08 percent of its router's CPU capacity during the last 5-second interval. However, a high percentage value might indicate that a router was performing process switching, where the CPU was directly involved in packet switching.

Example 3-18 show processes cpu | include IP Input *Command Output*

```
R4#show processes cpu | include IP Input
63          3178      7320          434  0.08%  0.06%  0.04%  0 IP Input
```

Example 3-19 shows sample output from the **show ip cef** command. The output contains the contents of the FIB for a router. Notice that the prefix is listed, followed by the next hop that will be used to reach the prefix, and then the interface that will be used to reach it. Note that if a next hop of the network prefix is set to *receive*, that network/IP is local to the router, and any packets destined to that specific IP will be processed by the CPU of the router. Examining the output closely, you will see that the receive entries are subnet IDs, local host IP addresses, and broadcast addresses, ensuring that they are processed by the router and not forwarded. The *attached* next hop indicates that the network is a directly connected route on the router.

Example 3-19 show ip cef *Command Output*

```
R4#show ip cef
Prefix      Next Hop      Interface
0.0.0.0/0   drop          Null0 (default route handler entry)
0.0.0.0/32   receive
10.1.1.0/24  10.3.3.1      FastEthernet0/0
10.1.1.2/32  10.3.3.1      FastEthernet0/0
10.3.3.0/24  attached      FastEthernet0/0
10.3.3.0/32  receive
10.3.3.1/32  10.3.3.1      FastEthernet0/0
10.3.3.2/32  receive
10.3.3.255/32 receive
10.4.4.0/24  10.3.3.1      FastEthernet0/0
10.5.5.0/24  10.3.3.1      FastEthernet0/0
10.7.7.0/24  10.3.3.1      FastEthernet0/0
10.7.7.2/32  10.3.3.1      FastEthernet0/0
10.8.8.0/24  attached      FastEthernet0/1
10.8.8.0/32  receive
10.8.8.1/32  receive
10.8.8.4/32  10.8.8.4      FastEthernet0/1
10.8.8.5/32  10.8.8.5      FastEthernet0/1
10.8.8.6/32  10.8.8.6      FastEthernet0/1
10.8.8.7/32  10.8.8.7      FastEthernet0/1
10.8.8.255/32 receive
192.168.0.0/24 10.3.3.1      FastEthernet0/0
224.0.0.0/4   drop
224.0.0.0/24  receive
255.255.255.255/32 receive
```

Example 3-20 shows sample output from the **show ip cef adjacency egress_interface_id next_hop_ip_address detail** command. This command shows the IP addresses that the router knows how to reach using the specified combination of next-hop IP address and egress interface. In this example, 10.8.8.6 is the IP address of a host and not a router. Therefore, no other IP addresses are known to have a next-hop IP address of 10.8.8.6 with an egress interface of Fast Ethernet 0/1.

Example 3-20 **show ip cef adjacency egress-interface-id next-hop-IP-address detail**
Command Output

```
R4#show ip cef adjacency fa 0/1 10.8.8.6 detail
IP CEF with switching (Table Version 25), flags=0x0
  25 routes, 0 reresolve, 0 unresolved (0 old, 0 new), peak 0
  25 leaves, 21 nodes, 25640 bytes, 90 inserts, 65 invalidations
  0 load sharing elements, 0 bytes, 0 references
  universal per-destination load sharing algorithm, id 24360DB1
  5(2) CEF resets, 1 revisions of existing leaves
  Resolution Timer: Exponential (currently 1s, peak 1s)
  0 in-place/0 aborted modifications
  refcounts: 5702 leaf, 5632 node

  Table epoch: 0 (25 entries at this epoch)

Adjacency Table has 5 adjacencies
10.8.8.6/32, version 10, epoch 0, cached adjacency 10.8.8.6
0 packets, 0 bytes
  via 10.8.8.6, FastEthernet0/1, 0 dependencies
    next hop 10.8.8.6, FastEthernet0/1
    valid cached adjacency
```

Example 3-21 shows sample output from the **show adjacency detail** command. When you see a particular adjacency listed in the FIB, you can issue this command to confirm that the router has information about how to reach that adjacency. In this case, if we need to send a packet to 10.3.3.1, we will send the packet out Fast Ethernet 0/0, which requires a Layer 2 frame with a source and destination MAC address. These MAC addresses are already listed in the adjacency table. The value 00D006FE9EA00009B7FAD1E00800 can be broken into three parts:

- 00D006FE9EA0 = Destination MAC address
- 0009B7FAD1E0 = Source MAC address
- 0800 = Well-known Ethertype value for IP

Example 3-21 show adjacency detail *Command Output*

```

R4#show adjacency detail
Protocol Interface      Address
IP        FastEthernet0/0      10.3.3.1 (19)
          32 packets, 1920 bytes
          00D006FE9EA00009B7FAD1E00800
          ARP           03:53:01
          Epoch: 0
IP        FastEthernet0/1 10.8.8.6 (5)
          4 packets, 264 bytes
          0008A3B895C40009B7FAD1E10800
          ARP           03:53:35
          Epoch: 0
...OUTPUT OMITTED...

```

Now that you have reviewed the different packet-switching options for a router, you can better analyze how a router is forwarding specific traffic. Following is a list of troubleshooting steps that you can follow if you suspect that network traffic is being impacted by a performance problem on one of the routers along the path from the source to the destination:



- Step 1.** Use the **tracert** command to determine which router along the path is causing excessive delay.
- Step 2.** After you identify a router that is causing unusually high delay, use the **show processes cpu** command to see the CPU utilization of that router and identify any processes that might be consuming an unusually high percentage of the CPU.
- Step 3.** Use the **show ip route ip_address** command to verify that the router has a route to the destination IP address.
- Step 4.** Use the **show ip cef** command to determine whether all the router interfaces are configured to use CEF.
- Step 5.** Use the **show ip cef ip_address 255.255.255.255** command to verify that CEF has an entry in its FIB that can reach the specified IP address. Part of the output from this command will be the next-hop adjacency to which traffic should be forwarded, along with the egress interface used to send traffic to that next hop.
- Step 6.** Issue the **show adjacency interface_type interface_number detail** command to verify that CEF has an entry in its adjacency table for the egress interface identified in Step 5.
- Step 7.** With the **show ip arp** command, you can then confirm that the router knows the MAC address associated with the next-hop IP address shown in the output from Step 6.

Step 8. You can then connect to the next-hop device and verify that the MAC address identified in Step 7 is indeed correct.

You can repeat these steps on the next-hop device or on another router whose response time displayed in the output from Step 1 is suspect.

Excessive Memory Utilization



Much like a PC, router performance can suffer if it lacks sufficient available memory. For example, perhaps you install a version of Cisco IOS on a router, and that router does not have the minimum amount of memory required to support that specific Cisco IOS image. Even though the router might load the image and function, its performance might be sluggish. Assuming that a router does have the recommended amount of memory for its installed Cisco IOS image, consider the following as potential memory utilization issues.

Memory Leak

When a router starts a process, that process can allocate a block of memory. When the process completes, the process should return its allocated memory to the router's pool of memory. If not all allocated memory is returned to the router's main memory pool, a memory leak occurs. Such a condition usually results from a bug in the Cisco IOS version running on the router, requiring an upgrade of the router's Cisco IOS image.

Example 3-22 shows sample output from the **show memory allocating-process totals** command. This command can help identify memory leaks. The output shows information about memory availability on a router after the Cisco IOS image of the router has been decompressed and loaded, and the total amount of memory that is being used by the various processes.

Example 3-22 **show memory allocating-process totals** Command Output

```
R4#show memory allocating-process totals
```

	Head	Total (b)	Used (b)	Free (b)	Lowest (b)	Largest (b)
Processor	83D27480	67463064	15347168	52115896	50311080	50127020
I/O	7C21800	4057088	2383016	1674072	1674072	1674044

Allocator PC Summary for: Processor

PC	Total	Count	Name
0x809D7A30	1749360	180	Process Stack
0x80A7F664	918024	10	Init
0x81CEF6A0	882576	4	pak subblock chunk
0x81C04D9C	595344	54	TCL Chunks
0x800902A4	490328	6	MallocLite

...OUTPUT OMITTED...

The *Head* column in the output refers to the address (in hexadecimal) of the memory allocation chain. The *Total* column is the total amount of memory available in bytes.

The *Used* column indicates how much has been used, and *Free* indicates how much is remaining. The *Lowest* column shows the lowest amount of free memory (in bytes) that has been available since the router last booted. The *Largest* column indicates the largest block of available memory. Following this summary information, the output shows detailed memory allocation information for each process running on a router. If a process is consuming a larger-than-normal amount of memory, it is likely because of a memory leak. A memory leak occurs when a process does not free the memory that it is finished using. Therefore, the block of memory remains reserved and will be released only when the router is reloaded. Typically, memory leaks result from bugs or poor coding in the Cisco IOS Software. The best solution is to upgrade the Cisco IOS Software to a version that fixes the issue.

Memory-Allocation Failure

A memory-allocation failure (which produces a `MALLOCFAIL` error message) occurs when a process attempts to allocate a block of memory and fails to do so. One common cause for a `MALLOCFAIL` error is a security issue. For example, a virus or a worm that has infested the network can result in a `MALLOCFAIL` error. Alternatively, a `MALLOCFAIL` error might result from a bug in the router's version of Cisco IOS. You can use the Cisco Bug Toolkit (available from www.cisco.com/cgi-bin/Support/Bugtool/launch_bugtool.pl) to research any such known issues with the version of Cisco IOS running on a router. Personally, I have witnessed the `MALLOCFAIL` error message when using an Integrated Services Router (ISR) that was running Network Address Translation (NAT), and another instance when I tried to load the complete Intrusion Prevention System (IPS) Signature Definition File on another ISR when I knew it could not handle it.

Buffer Leak

Similar to a memory leak, in which a process does not return all of its allocated memory to the router upon terminating, a buffer leak occurs when a process does not return a buffer to the router when the process has finished using the buffer. Consider the output of the `show interfaces` command shown in Example 3-23.

Example 3-23 Identifying a Wedged Interface

```
R4#show interfaces
...OUTPUT OMITTED...
  Input queue: 76/75/780/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
...OUTPUT OMITTED...
```

Notice the numbers 76 and 75 highlighted in the output. These values indicate that an input queue of the interface has a capacity of 75 packets and that the queue currently has 76 packets. These values indicate an oversubscription of the queue space. An interface in this condition is called a *wedged interface*. In such a condition, the router does not forward traffic coming into the wedged interface.

The **show buffers** command can also help to diagnose a buffer leak. To illustrate, consider the output of the **show buffers** command shown in Example 3-24.

Example 3-24 **show buffers** Command Output

```
R4#show buffers
Buffer elements:
    1118 in free list (500 max allowed)
    570 hits, 0 misses, 1119 created

Public buffer pools:
Small buffers, 104 bytes (total 71, permanent 50, peak 71 @ 00:21:43):
    53 in free list (20 min, 150 max allowed)
    317 hits, 7 misses, 0 trims, 21 created
    0 failures (0 no memory)
Middle buffers, 600 bytes (total 49, permanent 25, peak 49 @ 00:21:43):
    5 in free list (10 min, 150 max allowed)
    122 hits, 8 misses, 0 trims, 24 created
...OUTPUT OMITTED...
```

This output indicates that the router has 49 middle buffers, but only 5 of those 49 buffers are available. Such a result might indicate a process allocating buffers but failing to deallocate them. Like a memory leak, a buffer leak might require updating the Cisco IOS image of a router.

Excessive BGP Memory Use

If a router is running Border Gateway Protocol (BGP), be aware that BGP runs multiple processes and can consume significant amounts of router memory. The **show processes memory | include BGP** command, as shown in Example 3-25, can show you how much memory the various BGP processes of a router are consuming. If BGP is consuming a large percentage of your router memory, you might consider filtering out unneeded BGP routes, upgrading the memory on that router, or running BGP on a different platform that has more memory.

Example 3-25 **show processes memory | include BGP** Command Output

R1#show processes memory include BGP ^ PID							
PID	TTY	Allocated	Freed	Holding	Getbufs	Retbufs	Process
184	0	0	0	7096	0	0	BGP Task
198	0	0	0	10096	0	0	BGP Scheduler
229	0	38808	0	11520	0	0	BGP Router
231	0	0	0	10096	0	0	BGP I/O
262	0	0	0	10096	0	0	BGP Scanner
284	0	0	0	7096	0	0	BGP Event

Depending on the router platform, your router might have multiple line cards with different amounts of memory available on each line card. The **show diag** command can help you isolate a specific line card that is running low on memory, perhaps because that line card is running BGP.

Exam Preparation Tasks

As mentioned in the section “How to Use This Book” in the Introduction, you have several choices for exam preparation: the exercises here; Chapter 22, “Final Preparation;” and the exam simulation questions on the CD-ROM.

Review All Key Topics

Review the most important topics in this chapter, noted with the Key Topic icon in the outer margin of the page. Table 3-5 lists a reference of these key topics and the page numbers on which each is found.



Table 3-5 *Key Topics for Chapter 3*

Key Topic Element	Description	Page Number
List	Components in a Catalyst switch	96
Table 3-2	Errors in the <code>show interfaces interface_type interface_number counters errors</code> command	98
List	Reasons why a packet could be punted from a switch's TCAM to its CPU	102
Section	High CPU utilization troubleshooting on a switch	105
List	Identifies processes that cause excessive router CPU utilization	107
Table 3-3	Commands for troubleshooting high CPU utilization	108
List	Three primary modes of packet switching	113
Table 3-4	Commands for troubleshooting a router's packet-switching modes	116
Step list	Example of troubleshooting the forwarding of packets	120
Section	Excessive memory utilization	121

Define Key Terms

Define the following key terms from this chapter and check your answers in the glossary:

backplane, control plane, forwarding logic, ingress port, egress port, half-duplex, full-duplex, TCAM, ARP Input process, Net Background process, IP Background process, TCP Timer process, process switching, fast switching, CEF, memory leak, memory-allocation failure, buffer leak

Complete Tables and Lists from Memory

Print a copy of Appendix C, “Memory Tables,” (found on the disc), or at least the section for this chapter, and complete the tables and lists from memory. Appendix D, “Memory Tables Answer Key,” also on the disc, includes completed tables and lists to check your work.

Command Reference to Check Your Memory

This section includes the most important EXEC commands covered in this chapter. It might not be necessary to memorize the complete syntax of every command, but you should be able to remember the basic keywords that are needed.

To test your memory of the commands, cover the right side of Table 3-6 with a piece of paper, read the description on the left side, and then see how much of the command you can remember.

The 300-135 TSHOOT exam focuses on practical, hands-on skills that are used by a networking professional. Therefore, you should be able to identify the commands needed to verify router and switch configurations.

Table 3-6 *EXEC Commands*

Task	Command Syntax
A Cisco Catalyst 3750E series switch command that can be used to verify the maximum and used TCAM resources for various services and features on the switch.	<code>show platform tcam utilization</code>
A Cisco Catalyst switch command that can be used to display the current SDM template being used on the switch.	<code>show sdm prefer</code>
Displays a router's ARP cache. (Note: If a large number of the entries are in the Incomplete state, you might suspect a malicious scan [for example, a ping sweep] of a subnet.)	<code>show ip arp</code>
Shows a collection of interface statistics. (Note: If the throttles, overruns, or ignored counters continually increment, you might suspect that the Net Background process is attempting to allocate buffer space for an interface from the router's main buffer pool.)	<code>show interface interface_type interface_number</code>
Provides information about the number of TCP segments a router sends and receives, including the number of connections initiated, accepted, established, and closed. (Note: A high number of connections might explain why the TCP Timer process is consuming excessive CPU resources.)	<code>show tcp statistics</code>

Task	Command Syntax
Displays average CPU utilization over 5-second, 1-minute, and 5-minute intervals, in addition to listing all the router processes and the percentage of CPU resources consumed by each of those processes.	show processes cpu
Shows a graphical view of CPU utilization over the past 60 seconds, 1 hour, and 3 days. (Note: This graphical view can indicate whether an observed high CPU utilization is a temporary spike in utilization or whether the high CPU utilization is an ongoing condition.)	show processes cpu history
Displays multiple interface statistics, including information about the packet-switching mode of an interface.	show ip interface <i>interface_type</i> <i>interface_number</i>
Shows the contents of the fast cache for a router if fast switching is enabled.	show ip cache
Displays information about the IP Input process on a router. (Note: The CPU utilization for this process might show a high value if the CPU of a router is actively engaged in process-switching traffic.)	show processes cpu include IP Input
Displays the router's Layer 3 forwarding information, in addition to multicast, broadcast, and local IP addresses.	show ip cef
Verifies that a valid adjacency exists for a connected host.	show adjacency
Displays destinations reachable through the combination of the specified egress interface and next-hop IP address.	show ip cef adjacency <i>egress_interface_id</i> <i>next_hop_ip_address</i> detail
Provides information contained in a router's adjacency table, including protocol and timer information.	show adjacency detail
Displays information about packets forwarded by the router using a packet-switching mechanism other than CEF.	show cef not-cef-switched
Shows information about memory availability on a router after the router's Cisco IOS image has been decompressed and loaded. (Note: This command can help identify memory leaks.)	show memory allocating-process totals
Shows how many buffers (of various types) are currently free. (Note: This command can be helpful in diagnosing a buffer leak.)	show buffers
Shows how much memory is being consumed by the various BGP processes of a router.	show processes memory include bgp
Shows the memory available on the line cards of a router.	show diag

This page intentionally left blank



This chapter covers the following topics:

- **Frame-Forwarding Process:** This section reviews the Layer 2 frame-forwarding process. To successfully troubleshoot Layer 2 issues, you need to have a complete understanding of this process.
- **Troubleshooting Trunks:** This section focuses on how to troubleshoot Layer 2 trunking issues.
- **Troubleshooting VTP:** This section focuses on how to troubleshoot issues relating to VLAN Trunking Protocol.
- **Troubleshooting VLANs:** This section identifies how to troubleshoot general issues relating to VLANs and end-user port assignments.
- **The MAC address table:** This section reviews how to use the MAC address table during your troubleshooting process.
- **Layer 2 Trouble Tickets:** This section provides trouble tickets that demonstrate how you can use a structured troubleshooting process to solve a reported problem.

Troubleshooting Layer 2 Trunks, VTP, and VLANs

Most enterprise LANs rely on some flavor of Ethernet technology (for example, Ethernet, Fast Ethernet, or Gigabit Ethernet). In addition, your overall campus design will determine whether you need to worry about Layer 2 technologies such as trunks, Virtual Trunking Protocol (VTP), Dynamic Trunking Protocol (DTP), and virtual local-area networks (VLANs). If your campus design has any Layer 2 links from the distribution layer to the access layer, you need to have the skills necessary to troubleshoot these Layer 2 technologies.

However, before you master the skills for troubleshooting these Layer 2 technologies, you need to have an understanding of Ethernet switch operations at Layer 2. This chapter sets the stage by reviewing basic Layer 2 switch operations, which will factor into discussions in future chapters. It then moves on to troubleshooting trunks, VTP, and VLANs.

“Do I Know This Already?” Quiz

The “Do I Know This Already?” quiz allows you to assess whether you should read this entire chapter thoroughly or jump to the “Exam Preparation Tasks” section. If you are in doubt about your answers to these questions or your own assessment of your knowledge of the topics, read the entire chapter. Table 4-1 lists the major headings in this chapter and their corresponding “Do I Know This Already?” quiz questions. You can find the answers in Appendix A, “Answers to the ‘Do I Know This Already?’ Quizzes.”

Table 4-1 “Do I Know This Already?” Section-to-Question Mapping

Foundation Topics Section	Questions
Frame-Forwarding Process	1–3
Troubleshooting Trunks	4–6
Troubleshooting VTP	7
Troubleshooting VLANs	8
The MAC Address Table	9–10

Caution The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you do not know the answer to a question or are only partially sure of the answer, you should mark that question as wrong for purposes of the self-assessment. Giving yourself credit for an answer that you correctly guess skews your self-assessment results and might provide you with a false sense of security.

1. Which header information is used by switches to learn which MAC address is reachable out a specific interface?
 - a. Source IP address
 - b. Destination IP address
 - c. Source MAC address
 - d. Destination MAC address
2. Which header information is used by switches to forward frames?
 - a. Source IP address
 - b. Destination IP address
 - c. Source MAC address
 - d. Destination MAC address
3. What does a switch do with an unknown unicast frame?
 - a. Drop it
 - b. Forward it out the port it is associated with
 - c. Use ARP to determine the MAC address of the IP address in the packet
 - d. Flood it out all ports except the port it was received on
4. Which two are examples of issues that could prevent a trunk from forming?
 - a. Encapsulation mismatch
 - b. Incompatible trunking modes
 - c. Password mismatch
 - d. Missing VLAN
5. Which two of the trunk mode examples will successfully form a trunk?
 - a. Access – Dynamic desirable
 - b. Dynamic Auto – Dynamic auto
 - c. Trunk – Dynamic auto
 - d. Trunk – Trunk nonegotiate

6. Which command enables you to verify the administrative mode and operational mode of an interface?
 - a. `show interfaces trunk`
 - b. `show run interface interface_type interface_number`
 - c. `show interfaces interface_type interface_number switchport`
 - d. `show interfaces`
7. Which command enables you to verify VTP configurations?
 - a. `show run`
 - b. `show interfaces`
 - c. `show vtp status`
 - d. `show vtp configurations`
8. Which two commands enable you to verify which VLAN a port is assigned to?
 - a. `show vlan brief`
 - b. `show interfaces interface_type interface_number switchport`
 - c. `show interfaces trunk`
 - d. `show mac address-table dynamic`
9. Which command enables you to verify which port a MAC address is being learned on?
 - a. `show vlan brief`
 - b. `show interfaces interface_type interface_number switchport`
 - c. `show interfaces trunk`
 - d. `show mac address-table dynamic`
10. What can we confirm when examining the MAC address table of a switch? (Choose two answers.)
 - a. The port a MAC address was learned on
 - b. The VLAN the MAC address is associated with
 - c. The administrative and operational mode of an interface
 - d. The number of devices physically connected to an interface

Foundation Topics

Frame-Forwarding Process

To successfully troubleshoot Layer 2 forwarding issues, you need a solid understanding of how a switch operates. You would have learned this back in CCNA Routing and Switching. However, we spend time here reviewing switch operations because our troubleshooting efforts will be based on this knowledge. This section reviews how a switch populates its MAC address table and how it decides what to do with a frame based on the information in the MAC address table.



Unlike Ethernet hubs, which take bits in one port and send those same bits out all other ports, Ethernet switches learn about the devices connected to their ports. Therefore, when an Ethernet switch sees a frame destined for a particular MAC address, the switch can consult its MAC address table to determine which port to forward the newly arrived frame out. This behavior results in more-efficient bandwidth utilization and improved security on a LAN. In addition, it eliminates the concern of collisions. Specifically, in a hub environment, if two endpoints each transmitted a data frame at the same time, those two frames would collide, resulting in both frames being corrupted because all ports on a hub are in a common collision domain. This collision would require each endpoint to retransmit its data frame. This is not a concern with switches because every port on an Ethernet switch is in its own collision domain.

Ethernet switches can dynamically learn the MAC addresses attached to various switchports by looking at the source MAC address on frames coming into a port. For example, if switchport Gigabit Ethernet 1/1 received a frame with a source MAC address of DDDD.DDDD.DDDD, the switch could conclude that MAC address DDDD.DDDD.DDDD resided off of port Gigabit Ethernet 1/1. As a result, it places an entry in the MAC address table indicating so. In the future, if the switch received a frame destined for a MAC address of DDDD.DDDD.DDDD, the switch would only send that frame out of port Gigabit Ethernet 1/1 because of the entry in the MAC address table.

Initially, however, a switch is unaware of what MAC addresses reside off of which ports (unless MAC addresses have been statically configured). Therefore, when a switch receives a frame destined for a MAC address not yet present in the switch's MAC address table, the switch floods that frame out of all the switchports in the same VLAN, other than the port on which the frame was received. Similarly, broadcast frames (that is, frames with a destination MAC address of FFFF.FFFF.FFFF) are always flooded out all switchports in the same VLAN except the port on which the frame was received. The reason broadcast frames are always flooded is that no endpoint will have a MAC address of FFFF.FFFF.FFFF, meaning that the FFFF.FFFF.FFFF MAC address will never be learned dynamically in the MAC address table of a switch. In addition, if you look at the output of the MAC address table, you will notice that the all F's MAC address is statically bound to the CPU, ensuring that it can never be learned dynamically, as shown in Example 4-1.

Example 4-1 show mac address-table *Command Output*

```
SW1#show mac address-table
      Mac Address Table
-----
Vlan    Mac Address      Type    Ports
----    -
A11     0100.0ccc.cccc   STATIC  CPU
A11     0100.0ccc.cccd   STATIC  CPU
A11     0180.c200.0000   STATIC  CPU
A11     0180.c200.0001   STATIC  CPU
A11     0180.c200.0002   STATIC  CPU
A11     0180.c200.0003   STATIC  CPU
A11     0180.c200.0004   STATIC  CPU
A11     0180.c200.0005   STATIC  CPU
A11     0180.c200.0006   STATIC  CPU
A11     0180.c200.0007   STATIC  CPU
A11     0180.c200.0008   STATIC  CPU
A11     0180.c200.0009   STATIC  CPU
A11     0180.c200.000a   STATIC  CPU
A11     0180.c200.000b   STATIC  CPU
A11     0180.c200.000c   STATIC  CPU
A11     0180.c200.000d   STATIC  CPU
A11     0180.c200.000e   STATIC  CPU
A11     0180.c200.000f   STATIC  CPU
A11     0180.c200.0010   STATIC  CPU
A11     ffff.ffff.ffff   STATIC  CPU
10      0050.b60c.f258   DYNAMIC Gi0/1
10      0800.2757.1b86   DYNAMIC Gi0/1
10      0800.275d.06d6   DYNAMIC Fa0/1
10      0800.27a2.ce47   DYNAMIC Fa0/2
10      2893.fe3a.e301   DYNAMIC Gi0/1
...output omitted...
```

To illustrate how a switch's MAC address table becomes populated, consider an endpoint named PC1 that wants to form a Telnet connection with a server, as shown in Figure 4-1. Also, assume that PC1 and its server reside on the same subnet (that is, no routing is required to get traffic between PC1 and its server) and are therefore in the same VLAN, in this case VLAN 100. Before PC1 can send a Telnet segment to its server, PC1 needs to know the IP address (that is, the Layer 3 address) and the MAC address (that is, the Layer 2 address) of the server. The IP address of the server is typically known or is resolved via a Domain Name System (DNS) lookup. In this example, assume that the server's IP address is known. To properly communicate over Ethernet, PC1 needs to know the server's Layer 2 MAC address. If PC1 does not already have the server's MAC address in its Address Resolution Protocol (ARP) cache, PC1 can send an ARP request to learn the server's MAC address.

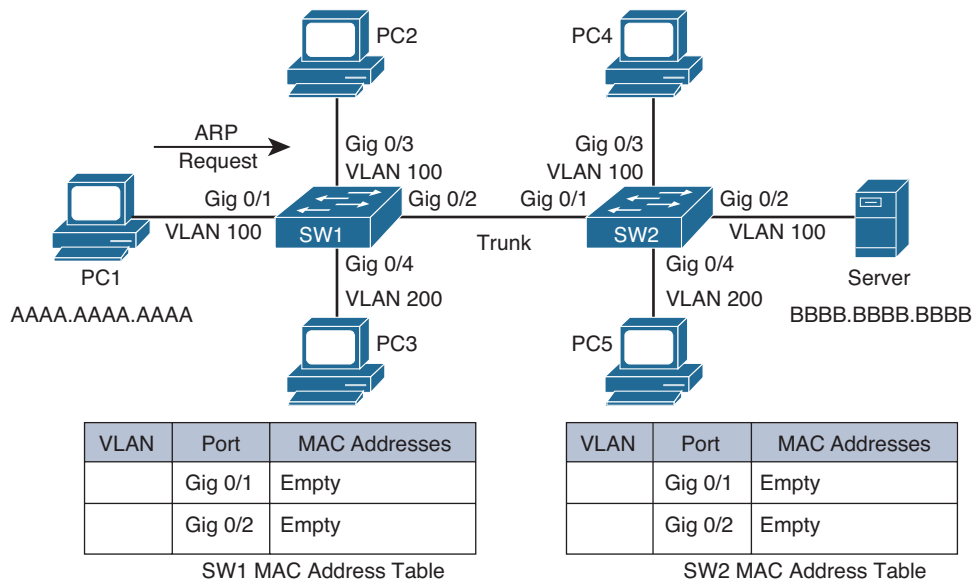


Figure 4-1 Endpoint Sending an ARP Request

When switch SW1 sees PC1's ARP request enter port Gig0/1, the PC1 MAC address of AAAA.AAAA.AAAA is added to the MAC address table of switch SW1 and associated with interface Gig0/1. Because Gig0/1 is a member of VLAN 100, the MAC is also associated with VLAN 100. Because the ARP request is a broadcast, its destination MAC address is FFFF.FFFF.FFFF (all F's). As discussed earlier, frames with a destination of all F's will be copied and flooded out all switchports except the port on which the frame was received. However, notice that port Gig0/1 on switch SW1 belongs to VLAN 100, whereas port Gig0/4 belongs to VLAN 200. This is important because frames are constrained to the VLAN from which they originated unless routed by a Layer 3 device. Therefore, the broadcast frame in this case is not flooded out Gig0/4 because Gig0/4 is a member of a different VLAN. Port Gig0/2, however, is a trunk port, and a trunk can carry traffic for multiple VLANs. Therefore, the ARP request is flooded out of port Gig0/2 and Gig0/3, as illustrated in Figure 4-2. Because the ARP request is for the MAC of the server, PC2 will ignore the ARP request.

When switch SW2 receives the ARP request inbound on its Gig0/1 trunk port, the source MAC address of AAAA.AAAA.AAAA is added to switch SW2's MAC address table, associated with Gig0/1 and VLAN 100. Also, similar to the behavior of switch SW1, switch SW2 floods the broadcast frame out of port Gig0/3 (a member of VLAN 100) and out of port Gig0/2 (also a member of VLAN 100), as depicted in Figure 4-3.

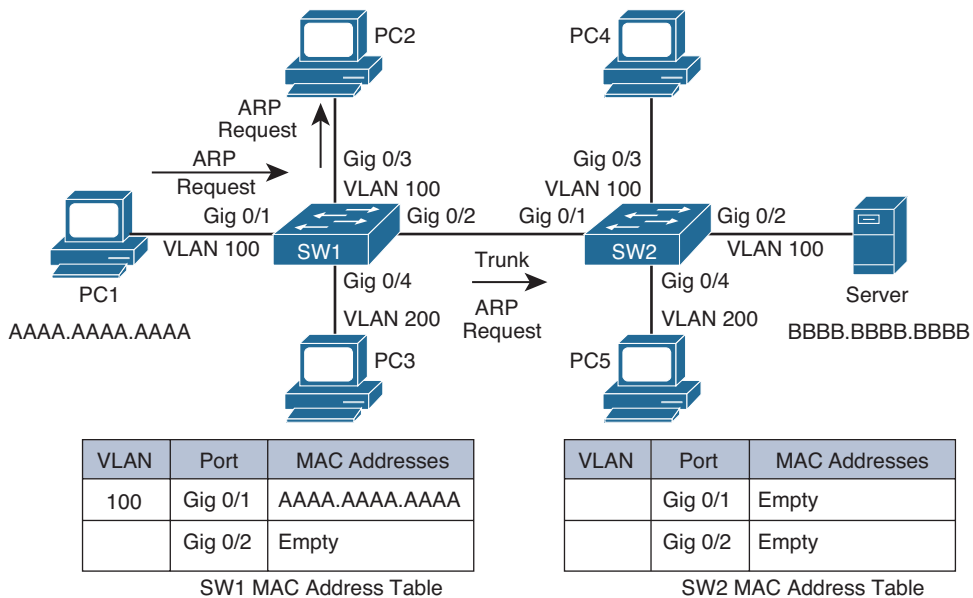


Figure 4-2 Switch SW1 Flooding the ARP Request

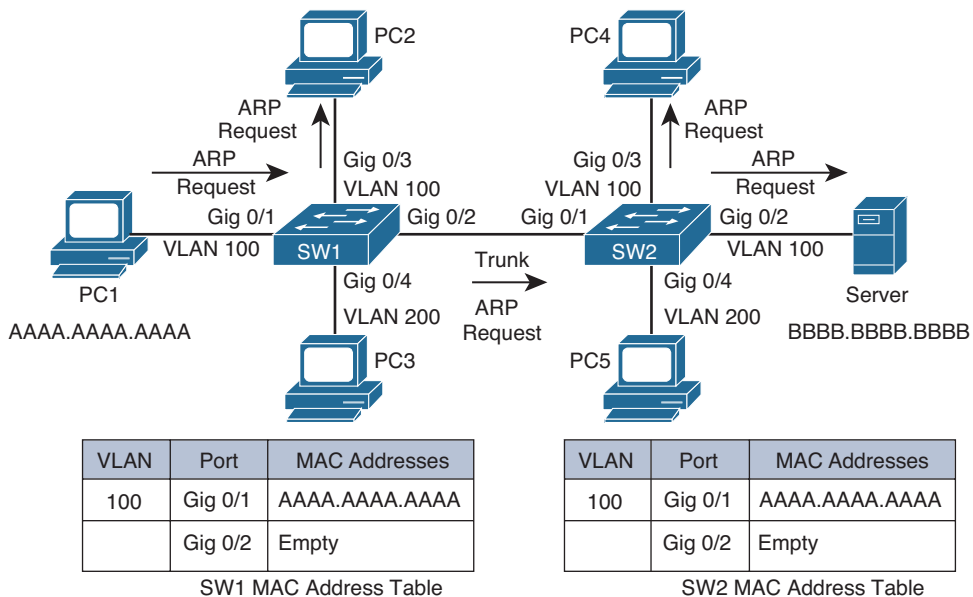


Figure 4-3 Switch SW2 Flooding the ARP Request

The server receives the ARP request and responds with an ARP reply, as shown in Figure 4-4. In addition, the server updates its ARP cache with a mapping of the IP and MAC address of PC1. Unlike the ARP request, the ARP reply frame is not a broadcast frame; it is a unicast frame. The ARP reply in this case has a destination MAC address of AAAA.AAAA.AAAA and a source MAC address of BBBB.BBBB.BBBB.

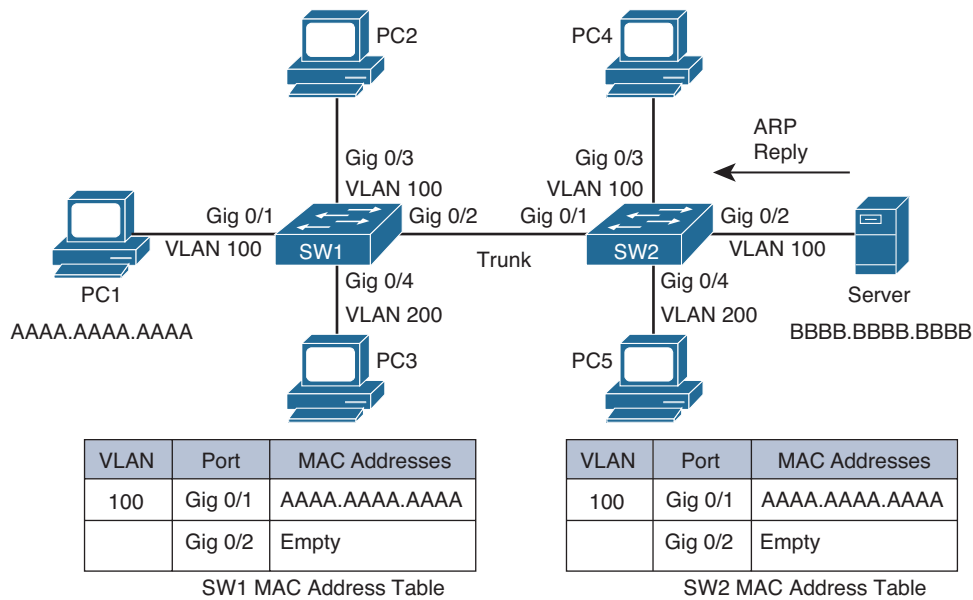


Figure 4-4 ARP Reply Sent from the Server

Upon receiving the ARP reply from the server, switch SW2 adds the server's MAC address of BBBB.BBBB.BBBB to its MAC address table, as shown in Figure 4-5. Also, the ARP reply is sent out only port Gig0/1 because switch SW2 knows that the destination MAC address of AAAA.AAAA.AAAA is reachable out port Gig0/1.

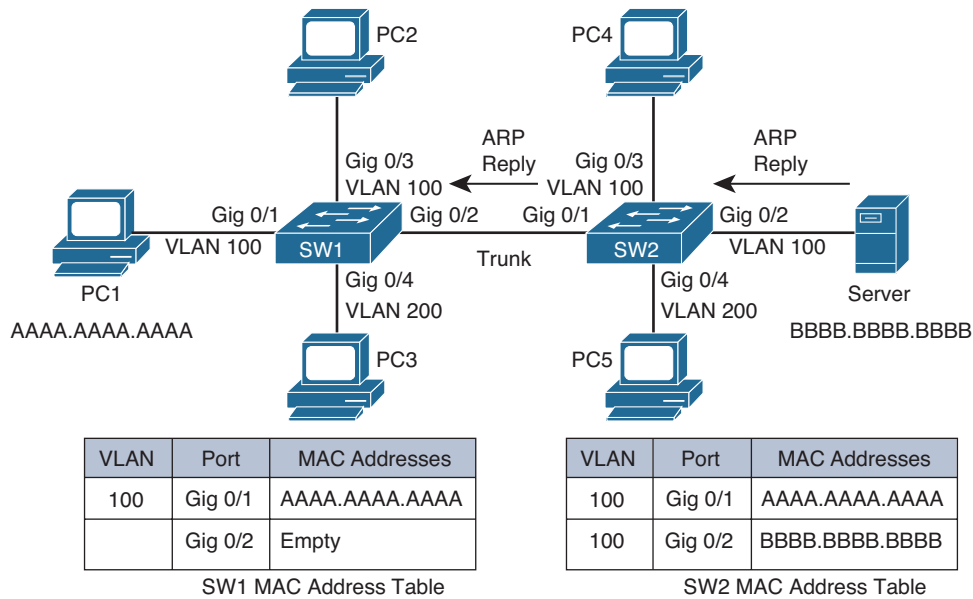


Figure 4-5 Switch SW2 Forwarding the ARP Reply

When receiving the ARP reply in its Gig0/2 port, switch SW1 adds the server's MAC address of BBBB.BBBB.BBBB to its MAC address table. Also, like switch SW2, switch SW1 now has an entry in its MAC address table for the frame's destination MAC address of AAAA.AAAA.AAAA. Therefore, switch SW1 forwards the ARP reply out port Gig0/1 to the endpoint of PC1, as illustrated in Figure 4-6.

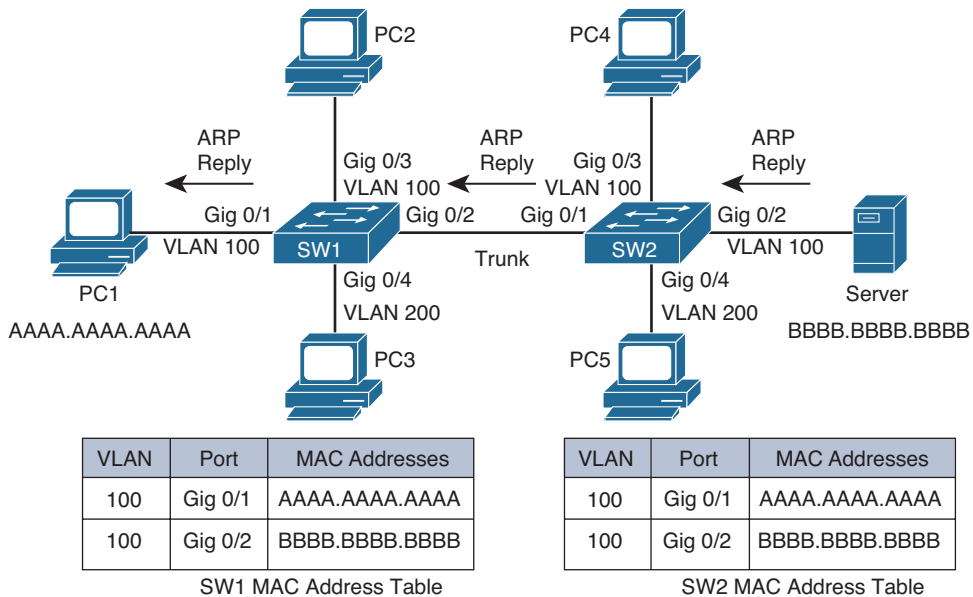
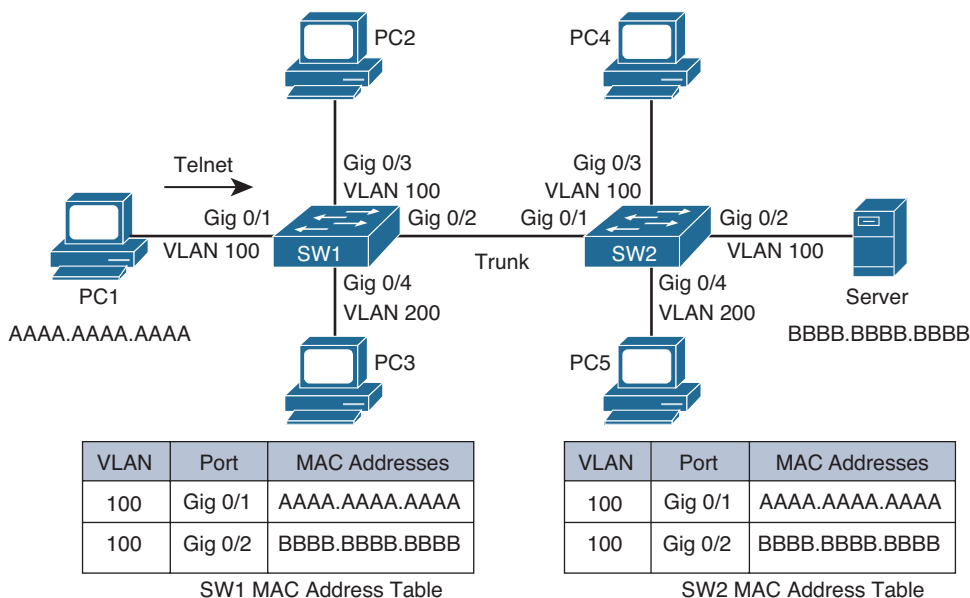
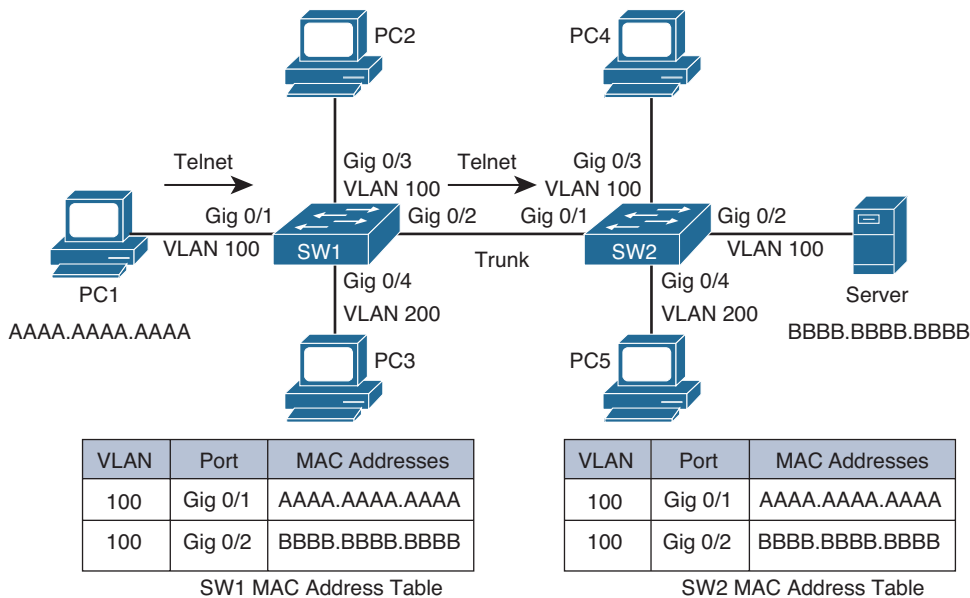


Figure 4-6 Switch SW1 Forwarding the ARP Reply

After receiving the server's ARP reply, PC1 now knows the MAC address of the server. Therefore, PC1 can send a properly constructed Telnet segment destined for the server, as depicted in Figure 4-7. The source MAC of the Layer 2 frame will be AAAA.AAAA.AAAA, and the destination MAC will be BBBB.BBBB.BBBB.

Switch SW1 has the server's MAC address of BBBB.BBBB.BBBB in its MAC address table. Therefore, when switch SW1 receives the frame from PC1, that frame is forwarded out of the Gig0/2 port of switch SW1, as shown in Figure 4-8.

**Figure 4-7** PC1 Sending a Telnet Segment**Figure 4-8** Switch SW1 Forwarding the Telnet Segment

Similar to the behavior of switch SW1, switch SW2 forwards the frame out its Gig0/2 port. This forwarding, shown in Figure 4-9, is possible because switch SW2 has an entry for the segment's destination MAC address of BBBB.BBBB.BBBB in its MAC address table.

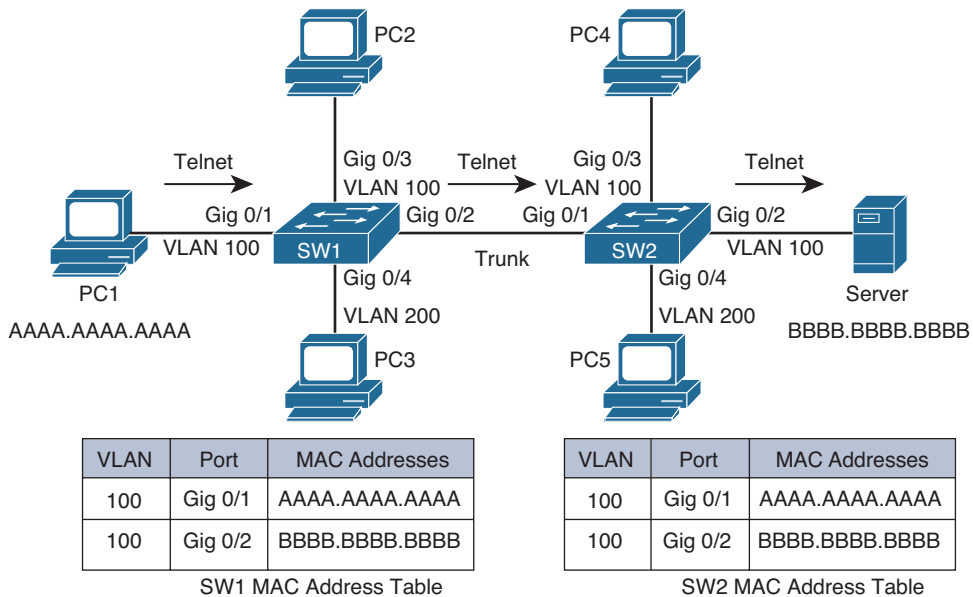


Figure 4-9 Switch SW2 Forwarding the Telnet Segment

Finally, the server responds to PC1, and a bidirectional Telnet session is established between the PC and the server, as illustrated in Figure 4-10. Because PC1 learned the MAC address of the server and the server learned the MAC address of PC1, as a result of PC1's earlier ARP request, both devices stored the MAC addresses in their local ARP caches; therefore, the transmission of subsequent Telnet segments does not require additional ARP requests. However, if unused for a period of time, entries in a device's ARP cache will time out.

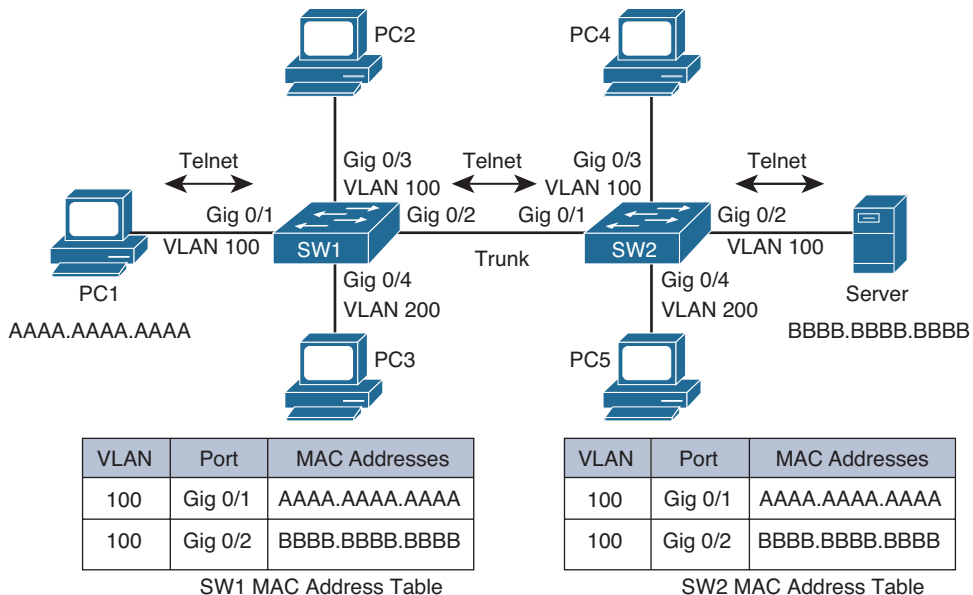


Figure 4-10 Bidirectional Telnet Session Between PC1 and the Server

When troubleshooting an issue involving Layer 2 switch communication, a thorough understanding of the preceding steps can help you identify potential problems quickly and efficiently. Take a moment and review Figure 4-10. Consider where issues might arise in the topology that would prevent PC1 and Server from communicating. The following list outlines a few potential issues that could arise:



- PC1 and Server have IP addresses in different subnets because of incorrect address or subnet mask.
- Interface Gig0/1 on SW1 or Gig0/2 on SW2 are not members of the correct VLAN.
- VLAN 100 is missing on SW1 or SW2.
- The trunk between SW1 and SW2 is not passing traffic for the necessary VLANs (VLAN 100 in this case).
- The trunk is not formed between SW1 and SW2.
- A VACL is denying PC1 from communicating with Server.
- Interface Gig0/1 on SW1, Gig0/2 on SW2, or the trunk interfaces are shut down or in the err-disabled state.

Troubleshooting Trunks

Trunks support multiple VLANs on a single physical link. A trunk can be between two switches, a switch and a router, and a switch and a server that is providing services for multiple VLANs. This section focuses on issues that prevent a trunk from being formed or passing traffic for a VLAN. Figure 4-11 serves as the topology for all of the examples.

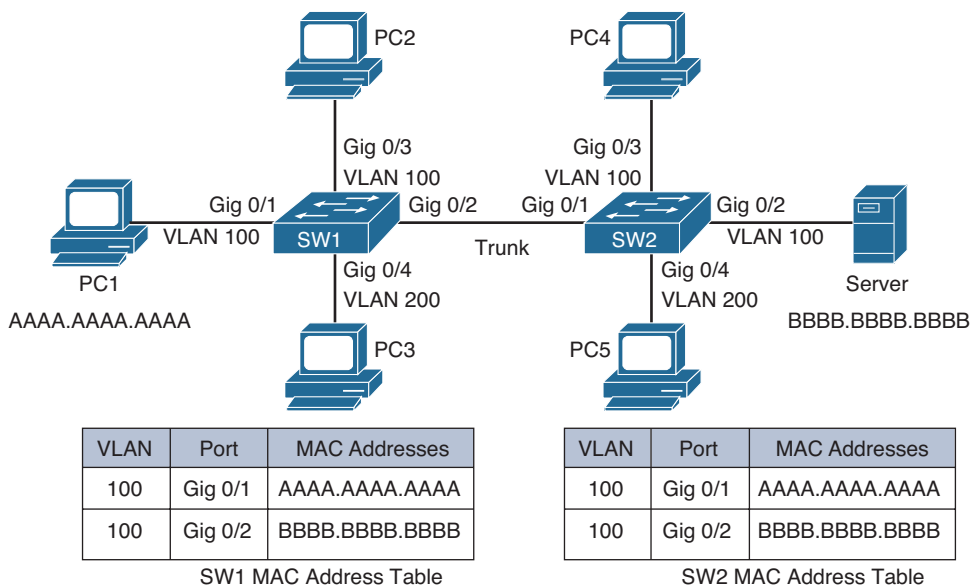


Figure 4-11 *Troubleshooting Trunks*

Encapsulation Mismatch

Two types of trunking encapsulations are supported by Cisco Catalyst switches: 802.1Q, which is an IEEE standard; and ISL (Inter-Switch Link), which is Cisco proprietary. 802.1Q adds a 4-byte tag to the Ethernet frame, whereas ISL encapsulates the entire Ethernet frame, resulting in an additional 30 bytes. Not all switches support both. For example, a Catalyst 2960 switch supports only 802.1Q, whereas a Catalyst 3560 and a Catalyst 3750-E support both. To form a trunk between two switches, the interfaces that will be forming the trunk must be using the same encapsulation type. By default, Cisco Catalyst switches that support only 802.1Q will use 802.1Q, Catalyst switches that support both 802.1Q and ISL will autonegotiate the encapsulation using DTP. Therefore, if you connect a Catalyst 2960 and a Catalyst 3750-E together, they will use 802.1Q because that is all the Catalyst 2960 can support. However, if you connect two 3750-Es together, they will negotiate the use of ISL because it is Cisco proprietary. If you are required to use 802.1Q trunks in your environment, you must manually change it from ISL to 802.1Q in that situation.

Because autonegotiation of encapsulation works very well, you will usually only have an encapsulation mismatch if someone is manually setting the trunking encapsulation. To verify the encapsulation type used on an interface, issue the **show interfaces interface_type interface_number switchport** command, as shown in Examples 4-2 and 4-3.



Example 4-2 *Output of show interfaces switchport Command on SW1 to Verify Encapsulation*

```
SW1#show interfaces gigabitethernet 0/2 switchport
Name: Gi0/2
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: Off
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 99 (NATIVE)
Administrative Native VLAN tagging: enabled
Voice VLAN: none
...output omitted...
```



Example 4-3 *Output of show interface switchport Command on SW2 to Verify Encapsulation*

```
SW2#show interfaces gigabitethernet 0/1 switchport
Name: Gi0/1
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: isl
```

```

Operational Trunking Encapsulation: isl
Negotiation of Trunking: Off
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 99 (NATIVE)
Administrative Native VLAN tagging: enabled
Voice VLAN: none
...output omitted...

```

From the **show interfaces switchport** output of Example 4-2 and Example 4-3, you can see that SW1 and SW2 are not using the same trunking encapsulation. SW1 is using 802.1Q, and SW2 is using ISL. Therefore, a trunk will not successfully form in this case.

You can also verify which trunking encapsulation is being used by looking at the output of **show interfaces trunk**, as shown in Example 4-4 and Example 4-5.

Example 4-4 *Output of show interfaces trunk Command on SW1 to Verify Encapsulation*

```

SW1#show interfaces trunk

Port      Mode      Encapsulation  Status        Native vlan
Gi0/2     on        802.1q         trunking      99

Port      Vlans allowed on trunk
Gi0/2     1-4094

Port      Vlans allowed and active in management domain
Gi0/2     1,100,200

Port      Vlans in spanning tree forwarding state and not pruned
Gi0/2     1,100,200

```

Example 4-5 *Output of show interface trunk Command on SW2 to Verify Encapsulation*

```

SW2#show interfaces trunk

Port      Mode      Encapsulation  Status        Native vlan
Gi0/1     on        isl            trunking      99

Port      Vlans allowed on trunk
Gi0/1     1-4094

Port      Vlans allowed and active in management domain
Gi0/1     1,100,200

Port      Vlans in spanning tree forwarding state and not pruned
Gi0/1     1,100,200

```

Incompatible Trunking Modes

There are different administrative trunking modes an interface can be configured to use when forming a trunk, as follows:

- **Access:** In this administrative mode, a switchport is manually configured to never become a trunk even if DTP messages are received. This mode is designed for ports that are connecting to, for example, end stations, servers, and printers, where a trunk should never be required because only a single VLAN is needed. This mode can be verified as shown in Example 4-6.
- **Trunk:** In this administrative mode, a switchport is manually configured to always be a trunk. This mode can be verified as shown in Example 4-7.
- **Dynamic desirable:** In this administrative mode, a switchport is aggressively trying to become a trunk by negotiating with the other end of the link to form a trunk using DTP. If the other end of the link agrees then a trunk is formed; if not, it remains an access port that will listen for DTP messages in addition to periodically sending DTP messages as it continues to try and form a trunk. This mode can be verified as shown in Example 4-8.
- **Dynamic auto:** In this administrative mode, a switchport is passively waiting for DTP messages to arrive asking it to form a trunk. If it receives them, it will form a trunk. If it does not receive any, it remains an access port that will listen for DTP messages. This mode can be verified as shown in Example 4-9.

Key Topic

Example 4-6 Verifying Trunking Administrative Mode (Access)

```
SW1#show interfaces gigabitethernet 0/1 switchport
Name: Gi0/1
Switchport: Enabled
Administrative Mode: static access
Operational Mode: static access
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: native
Negotiation of Trunking: Off
Access Mode VLAN: 100 (VLAN100)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: enabled
Voice VLAN: none
...output omitted...
```

Key Topic

Example 4-7 Verifying Trunking Administrative Mode (Trunk)

```
SW1#show interfaces gigabitethernet 0/2 switchport
Name: Gi0/2
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
```

```
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: Off
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 99 (NATIVE)
Administrative Native VLAN tagging: enabled
Voice VLAN: none
...output omitted...
```

Example 4-8 Verifying Trunking Administrative Mode (Dynamic Desirable)

```
SW1#show interfaces gigabitethernet 0/2 switchport
Name: Gi0/2
Switchport: Enabled
Administrative Mode: dynamic desirable
Operational Mode: trunk
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 99 (NATIVE)
Administrative Native VLAN tagging: enabled
Voice VLAN: none
...output omitted...
```

Example 4-9 Verifying Trunking Administrative Mode (Dynamic Auto)

```
SW1#show interfaces gigabitethernet 0/2 switchport
Name: Gi0/2
Switchport: Enabled
Administrative Mode: dynamic auto
Operational Mode: trunk
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 99 (NATIVE)
Administrative Native VLAN tagging: enabled
Voice VLAN: none
...output omitted...
```

The default administrative mode varies by Catalyst switch model. To verify the default administrative mode on your model, issue the **show interfaces interface_type interface_number switchport** command for an interface that is still using factory default settings. Example 4-10 shows that interface Gigabit Ethernet 0/1 is using factory default settings, because no other configurations have been applied to the interface, as shown in the **show run interface gigabitethernet 0/1** output. The output of **show interfaces gigabitethernet 0/1 switchport | include Administrative Mode** indicates that the trunking administrative

mode is dynamic auto. Therefore, we can conclude dynamic auto is the default on this switch because there is no command in the running configuration that indicates otherwise.

Example 4-10 Verifying Default Trunking Mode on SW2

```
SW2#show run interface gigabitethernet 0/1
Building configuration...

Current configuration : 50 bytes
!
interface GigabitEthernet0/1
end

SW2#show interfaces gig 0/1 switchport | include Administrative Mode
Administrative Mode: dynamic auto
```

Some of these administrative modes are compatible with each other and will form a trunk, whereas others are not, as shown in Table 4-2. While you are looking at Table 4-12, remember that dynamic auto, dynamic desirable, and trunk all use DTP by default.

Table 4-2 Comparing Trunking Administrative Modes

		SW1				
SW2		Dynamic Auto	Dynamic Desirable	Trunk	Trunk Nonnegotiate	Access
	Dynamic Auto	Access	Trunk	Trunk	Limited connectivity	Access
	Dynamic Desirable	Trunk	Trunk	Trunk	Limited connectivity	Access
	Trunk	Trunk	Trunk	Trunk	Trunk	Limited connectivity
	Trunk Nonnegotiate	Limited connectivity	Limited connectivity	Trunk	Trunk	Limited connectivity
	Access	Access	Access	Limited connectivity	Limited connectivity	Access

As you can see in Table 4-2, if both switchports are configured as dynamic auto, a trunk will not form. The switchports will remain as access ports and pass traffic for the VLAN the port is a member of. To form a trunk with a switchport that is dynamic auto, the other switchport must be using dynamic desirable or trunk (using DTP). Limited connectivity is a result of one side being operationally a trunk and the other side being operationally an access port. Connectivity will occur only if the access port VLAN on one switch happens to be the same as the native VLAN for the 802.1Q trunk on the other

switch. If not, connectivity will be broken. The reason is because the access port sends the frames untagged, and once the trunk port receives them at the other end, it considers them as part of the native VLAN because of the lack of a tag. If these VLAN numbers match, the frames can be successfully forwarded without a problem. However, if the native VLAN does not match with the VLAN configured on the access port, the frames when entering or leaving the trunk port on the switch will be part of a different VLAN than the access port and the frames are no longer forwarded correctly, and connectivity is broken. Memorizing Table 4-2 will definitely prove beneficial if you ever have to troubleshoot trunk links that are not forming.

VTP Domain Name Mismatch

We will cover VTP in detail shortly. However, if you are using DTP to dynamically form trunks and the VTP domain name does not match between the two switches, a trunk will not be formed, as shown in Example 4-11.

Example 4-11 VTP Domain Name Mismatch Causes Trunk Not to Form

```
SW1#
%DTP-5-DOMAINMISMATCH: Unable to perform trunk negotiation on port Gi0/2 because of
VTP domain mismatch.
```

Native VLAN Mismatch

Trunk issues with the native VLAN only surface when we are using IEEE 802.1Q trunking encapsulation. The concept of a native VLAN does not exist with Cisco ISL trunking encapsulation. The native VLAN by default is VLAN 1 and is used to carry untagged traffic across an 802.1Q trunk. It is imperative that the native VLAN matches on both sides of a trunk link. If it does not, it is possible for traffic to leak from one VLAN to another, resulting in an undesired forwarding behavior and possible errors with Spanning Tree Protocol.

With a native VLAN mismatch, the trunk forms, and syslog messages are generated, as shown in Example 4-12. From the example, you can see that Cisco Discovery Protocol (CDP) is warning you about the native VLAN mismatch; however, if CDP is not enabled, this message would not appear. Example 4-13 displays the output of **show interfaces trunk** on SW1 and SW2, confirming that we have a native VLAN mismatch.



Example 4-12 Result of a Native VLAN Mismatch on a Trunk

```
SW1#
%CDP-4-NATIVE_VLAN_MISMATCH: Native VLAN mismatch discovered on GigabitEthernet0/2
(1), with SW2 GigabitEthernet0/1 (99).
SW2#
%CDP-4-NATIVE_VLAN_MISMATCH: Native VLAN mismatch discovered on GigabitEthernet0/1
(99), with SW1 GigabitEthernet0/2 (1).
```

Example 4-13 *Confirming the Native VLAN Mismatch with the show interfaces trunk Command*

```
SW1#show interfaces trunk
Port      Mode      Encapsulation  Status  Native vlan
Gi0/2     desirable n-802.1q       trunking 1
...output omitted...

SW2#show interfaces trunk
Port      Mode      Encapsulation  Status  Native vlan
Gi0/1     desirable n-802.1q       trunking 99
...output omitted...
```

Allowed VLANs**Key Topic**

By default, traffic for all VLANs will be forwarded on a trunk. You can modify this behavior by identifying which VLANs are allowed on the trunk. You can accomplish this manually or dynamically. If you are using VTP to propagate VLAN configuration information, you can use the VTP pruning feature, which dynamically determines which VLANs are needed on each of the trunks. You can enable VTP pruning with the **vtp pruning** global configuration command. Many prefer to control the VLANs allowed on trunks manually with the **switchport trunk allowed vlans *vlan_id*** command in interface configuration mode. You can verify which VLANs are allowed on a trunk a few different ways. You can use the **show interfaces trunk** command, the **show interface *interface_type interface_number* switchport** command, or review the interface configuration in the running configuration. Example 4-14 displays the output from these three commands. Focus on the highlighted text because it identifies which VLANs are allowed on the trunk. If traffic is not flowing across a trunk for a specific VLAN, make sure that the VLAN is allowed on the trunk.

Example 4-14 *Verifying Allowed VLANs on a Trunk*

```
SW1#show interfaces trunk
Port      Mode      Encapsulation  Status  Native vlan
Gi0/2     desirable n-802.1q       trunking 99

Port      Vlans allowed on trunk
Gi0/2     100,200

Port      Vlans allowed and active in management domain
Gi0/2     100,200

Port      Vlans in spanning tree forwarding state and not pruned
Gi0/2     100,200

SW1#show interfaces gigabitethernet 0/2 switchport
Name: Gi0/2
Switchport: Enabled
```



```

...output omitted...
Trunking VLANs Enabled: 100,200
Pruning VLANs Enabled: 2-1001
Capture Mode Disabled
...output omitted...

SW1#show run interface gigabitethernet 0/2
Building configuration...

Current configuration : 167 bytes
!
interface GigabitEthernet0/2
  switchport trunk native vlan 99
  switchport trunk allowed vlan 100,200
  switchport mode dynamic desirable
end

```

Troubleshooting VTP

Picture a network with 50 switches and 75 VLANs. You have been tasked with deploying these 75 VLANs to all 50 switches. This is a large task that is definitely prone to human error. VLAN Trunking Protocol (VTP) is designed to ease the deployment of VLAN configuration information between switches across trunk links. This section explains the reasons why VTP might not be sharing VLAN configuration information with other switches in the domain. Figure 4-11 is used as the topology for the examples. SW1 and SW2 need to have the same VLAN database.

Domain Name Mismatch

Switches that will learn VLAN configuration information from each other using VTP need to be in the same VTP domain. The VTP domain is identified by a name known as the VTP domain name, and it can be anything you want it to be. However, it must match on the devices that will be exchanging VLAN configuration information. Suppose, for example, that SW1 in Figure 4-11 is using a VTP domain name of TSHOOT and SW2 is using a VTP domain name of TSHOOT. Obviously, they match. What about SW1 using TSHOOT and SW2 using TSHOOT? It looks like they match, but they do not. The VTP domain name for SW2 has a zero (0) in it instead of the letter O. Compare Examples 4-15 and 4-16, which display the output of **show vtp status** on SW1 and SW2. Are SW1 and SW2 in the same VTP domain?



Example 4-15 *Verifying the VTP Domain Name on SW1*

```

SW1#show vtp status
VTP Version capable      : 1 to 3
VTP version running      : 3
VTP Domain Name          : Tshoot

```

```
VTP Pruning Mode           : Disabled
VTP Traps Generation      : Disabled
Device ID                 : 001c.57fe.f600
...output omitted...
```



Example 4-16 *Verifying the VTP Domain Name on SW2*

```
SW2#show vtp status
VTP Version capable       : 1 to 3
VTP version running       : 3
VTP Domain Name           : TSHOOT
VTP Pruning Mode          : Disabled
VTP Traps Generation      : Disabled
Device ID                 : 2893.fe3b.0100
...output omitted...
```

Note that case does matter for the VTP domain name. Therefore, SW1 and SW2 are in completely different VTP domains and will not share VLAN configuration information with each other. In addition, as mentioned earlier, if you are using DTP to form a trunk and you have a VTP domain name mismatch, a trunk will not form.

Version Mismatch

There are three versions of VTP: VTPv1, VTPv2, and VTPv3. VTPv1 is the default. If you are running VTPv1, all switches need to be using VTPv1 to successfully exchange VLAN configuration information. If you are running VTPv2 or VTPv3 the switches can be using VTPv2 or VTPv3 because they are compatible. However, to reduce the possibility of issues, it is recommended that you avoid mixing VTP versions. To verify the VTP version in use on a switch, issue the **show vtp status** command, as shown in Example 4-17. Also notice in the output that SW2 is capable of running all three versions of VTP.

Example 4-17 *Verifying the VTP Version on SW2*

```
SW2#show vtp status
VTP Version capable       : 1 to 3
VTP version running       : 3
VTP Domain Name           : TSHOOT
VTP Pruning Mode          : Disabled
VTP Traps Generation      : Disabled
Device ID                 : 2893.fe3b.0100
...output omitted...
```

Mode Mismatch

VTP has four modes of operation: Server, Client, Transparent, and Off. For a switch to use the VLAN configuration information in a VTP message, it must be in Server or Client mode. A switch operating in Transparent mode will ignore the information contained in

a VTP message; however, it will still forward on the message to other switches. In Off mode, the switch behaves the same as Transparent mode, except that it will not forward on VTP messages that it receives. Therefore, if you are troubleshooting an issue that involves missing VLANs on a switch and you are using VTP, check whether the switch is in VTP Transparent mode or Off. To verify the VTP mode used on a switch, issue the **show vtp status** command, as shown in Examples 4-18 and 4-19. In addition, with VTPv3, only the VTP primary server can add or delete VLANs.

Example 4-18 *Verifying the VTP Mode on SW1*

```
SW1#show vtp status
VTP Version capable      : 1 to 3
VTP version running      : 3
VTP Domain Name          : SWITCH
VTP Pruning Mode         : Disabled
VTP Traps Generation     : Disabled
Device ID                 : 2893.fe3b.0100

Feature VLAN:
-----
VTP Operating Mode       : Server
Number of existing VLANs : 10
Number of existing extended VLANs : 0
Maximum VLANs supported locally : 1005
Configuration Revision   : 3
...output omitted...
```

Example 4-19 *Verifying the VTP Mode on SW2*

```
SW2#show vtp status
VTP Version capable      : 1 to 3
VTP version running      : 3
VTP Domain Name          : SWITCH
VTP Pruning Mode         : Disabled
VTP Traps Generation     : Disabled
Device ID                 : 001c.57fe.f600

Feature VLAN:
-----
VTP Operating Mode       : Client
Number of existing VLANs : 10
Number of existing extended VLANs : 0
Maximum VLANs supported locally : 255
Configuration Revision   : 3
...output omitted...
```

Password Mismatch

To ensure that a switch only uses VTP configuration information from legitimate sources, it is recommended that a VTP password is set. When a switch receives a VTP message from another switch, it will verify that the attached message digest 5 (MD5) algorithm hash matches its local hash. If it matches, the VTP message is from a legitimate source and is processed. If not, the VTP message is discarded. Remember that the VTP password is case sensitive. Example 4-20 shows how you can verify the password that is configured with the **show vtp password** command and the hash value that will be used with the **show vtp status** command.

Example 4-20 Verifying VTP Passwords

```
SW1#show vtp password
VTP Password: CCNP

SW1#show vtp status
VTP Version capable      : 1 to 3
VTP version running      : 3
VTP Domain Name          : SWITCH
VTP Pruning Mode         : Disabled
VTP Traps Generation     : Disabled
Device ID                : 2893.fe3b.0100

Feature VLAN:
-----
VTP Operating Mode       : Server
Number of existing VLANs : 11
Number of existing extended VLANs : 0
Maximum VLANs supported locally : 1005
Configuration Revision   : 2
Primary ID               : 2893.fe3a.e300
Primary Description      : DSW1
MD5 digest               : 0x98 0x29 0xB8 0x5D 0x4D 0x48 0x71 0xE3
                        : 0x8A 0x93 0x8E 0x82 0x2B 0xEA 0xA0 0x45
...output omitted...
```

Higher Revision Number

When a switch in VTP server mode makes a change to the VLAN database, it increments the configuration revision number shown in Example 4-20. Currently it is 2, but if another VLAN were added or a modification were made that affected the VLAN database, VTP would increment the configuration revision number. This number is extremely important because the switch with the higher configuration revision number is considered to have the most up-to-date and valid VLAN database. However, this might not always be the case. For example, suppose that you are preparing for the TSHOOT exam and you are troubleshooting VLANs. You keep adding and deleting VLANs while using

VTPv1 to propagate your changes to the other switches in your lab pod. Now you have a really high configuration revision number. The next day a coworker plugs your lab pod into the production network, and your lab VLAN database overwrites the VLAN database of the production network because you were using the same domain name and password on your lab devices and the lab had a higher configuration revision number than the production network. Now you need to rebuild the production VLAN database or restore it from backup, if you have one.

You need to prevent this from ever happening by ensuring no one uses the same VTP domain name or password on other devices and then plugs them into the production network. However, that is hard to control. So, it is better to run all the switches in Transparent mode and only use Server or Client mode when you are building the VLAN database or making significant changes that have to be propagated to all the other switches. This is because Transparent mode switches will not update their VLAN information from VTP messages, protecting you from having your VLAN database overwritten. You may also want to consider having all switches in VTP Transparent mode when they are added to the domain so that their configuration revision number is 0, which it always is for Transparent mode. Your best option is to use VTPv3 because only the VTP primary server will be considered a trusted source of VTP messages within the VTP domain, and any other VTP messages will be ignored, ensuring that your database is not overwritten by a rogue switch.

Troubleshooting VLANs

Our discussions have led us to this important point in this chapter: Being able to identify and solve issues with VLANs. This is an important task for any troubleshooter. Some of these issues could be a result of a trunk or VTP issue, as previously discussed. This section identifies the issues that might arise with VLANs and how you can fix them. The discussion is based on Figure 4-11.

Incorrect IP Addressing

It all starts with the client configuration. If the IP address, subnet mask, or default gateway are not configured correctly, frames will not flow as expected. Example 4-21 displays the output of **ipconfig** on PC1 and Server. If you look closely, you will notice that Server is not addressed correctly, and therefore not in the same subnet. When PC1 needs to send data to Server, because they are not on the same subnet, PC1 will send the frame to its default gateway so that it can be routed to a different subnet. However, this process will fail at some point because both PC1 and Server cannot be in the same Layer 2 VLAN (as Figure 4-11 shows), within different IP networks. They need to be in the same subnet if they are in the same VLAN so that frames can be sent from PC1 directly to Server based on the Layer 2 MAC addresses.

Example 4-21 *Verifying End-User IP Addresses*

```

PC1>ipconfig
Windows IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . : 
    IP Address. . . . .               : 10.1.100.10
    Subnet Mask . . . . .             : 255.255.255.0
    Default Gateway . . . . .         : 10.1.100.1

Server>ipconfig
Windows IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . : 
    IP Address. . . . .               : 10.1.10.11
    Subnet Mask . . . . .             : 255.255.255.0
    Default Gateway . . . . .         : 10.1.10.1

```

Missing VLAN

For a switch to associate switchports with VLANs or to pass traffic over a trunk for a VLAN, the switch needs to know about the VLAN. The command **show vlan brief**, as shown in Example 4-22, displays the VLANs that are known by the switch.

**Example 4-22** *Verifying VLANs on a Switch*

```

SW1#show vlan brief

```

VLAN	Name	Status	Ports
1	default	active	Gi0/5, Gi0/6, Gi0/7, Gi0/8, Gi0/9, Gi0/10, Gi0/11, Gi0/12, Gi0/13, Gi0/14, Gi0/15, Gi0/16, Gi0/17, Gi0/18, Gi0/19, Gi0/20, Gi0/21, Gi0/22, Gi0/23, Gi0/24, Te1/0/1, Te1/0/2
99	NATIVE	active	
100	10.1.100.0/24	active	Gi0/1, Gi0/3
200	10.1.200.0/24	active	Gi0/4
1002	fddi-default	act/unsup	
1003	trcrf-default	act/unsup	
1004	fddinet-default	act/unsup	
1005	trbrf-default	act/unsup	

If any VLANs are missing from the output of **show vlan brief** that should be there, you need to find out why. If VLANs are configured manually in your organization, the answer is one of two reasons: Someone forgot to configure the VLAN on the switch, or someone deleted the VLAN on the switch. If the creation and deletion of VLANs is learned by other switches through VTP, you need to troubleshoot why VTP is not propagating the VLAN information to the other switches. However, it is important to remember that if you are using VTPv1 or 2 and a switch is added to the domain with the correct password, and has a higher revision number, the VLAN database in your VTP domain will be overwritten by this switch. Therefore, if you are missing VLANs, this could be the reason why.

In Example 4-23, which displays the output of **show interfaces gigabitethernet 0/1 switchport**, focus on the highlighted text. Notice in brackets the name of the VLAN. It is listed as (Inactive). This is a great sign that the interface belongs to a VLAN that does not currently exist on the switch. Note that even though the port is up/up, because the VLAN does not exist, the port will not be forwarding traffic.

Example 4-23 Identifying Missing VLANs on a Switch

```
SW1#show interfaces gigabitethernet 0/1 switchport
Name: Gi0/1
Switchport: Enabled
Administrative Mode: static access
Operational Mode: static access
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: native
Negotiation of Trunking: Off
Access Mode VLAN: 100 (Inactive)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: enabled
Voice VLAN: none
```

Incorrect Port Assignment



Once VLANs are created, switchports need to be assigned to VLANs. The assignments should be based on which device is going to be connected to that port (based on IP address, subnet mask, and default gateway). For example, in Figure 4-11, PC1, PC2, PC4, and Server have to be in the same logical subnet because they are all connected to ports in VLAN 100. PC3 and PC5 have to be in the same subnet (but different from the other devices) because they are connected to ports in VLAN 200. If this is not done, the VLAN to switchport assignments would be incorrect, and the switch would not be able to forward the frames successfully between the devices within the same VLAN. Example 4-24 displays the output of **show vlan brief**, which identifies the VLANs ports are assigned to. By default, all ports are assigned to VLAN 1. Gig0/1 and Gig0/3 have been statically assigned to VLAN 100, and Gig0/4 has been statically assigned to VLAN 200.

Example 4-24 *Verifying Switchport Assignment*

```
SW1#show vlan brief
VLAN Name                Status    Ports
-----
1    default                active    Gi0/5, Gi0/6, Gi0/7, Gi0/8,
                                           Gi0/9, Gi0/10, Gi0/11, Gi0/12,
                                           Gi0/13, Gi0/14, Gi0/15, Gi0/16,
                                           Gi0/17, Gi0/18, Gi0/19, Gi0/20,
                                           Gi0/21, Gi0/22, Gi0/23, Gi0/24,
                                           Te1/0/1, Te1/0/2
99   NATIVE                  active
100  10.1.100.0/24            active    Gi0/1, Gi0/3
200  10.1.200.0/24            active    Gi0/4
1002 fddi-default            act/unsup
1003 trcrf-default          act/unsup
1004 fddinet-default         act/unsup
1005 trbrf-default          act/unsup
```

It is important to note that ports that belong to VLANs that do not exist will not be displayed in the output of **show vlan brief**. As Example 4-23 displayed, they will appear as (Inactive) in the output of **show interfaces switchport**. In addition, trunk ports will not appear in the output of **show vlan brief**. Notice in Example 4-24 that Gig0/2 is missing because it is a trunk port and does not belong to any single VLAN. It is passing traffic for multiple VLANs.

The MAC Address Table

The MAC address table is the most important table for the switch. The MAC address table is the structure that is used by the switch to make a forwarding decision. If the MAC address table is not being populated the way you expect it, you will need to figure out why. This section covers the MAC address table and its importance, using Figure 4-11 as the reference topology.



Example 4-25 displays the dynamically learned MAC addresses on SW1 with the command **show mac address-table dynamic**. The structure of the table is important. It lists the VLANs, the dynamically learned MAC addresses, and the ports. This information is extremely valuable. As discussed earlier, it is populated based on the source MAC address of the frame when it arrives on a switchport. Therefore, when SW1 received a frame inbound on Gigabit Ethernet 0/1 from PC1, it learned the MAC from the frame and associated it with the port it arrived on and the VLAN the port is a member of.

Example 4-25 *SW1's MAC Address Table*

```
SW1#show mac address-table dynamic
Mac Address Table
-----
```


Vlan	Mac Address	Type	Ports
----	-----	-----	-----
100	aaaa.aaaa.aaaa	DYNAMIC	Gi0/1
100	bbbb.bbbb.bbbb	DYNAMIC	Gi0/2
100	cccc.cccc.cccc	DYNAMIC	Gi0/3
100	dddd.dddd.dddd	DYNAMIC	Gi0/2
200	3333.3333.3333	DYNAMIC	Gi0/4
200	5555.5555.5555	DYNAMIC	Gi0/2
Total Mac Addresses for this criterion: 6			

Let's look at an example. What can we conclude by looking at the MAC address table for SW1 displayed in Example 4-26 when comparing it to Figure 4-11?

Example 4-26 *Example of SW1's MAC Address Table*

```
SW1#show mac address-table dynamic
```

Mac Address Table

Vlan	Mac Address	Type	Ports
----	-----	-----	-----
100	bbbb.bbbb.bbbb	DYNAMIC	Gi0/2
100	cccc.cccc.cccc	DYNAMIC	Gi0/3
100	dddd.dddd.dddd	DYNAMIC	Gi0/2
200	3333.3333.3333	DYNAMIC	Gi0/4
200	5555.5555.5555	DYNAMIC	Gi0/2
200	aaaa.aaaa.aaaa	DYNAMIC	Gi0/1
Total Mac Addresses for this criterion: 6			

When comparing Figure 4-11 with Example 4-26, we can conclude that interface Gigabit Ethernet 0/1 is not a member of the correct VLAN. The MAC address table shows the MAC address of PC1 (AAAA.AAAA.AAAA) was learned on the correct interface, but the VLAN number is 200 instead of 100. Reviewing the output of **show vlan brief** and **show interfaces gigabitethernet 0/1 switchport**, as demonstrated in Example 4-27, confirms this for us. Our next step is to reassign the port to the correct VLAN.

Example 4-27 *Confirming SW1's VLAN Assignments*

```
SW1#show vlan brief
```

VLAN	Name	Status	Ports
----	-----	-----	-----
1	default	active	Gi0/5, Gi0/6, Gi0/7, Gi0/8, Gi0/9, Gi0/10, Gi0/11, Gi0/12, Gi0/13, Gi0/14, Gi0/15, Gi0/16, Gi0/17, Gi0/18, Gi0/19, Gi0/20, Gi0/21, Gi0/22, Gi0/23, Gi0/24, Tel0/0/1, Tel0/0/2
99	NATIVE	active	

```

100 10.1.100.0/24          active   Gi0/3
200 10.1.200.0/24          active   Gi0/1, Gi0/4
1002 fddi-default          act/unsup
1003 trcrf-default          act/unsup
1004 fddinet-default        act/unsup
1005 trbrf-default          act/unsup

SW1#show interfaces gigabitethernet 0/1 switchport
Name: Gi0/1
Switchport: Enabled
Administrative Mode: static access
Operational Mode: static access
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: native
Negotiation of Trunking: Off
Access Mode VLAN: 200 (10.1.200.0/24)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: enabled
Voice VLAN: none
...output omitted...

```

While troubleshooting, if you ever need to clear the dynamic entries in the MAC address table immediately so that they can be relearned, giving you the opportunity to confirm the correct associations, issue the **clear mac address-table dynamic EXEC** command.

Layer 2 Trouble Tickets

This section presents various trouble tickets relating to the topics discussed earlier in the chapter. The purpose of these trouble tickets is to give a process that you can follow when troubleshooting in the real world or in an exam environment. All trouble tickets in this section are based on the topology depicted in Figure 4-12.

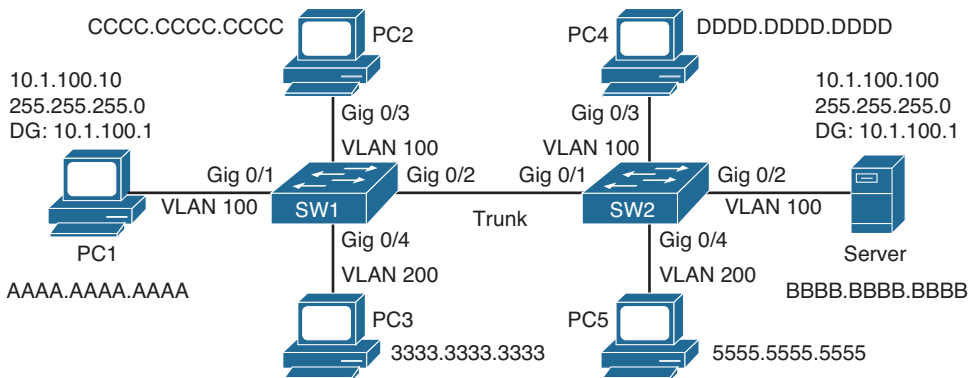


Figure 4-12 Topology for Trouble Tickets

Trouble Ticket 4-1

Problem: A user on PC1 indicates that he is not able to access a document on Server.

This is a typical description within a trouble ticket. Therefore, the first process is to verify the issue. A simple ping from PC1 will help us with this, as shown in Example 4-28.

Example 4-28 Issuing a Ping from PC1

```
PC1>ping 10.1.100.100

Pinging 10.1.100.100 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.1.100.100:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

The output of Example 4-28 indicates that the ping failed. What did we learn from this ping? We learned that we have no connectivity from Layer 1 to Layer 3 of the OSI model. Therefore, we can focus our troubleshooting efforts at these layers. However, let's verify whether others are having the same issue. A ping from PC2 is successful, as shown in Example 4-29. Therefore, it is not a problem with the server or the path from PC2 to the server, which is similar to PC1.

Example 4-29 Issuing a Ping from PC2

```
PC2>ping 10.1.100.100

Pinging 10.1.100.100 with 32 bytes of data:

Reply from 10.1.100.100: bytes=32 time 1ms TTL=128
Reply from 10.1.100.100: bytes=32 time 1ms TTL=128
Reply from 10.1.100.100: bytes=32 time 1ms TTL=128
Reply from 10.1.100.100: bytes=32 time 1ms TTL=128

Ping statistics for 10.1.100.100:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Let's start by checking the IP address of PC1. Using the `ipconfig` command, as shown in Example 4-30, indicates that the IP address, subnet mask, and default gateway are 10.1.100.10, 255.255.255.0, and 10.1.100.1. According to Figure 4-11, these are correct.

Example 4-30 *Verifying PC1's Layer 3 Settings*

```

PC1>ipconfig
Windows IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . : 
    IP Address. . . . .               : 10.1.100.10
    Subnet Mask . . . . .             : 255.255.255.0
    Default Gateway . . . . .         : 10.1.100.1

```

The next step is to check the MAC address table on SW1 using the command **show mac address-table dynamic**. Example 4-31 shows that the MAC address of PC1 was learned on Gigabit Ethernet 0/1, which is correct, but it is associated with VLAN 1 instead of VLAN 100. It appears we have found the problem. However, let's confirm this further with the **show vlan brief** command, as shown in Example 4-32.

Example 4-31 *Verifying PC1 in the MAC Address Table on SW1*

```

SW1#show mac address-table dynamic
    Mac Address Table

-----

Vlan    Mac Address      Type        Ports
-----  -
1       aaaa.aaaa.aaaa   DYNAMIC     Gi0/1
100     bbbb.bbbb.bbbb   DYNAMIC     Gi0/2
100     cccc.cccc.cccc   DYNAMIC     Gi0/3
100     dddd.dddd.dddd   DYNAMIC     Gi0/2
200     3333.3333.3333   DYNAMIC     Gi0/4
200     5555.5555.5555   DYNAMIC     Gi0/2
Total Mac Addresses for this criterion: 6

```

Example 4-32 *Verifying VLAN Port Assignments with the show vlan brief Command*

```

SW1#show vlan brief
VLAN Name                Status    Ports
-----  -
1    default                active    Gi0/1, Gi0/5, Gi0/6, Gi0/7,
                                           Gi0/8, Gi0/9, Gi0/10, Gi0/11,
                                           Gi0/12, Gi0/13, Gi0/14, Gi0/15,
                                           Gi0/16, Gi0/17, Gi0/18, Gi0/19,
                                           Gi0/20, Gi0/21, Gi0/22, Gi0/23,
                                           Gi0/24, Te1/0/1, Te1/0/2
99    NATIVE                 active
100   10.1.100.0/24           active    Gi0/3

```

200	10.1.200.0/24	active	Gi0/4
1002	fddi-default	act/unsup	
1003	trcrf-default	act/unsup	
1004	fddinet-default	act/unsup	
1005	trbrf-default	act/unsup	

To solve the problem, we change the switchport VLAN assignment with the **switchport access vlan 100** interface command and verify that the problem is solved by pinging from PC1 again. Example 4-33 confirms that the problem is solved.

Example 4-33 *Confirming That the Problem Is Solved with a Successful Ping*

```
PC1>ping 10.1.100.100

Pinging 10.1.100.100 with 32 bytes of data:

Reply from 10.1.100.100: bytes=32 time 1ms TTL=128
Reply from 10.1.100.100: bytes=32 time 1ms TTL=128
Reply from 10.1.100.100: bytes=32 time 1ms TTL=128
Reply from 10.1.100.100: bytes=32 time 1ms TTL=128

Ping statistics for 10.1.100.100:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Trouble Ticket 4-2

Problem: A user on PC2 indicates that she is not able to access a document on Server.

As before, the first process is to verify the issue. A simple ping from PC2 will help us with this, as shown in Example 4-34.

Example 4-34 *Issuing a Ping from PC2*

```
PC2>ping 10.1.100.100

Pinging 10.1.100.100 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.1.100.100:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

The output of Example 4-34 indicates that the ping failed. What did we learn from this ping? We learned that we have no connectivity from Layer 1 to Layer 3 of the OSI model. Therefore, we can focus our troubleshooting efforts at these layers. However, let's verify whether others are having the same issue. A ping from PC1 fails, as shown in Example 4-35.

Example 4-35 *Issuing a Ping from PC1*

```
PC1>ping 10.1.100.100

Pinging 10.1.100.100 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.1.100.100:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Therefore, this is not an isolated issue, and we should be looking for causes that would affect multiple users. First thing that comes to mind is a missing VLAN on SW1. PC1 and PC2 are both members of VLAN 100. Using the command **show vlan brief** on SW1 will verify whether the VLAN exists and which switchports are associated with it. As you can see from Example 4-36, VLAN 100 exists, and both switchports for PC1 and PC2 are associated with it.

Example 4-36 *Verifying That VLAN 100 Exists on SW1 with show vlan brief*

```
SW1#show vlan brief
```

VLAN	Name	Status	Ports
1	default	active	Gi0/5, Gi0/6, Gi0/7, Gi0/8, Gi0/9, Gi0/10, Gi0/11, Gi0/12, Gi0/13, Gi0/14, Gi0/15, Gi0/16, Gi0/17, Gi0/18, Gi0/19, Gi0/20, Gi0/21, Gi0/22, Gi0/23, Gi0/24, Te1/0/1, Te1/0/2
99	NATIVE	active	
100	10.1.100.0/24	active	Gi0/1, Gi0/3
200	10.1.200.0/24	active	Gi0/4
1002	fddi-default	act/unsup	
1003	trcrf-default	act/unsup	
1004	fddinet-default	act/unsup	
1005	trbrf-default	act/unsup	

However, this is not enough evidence to shift our focus just yet. The most important information comes from the MAC address table. This will truly verify that the MAC

addresses of PC1 and PC2 are being learned on the correct interfaces and are being associated with the correct VLAN. Example 4-37 displays the output of the **show mac address-table dynamic** command and confirms for us that the MAC addresses are learned correctly and that the ports are associated with the correct VLANs.

Example 4-37 *Verifying the MAC Address in the MAC Address Table*

```
SW1#show mac address-table dynamic
      Mac Address Table
-----
Vlan    Mac Address      Type        Ports
----    -
100     aaaa.aaaa.aaaa   DYNAMIC     Gi0/1
100     cccc.cccc.cccc   DYNAMIC     Gi0/3
200     3333.3333.3333   DYNAMIC     Gi0/4
200     5555.5555.5555   DYNAMIC     Gi0/2
Total Mac Addresses for this criterion: 4
```

However, look very closely at the MAC address table in Example 4-37. What is missing? Do you see any reference to the MAC address of Server? The MAC address of Server is not being learned on Gigabit Ethernet 0/2 of SW1. As a matter of fact, neither is PC4. However, PC5 is being learned. This is a good indication that traffic for VLAN 100 is not being allowed over the trunk. Let's verify this on SW1 with the command **show interfaces trunk**, as shown in Example 4-38. This output shows that VLAN 100 and 200 are allowed on the trunk between SW1 and SW2.

Example 4-38 *Verifying Allowed VLANs on SW1 Trunks*

```
SW1#show interfaces trunk
Port      Mode           Encapsulation  Status        Native vlan
Gi0/2     desirable      n-802.1q       trunking      99

Port      Vlans allowed on trunk
Gi0/2     100,200

Port      Vlans allowed and active in management domain
Gi0/2     100,200

Port      Vlans in spanning tree forwarding state and not pruned
Gi0/2     100,200
```

Let's check the output of **show interfaces trunk** on SW2. As shown in Example 4-39, VLAN 200 is the only VLAN allowed on the trunk link. A further examination of the running configuration, as shown in Example 4-40, indicates that only VLAN 200 is allowed on the trunk.

Example 4-39 *Verifying Allowed VLANs on SW2 Trunks*

```
SW2#show interfaces trunk
```

Port	Mode	Encapsulation	Status	Native vlan
Gi0/1	desirable	n-802.1q	trunking	99


```
Port Vlan allowed on trunk
```

Port	Vlan allowed on trunk
Gi0/1	200


```
Port Vlan allowed and active in management domain
```

Port	Vlan allowed and active in management domain
Gi0/2	200


```
Port Vlan in spanning tree forwarding state and not pruned
```

Port	Vlan in spanning tree forwarding state and not pruned
Gi0/2	200

Example 4-40 *Verifying Interface Configuration in the Running Configuration*

```
SW2#show run interface gigabitethernet 0/1
Building configuration...

Current configuration : 167 bytes
!
interface GigabitEthernet0/1
  switchport trunk native vlan 99
  switchport trunk allowed vlan 200
  switchport mode dynamic desirable
end
```

After issuing the interface command **switchport trunk allowed VLAN 100,200** on SW2 to allow both VLAN 100 and 200, you ping from PC1 and PC2 again to verify that the issue is solved. The ping is successful from PC1 and PC2, as illustrated in Example 4-41.

Example 4-41 *Verifying That the Issue Is Solved*

```
PC1>ping 10.1.100.100
Pinging 10.1.100.100 with 32 bytes of data:

Reply from 10.1.100.100: bytes=32 time 1ms TTL=128
Reply from 10.1.100.100: bytes=32 time 1ms TTL=128
Reply from 10.1.100.100: bytes=32 time 1ms TTL=128
Reply from 10.1.100.100: bytes=32 time 1ms TTL=128

Ping statistics for 10.1.100.100:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

PC2>ping 10.1.100.100
```



```
Pinging 10.1.100.100 with 32 bytes of data:

Reply from 10.1.100.100: bytes=32 time 1ms TTL=128
Reply from 10.1.100.100: bytes=32 time 1ms TTL=128
Reply from 10.1.100.100: bytes=32 time 1ms TTL=128
Reply from 10.1.100.100: bytes=32 time 1ms TTL=128

Ping statistics for 10.1.100.100:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Exam Preparation Tasks

As mentioned in the section “How to Use This Book” in the Introduction, you have a couple of choices for exam preparation: the exercises here; Chapter 22, “Final Preparation;” and the exam simulation questions on the CD-ROM.

Review All Key Topics

Review the most important topics in this chapter, noted with the Key Topic icon in the outer margin of the page. Table 4-2 lists a reference of these key topics and the page numbers on which each is found.



Table 4-3 *Key Topics for Chapter 4*

Key Topic Element	Description	Page Number
Paragraph	A review of the frame-forwarding process	132
List	Outlines potential issues that arise with a Layer 2 topology	140
Example 4-2	Output of show interfaces switchport command on SW1 to verify encapsulation	141
Example 4-3	Output of show interfaces switchport command on SW2 to verify encapsulation	141
Example 4-6	Verifying trunking administrative mode (access)	143
Example 4-7	Verifying trunking administrative mode (trunk)	143
Example 4-12	Result of native VLAN mismatch on trunk	146
Section	Allowed VLANs	147
Example 4-15	Verifying the VTP domain name on SW1	148
Example 4-16	Verifying the VTP domain name on SW2	149
Example 4-22	Verifying VLANs on a switch	153
Section	Incorrect port assignment	154
Paragraph	Using the MAC address table during troubleshooting	155

Define Key Terms

Define the following key terms from this chapter and check your answers in the glossary:

frame, MAC address table, source MAC, destination MAC, encapsulation, 802.1Q, ISL, trunk, access port, dynamic desirable, dynamic auto, native VLAN, VTP, VTP domain name, VLAN

Complete Tables and Lists from Memory

Print a copy of Appendix C, “Memory Tables,” (found on the disc), or at least the section for this chapter, and complete the tables and lists from memory. Appendix D, “Memory Tables Answer Key,” also on the disc, includes completed tables and lists to check your work.

Command Reference to Check Your Memory

This section includes the most important EXEC **show** commands covered in this chapter. It might not be necessary to memorize the complete syntax of every command, but you should be able to remember the basic keywords that are needed.

To test your memory of the commands, cover the right side of Table 4-4 with a piece of paper, read the description on the left side, and then see how much of the command you can remember.

The 300-135 TSHOOT exam focuses on practical, hands-on skills that are used by a networking professional. Therefore, you should be able to identify the commands needed to successfully troubleshoot switches.

Table 4-4 EXEC CLI *show* Commands

Task	Command Syntax
Displays the contents of the MAC address table, including the MAC address associated with a port and the VLAN the port is a member of. Without the dynamic keyword, both static and dynamic entries are displayed. With the dynamic keyword, only dynamically learned entries are displayed.	show mac address-table [dynamic]
Clears dynamically learned MAC addresses from the MAC address table of a switch; this can allow a troubleshooter to determine whether a previously learned MAC address is relearned. Note that on some versions of Cisco IOS running on Cisco Catalyst switches, the clear mac address-table command contains a hyphen between mac and address (that is, clear mac-address-table).	clear mac address-table dynamic
Shows to which VLANs the ports of a switch belong.	show vlan brief
Displays which VLANs are permitted on the trunk ports of a switch and which switchports are configured as trunks.	show interfaces trunk

Task	Command Syntax
Displays VLAN and trunk information related to a switchport. You can verify the operational mode (access or trunk), in addition to the encapsulation (802.1Q or ISL). You can also verify the access VLAN the port will be a member of if it is an access port, in addition to the native VLAN if it is a trunk port.	show interfaces <i>interface_type</i> <i>interface_number</i> switchport
Displays the VTP domain name, configuration revision number, version, mode, and MD5 hash.	show vtp status
Displays the configured VTP password.	show vtp password



This chapter covers the following topics:

- **Spanning-Tree Protocol Overview:** This section reviews how STP determines the STP topology from root bridge election to which ports will be nondesignated.
- **Collecting Information About an STP Topology:** This section identifies the **show** commands required to successfully troubleshoot STP issues.
- **STP Troubleshooting Issues:** This section focuses on what could happen if STP is not behaving as expected.
- **Troubleshooting STP Features:** This section reviews STP features such as PortFast, BPDU Guard, Root Guard, and BPDU Filter. It also identifies the **show** commands that can help during the troubleshooting process.
- **STP Trouble Tickets:** This section provides trouble tickets that demonstrate how a structured troubleshooting process can be used to solve a reported problem.
- **Troubleshooting Layer 2 EtherChannel:** This section reviews how Layer 2 EtherChannels are formed and identifies issues that could cause them to fail.
- **EtherChannel Trouble Tickets:** This section provides trouble tickets that demonstrate how a structured troubleshooting process can be used to solve a reported problem.

Troubleshooting STP and Layer 2 EtherChannel

Maintaining high availability for today's enterprise networks is a requirement for many applications, such as voice and e-commerce, which can impact a business's bottom line if these applications are unavailable for even a short period. To improve availability, many enterprise networks interconnect Layer 2 switches with redundant connections, allowing a single switch or a single link to fail while still maintaining connectivity between any two network endpoints. Such a redundant topology, however, can result in Layer 2 loops, which can cause frames to endlessly circle a LAN (for example, broadcast frames creating a broadcast storm). Therefore, Spanning Tree Protocol (STP) is used to logically break these Layer 2 topological loops by strategically blocking ports, while being able to detect a link failure and bring up a previously blocked switchport to restore connectivity. This chapter reviews the operation of STP and focuses on troubleshooting STP issues.

In addition, this chapter reviews how you can combine multiple physical Layer 2 switchports into a logical EtherChannel bundle. This increases the total bandwidth available on uplinks and tricks STP into thinking there is only one port between the switches instead of multiple ports. As a result, all links are used for traffic forwarding instead of STP blocking them.

“Do I Know This Already?” Quiz

The “Do I Know This Already?” quiz allows you to assess whether you should read this entire chapter thoroughly or jump to the “Exam Preparation Tasks” section. If you are in doubt about your answers to these questions or your own assessment of your knowledge of the topics, read the entire chapter. Table 5-1 lists the major headings in this chapter and their corresponding “Do I Know This Already?” quiz questions. You can find the answers in Appendix A, “Answers to the ‘Do I Know This Already?’ Quizzes.”

Table 5-1 “Do I Know This Already?” Section-to-Question Mapping

Foundation Topics Section	Questions
Spanning-Tree Protocol Overview	1–4
Collecting Information About an STP Topology	5
STP Troubleshooting Issues	6
Troubleshooting STP Features	7
Troubleshooting Layer 2 EtherChannel	8–10

Caution The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you do not know the answer to a question or are only partially sure of the answer, you should mark that question as wrong for purposes of the self-assessment. Giving yourself credit for an answer that you correctly guess skews your self-assessment results and might provide you with a false sense of security.

1. What determines the switch that will be the STP root bridge for a VLAN?
 - a. Lowest priority
 - b. Lowest MAC address
 - c. Lowest bridge ID
 - d. Lowest cost
2. What is the STP port type for all ports on a root bridge?
 - a. Designated port
 - b. Root port
 - c. Nondesignated port
 - d. Nonroot port
3. When determining the root port of a nonroot bridge, if cost is tied, what is referenced next to break the tie?
 - a. Downstream bridge ID
 - b. Upstream bridge ID
 - c. Downstream port ID
 - d. Upstream port ID
4. What is the maximum age for an STP BPDU in seconds?
 - a. 2
 - b. 15
 - c. 20
 - d. 50
5. Which two of the following commands are most helpful in determining STP information for a Layer 2 switch?
 - a. `show spanning-tree vlan`
 - b. `debug spanning-tree state`
 - c. `show spanning-tree interface`
 - d. `show port span`

6. What are two common issues that could result from an STP failure?
 - a. Tagged frames being sent into a native VLAN
 - b. Broadcast storms
 - c. MAC address table filling to capacity
 - d. MAC address table corruption
7. Which STP feature ensures that certain ports in the STP topology never become root ports, and if the port receives a superior BPDU it places it in the root inconsistent state?
 - a. BPDU Guard
 - b. BPDU Filter
 - c. Root Guard
 - d. PortFast
8. Which switch feature allows multiple physical links to be bonded into a logical link?
 - a. STP
 - b. EtherChannel
 - c. PortFast
 - d. Switch virtual interfaces
9. What must match on physical switchports to successfully form an EtherChannel bundle? (Choose three.)
 - a. Interface speed
 - b. Interface mode (access/trunk)
 - c. Native VLAN
 - d. STP port cost
10. What combination will successfully form a Cisco proprietary Layer 2 EtherChannel bundle?
 - a. Active – Passive
 - b. On – Active
 - c. Desirable – Auto
 - d. Desirable – Passive

Foundation Topics

Spanning Tree Protocol Overview

Network availability at Layer 2 of the OSI model requires redundant links between the switches in your topology as well as redundant paths through the network. However, this creates a problem known as a *Layer 2 loop*, as shown in Figure 5-1. Notice how traffic from SW1 can be sent on both links to SW2 and vice versa. Therefore, traffic sent from SW1 on one link to SW2 can go back to SW1 on the other link and continue indefinitely because there is no mechanism built in to a Layer 2 frame that will stop the frame from looping forever through the network, as shown with Loop1 in Figure 5-1. In addition, notice how there is a larger loop between SW1, SW3, and SW2 (Loop 2). Therefore, frames sent out any of the interfaces interconnecting these switches could loop indefinitely through the network as well. This is different from Layer 3 packets that have a time-to-live (TTL) field that will terminate the packet if it does not reach its destination within a finite number of router hops. Therefore, Layer 2 loops need to be prevented by a protocol known as *Spanning Tree Protocol* (STP). IEEE 802.1D STP allows a network to physically have Layer 2 loops while strategically blocking data from flowing over one or more switchports to prevent the looping of traffic.

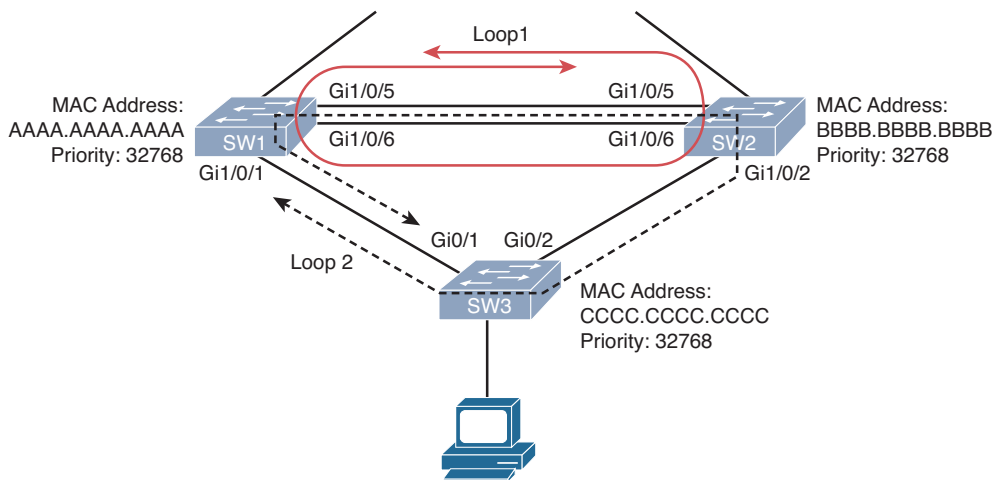


Figure 5-1 Layer 2 Loops

You need to have a solid understanding of how STP makes decisions when troubleshooting Layer 2 issues. Therefore, this section reviews how an STP topology is dynamically formed. In addition, this section discusses commands useful in troubleshooting STP issues.

Reviewing STP Operation

STP uses Bridge Protocol Data Units (BPDUs) to build the STP topology. BPDU packets contain information on ports, addresses, priorities, and costs needed to build the STP topology and ensure that the data ends up where it was intended to go. BPDU messages are exchanged every 2 seconds by default across switches to detect loops in a network topology. The loops are then removed by logically blocking selected bridge interfaces and placing them in the blocked state.

STP prevents Layer 2 loops from occurring in a network, because such an occurrence could result in a broadcast storm or the corruption of a switch's MAC address table. Switches in an STP topology are classified as one of the following:

Key Topic

- **Root bridge:** The root bridge is a switch elected to act as a reference point for a spanning tree topology. The switch with the lowest bridge ID (BID) is elected as the root bridge. The BID is made up of a priority value (default is 32768) and a MAC address (base Ethernet MAC of switch as shown in the output of the **show version** command.). The priority is used first; only if the priority is tied between two or more switches will the MAC address be used to break the tie.
- **Nonroot bridge:** All other switches in the STP topology are considered nonroot bridges.

Figure 5-2 illustrates the root bridge election in a network. Notice that because all bridge priorities are 32768 (default), the switch with the lowest MAC address (that is, SW1) is elected as the root bridge. The MAC address is read left to right. Because a MAC address is based on hexadecimal, lower to higher is 0–9, then A–F.

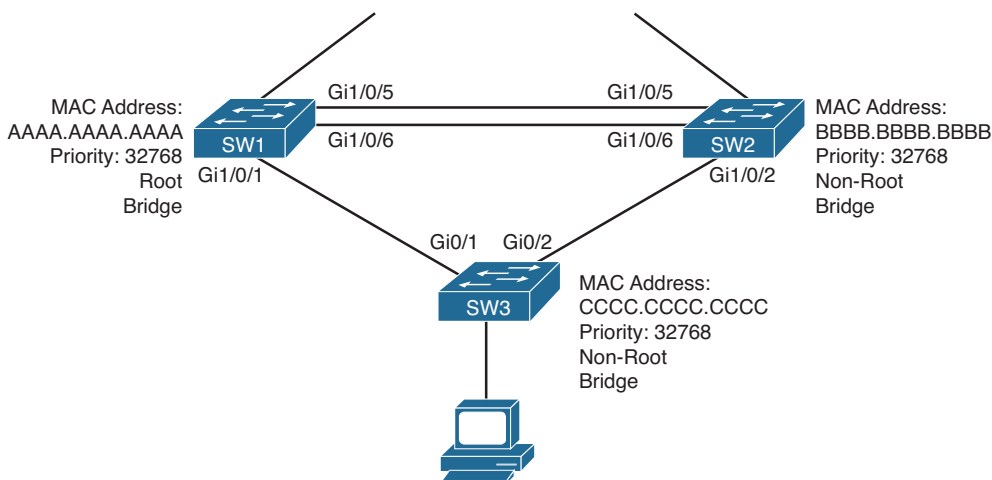


Figure 5-2 Root Bridge Election

Key Topic

Remember the golden rule of STP: Lower is better and ties are not acceptable.

Note Remembering this rule will help you during each step of the election processes.

Switchports in an STP topology are categorized as one of the following port roles described in Table 5-2 and illustrated in Figure 5-3.



Table 5-2 STP Port Roles

Port Roles	Description
Root port (RP)	Every nonroot bridge has a single root port (this is mandatory). It is the port on the switch that is closest to the root bridge, in terms of cost, which is inversely proportional to bandwidth by default. If cost is tied, the upstream BID is used to break the tie. If the upstream BID is tied, the upstream port ID (PID) is used to break the tie.
Designated port (DP)	Every network segment has a single designated port (this is mandatory). It is the port on the segment that is closest to the root bridge, in terms of cost. If cost is tied, the upstream BID is used to break the tie. If the upstream BID is tied, the upstream port ID (PID) is used to break the tie.

Note Because all ports on the root bridge are as close as you could get to the root bridge, all ports on a root bridge are DPs.

Nondesignated port (X) These are the ports blocking traffic to create a loop-free topology.

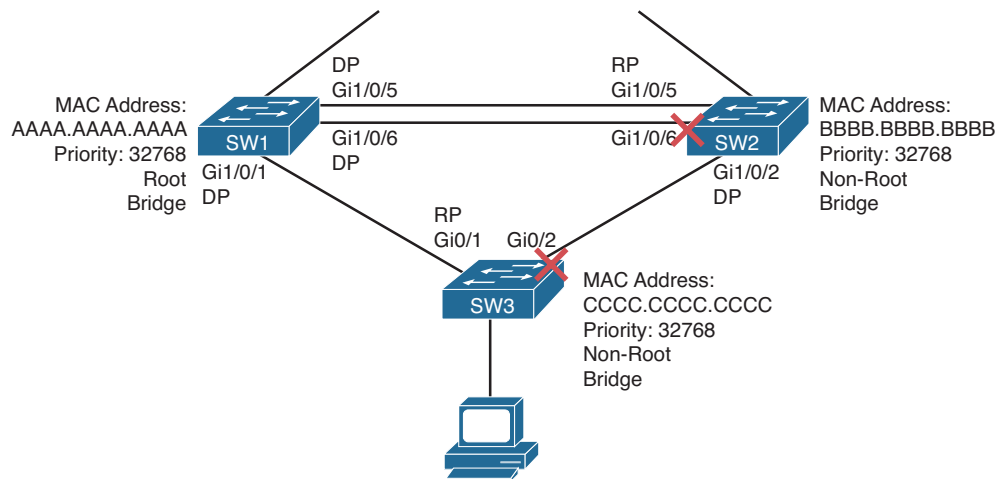


Figure 5-3 STP Port Roles

Table 5-3 shows the default port costs for various link speeds for both 802.1D STP and its successor 802.1D-2004 STP. Notice the higher the speed the lower the cost. Remember that a lower cost is better and that the cost used is the cumulative path cost.

Key Topic

Table 5-3 *Default Port Costs*

Link Speed	802.1D STP Port Cost	802.1D-2004 STP Port Cost
10 Mbps (Ethernet)	100	2000000
100 Mbps (Fast Ethernet)	19	200000
1 Gbps (Gigabit Ethernet)	4	20000
10 Gbps (Ten Gig Ethernet)	2	2000
100 Gbps	N/A	200
1 Tbps	N/A	20
10 Tbps	N/A	2

Key Topic

Determining Root Port

Being able to determine why a port has a specific role is important for troubleshooting and tuning the STP topology. Notice the root port for switch SW2 is Gig 1/0/5 in Figure 5-3. Why was it chosen as the root port? If you are not sure, review the following steps for determining the root port on a switch:

1. Identify the port that has the lowest cumulative cost path to the root bridge. In Figure 5-3, the total cost from SW2 Gi1/0/5 to the root bridge is 4. The total cost from SW2 Gi1/0/6 to the root bridge is 4. The total cost from SW2 Gi1/0/2 to the root bridge is (4 + 4) 8. Remember, lower is better and ties are not acceptable. In this case, we have a tie for the lowest value at 4. When the path cost is tied, you use the lowest upstream BID as a tiebreaker. Proceed to Step 2.
2. Identify the SW2 port (Gi1/0/5 or Gi1/0/6) that receives a BPDU with a lower upstream BID. In this case, the BID received in the BPDUs from SW1 is tied. The priority is checked first for the BPDUs received by SW2 on Gi1/0/5 and Gi1/0/6 from SW1. In Figure 5-3, the BPDUs will have the same priority because they are sent from the same switch (SW1) with a priority of 32768. Next is to compare the MAC addresses listed in the BPDUs. Again, they are both the same because switches use the same base Ethernet MAC address for all BPDUs sent on all interfaces. Therefore, both received BPDUs from SW1 have a priority of 32768 and a MAC of AAAA.AAAA.AAAA. When the upstream BID is tied, you use the upstream PID to break the tie. Proceed to Step 3.
3. Identify the port that receives a BPDU with a lower upstream PID. When SW1 sends BPDUs, it includes a PID. This PID includes a port priority number and an interface number. The priority number can be manually changed (default 128); however,

the interface number cannot. It is generated by the switch to identify the port. In Figure 5-3, SW1 would more than likely have a PID of 128.5 on Gi1/0/5 and 128.6 on Gi1/0/6 by default. As a result, when SW2 receives the BPDUs from SW1, the received BPDU on Gi1/0/5 has a PID attached of 128.5 and the received BPDU on Gi1/0/6 has a PID attached of 128.6. Lower is better; therefore, SW2 Gi1/0/5 is elected the root port based on the PID value sent from SW1 in the BPDUs.

Focusing on SW3 in Figure 5-3 shows a total cost of 4 to get to the root bridge using Gi0/1 and a total cost of 8 using Gi0/2. Therefore, Gi0/1 is elected as the root port.



Determining Designated Port

When determining the designated ports for each segment, you follow the same steps listed in the previous section for the root port election. Remember that every port on the root bridge will be a designated port. Therefore, without performing any calculations, you already know a few designated ports in the topology. As a result, in Figure 5-3 the only link/segment remaining without a designated port is the segment between SW2 and SW3. We can see that it is already labeled as Gi1/0/2 on SW2, but why? Let's walk through the steps together:

1. Identify the port on the segment with the lowest cumulative cost back to the root bridge. SW2 Gi1/0/2 has a cumulative cost (including the cost of the segment itself) of $(4 + 4) = 8$. SW3 Gi0/2 has a cumulative cost (including the cost of the segment itself) of $(4 + 4) = 8$. We have a tie, so we move on to Step 2.
2. Find the upstream switch with the lowest BID. This is tricky if you do not know where to position yourself. Here is my trick. Pretend you are standing in the middle of the segment between SW2 and SW3. Point to SW2. What is the priority? 32768. Point to SW3. What is the priority? 32768. We have a tie. We then need to look at the MAC address. Still standing in the middle of the segment, point to SW2. What is the MAC address? BBBB.BBBB.BBBB. Point to SW3. What is the MAC address? CCCC.CCCC.CCCC. Which one is lower? It is the MAC address of SW2.

Therefore, SW2's port Gi1/0/2 is the designated port for the segment between SW2 and SW3.

Determining Nondesignated Port

Every other port that is not a root port or a designated port is a nondesignated port and will be blocking traffic, as depicted in Figure 5-3. Nondesignated ports do not forward traffic during normal operation but do receive BPDUs to determine the state of the STP topology. If a link in the topology goes down, the nondesignated port indirectly detects the link failure from BPDUs and determines whether it needs to transition to the forwarding state or not to ensure network availability while preventing loops.

If a nondesignated port does need to transition to the forwarding state, the type of STP in use will determine how long it takes to transition to the forwarding state. STP (802.1D), Common Spanning Tree (CST), and Cisco's implementation of STP (PVST+) transition through the following states:



- **Blocking:** The port remains in the blocking state until it needs to transition. If it needs to transition, it will wait for 20 seconds by default. This is known as the *max age* time. It is essentially the time-to-live of a BPDU. A BPDU is only valid for 20 seconds. If a new BPDU is not received before the max age time expires, the switch considers the BPDU stale and transitions to the listening state. During the blocking state, a nondesignated port evaluates BPDUs in an attempt to determine its role in the spanning tree.
- **Listening:** The port remains in this state for 15 seconds by default (15 seconds is known as the *forward delay*). During this time, the port sources BPDUs, which inform adjacent switches of the port's intent to forward data. In addition, it receives BPDUs from other switches, which will help in the building of the STP topology and determining the root ports and designated ports.
- **Learning:** The port moves from the listening state to the learning state and remains in this state for 15 seconds by default. During this time, the port begins to add entries to its MAC address table while still sending and receiving BPDUs to ensure that the decisions made in relation to the STP topology are still accurate.
- **Forwarding:** The port moves from the learning state to the forwarding state and begins to forward frames while learning MAC addresses and sending and receiving BPDUs. Root ports and designated ports are in this state.

As you can see, the total time to transition from the blocking state to the forwarding state is 50 seconds with 802.1D.

Rapid Spanning Tree Protocol (802.1w) and Multiple Spanning Tree Protocol (802.1s) use a handshaking mechanism rather than timers as their primary method of convergence. Therefore, convergence is 5 seconds or less. If the handshaking mechanism fails, 802.1w and 802.1s rely on the same timers as 802.1D as backup. In addition, if a neighboring switch is using 802.1D, timers are used with them for backward compatibility.

Collecting Information About an STP Topology

Cisco Catalyst switches will dynamically form a spanning-tree topology using default port costs and bridge priorities right out of the box. You do not have to do anything. However, the resulting STP topology might not be the best for your organization. For example, you might want to influence a particular switch to become a root bridge to ensure optimal traffic forwarding through a Layer 2 topology. Or, you might want traffic for one VLAN to take a certain path while traffic for other VLANs takes a different path. If you ever need to manipulate STP, which will more than likely be the case, you need to know the current topology and how to modify it. This section identifies the various methods we can use to gather information about our STP topology.

Gathering STP Information

When troubleshooting an STP topology, one of the first tasks is to learn which switch is acting as the root bridge, in addition to learning the port roles on the various switches

in the topology. Not only is this information important in understanding how frames are currently flowing through the topology, but comparing the current STP state of a topology to a baseline state can also provide clues as to the underlying cause of an issue, such as suboptimal traffic forwarding.



The **show spanning-tree [vlan {vlan_id}]** command can display information about the STP state of a switch. Consider Example 5-1, which shows the output from the **show spanning-tree vlan 1** command. The VLAN is specified because Cisco Catalyst switches use Per-VLAN Spanning Tree + (PVST+) by default. PVST+ allows a switch to run a separate STP instance for each VLAN. The output in Example 5-1 shows that SW3 is not the root bridge for the spanning tree of VLAN 1. This is because the MAC address of the root bridge (Root ID) differs from the MAC address of SW3 (Bridge ID). In addition, there is a root port on the switch, which a root bridge cannot have, and it does not state that this switch is the root bridge. The Gig 0/1 port of switch SW3 is the root port of the switch, whereas port Gig 0/2 is a nondesignated port. (That is, it is a blocking port.) Note that the port cost of Gig 0/1 is 4, and the port cost of Gig 0/2 is 4 as well.

Example 5-1 show spanning-tree vlan Command Output

```
SW3#show spanning-tree vlan 1
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority    32768
             Address     aaaa.aaaa.aaaa
             Cost        4
             Port        25 (GigabitEthernet0/1)
             Hello Time   2 sec   Max Age 20 sec   Forward Delay 15 sec

  Bridge ID   Priority    32769 (priority 32768 sys-id-ext 1)
             Address     cccc.cccc.cccc
             Hello Time   2 sec   Max Age 20 sec   Forward Delay 15 sec
             Aging Time   300

Interface          Role Sts Cost          Prio.Nbr Type
-----
Gi0/1              Root FWD 4             128.25   P2p
Gi0/2              Altn BLK 4             128.26   P2p
```

The **show spanning-tree interface interface_type interface_number detail** command, as shown in Example 5-2, displays the number of BPDUs sent and received, the port identifier, and the designated root and designated bridge priority and MAC address. Note that in a stable topology, root ports should only receive BPDUs, and designated ports should only send BPDUs. Therefore, if you see a high number of sent and received BPDUs on ports, you have an unstable STP topology and need to determine why this is so and fix it.

Example 5-2 `show spanning-tree interface interface_type interface_number detail`
Command Output

```
SW3#show spanning-tree interface gig 0/1 detail
Port 25 (GigabitEthernet0/1) of VLAN0001 is root forwarding
  Port path cost 4, Port priority 128, Port Identifier 128.25.
  Designated root has priority 32768, address aaaa.aaaa.aaaa
  Designated bridge has priority 32768, address aaaa.aaaa.aaaa
  Designated port id is 128.1, designated path cost 0
  Timers: message age 2, forward delay 0, hold 0
  Number of transitions to forwarding state: 1
  Link type is point-to-point by default
  BPDU: sent 1, received 1245
```

Gathering MSTP Information

Multiple Spanning Tree Protocol (MSTP) allows you to group multiple VLANs into a single STP instance. This significantly improves STP in end-to-end VLAN deployments where a large number of VLANs are maintained by many switches. When you group various VLANs together into the same instance, the CPU does not have to process BPDUs for all the different VLANs. In fact, with MSTP, only MST0 (known as the IST) is used to send BPDUs, and all the other MST instances are listed in the MST0 BPDUs as M-records. This improves CPU performance.

Consider this. If you have 100 VLANs and you only have 2 uplinks from an access layer switch to the distribution layer, you can group half the VLANs in one instance and the other half in another instance. You can then manipulate who the root bridge is so that one instance ends up using one uplink and the other instance uses the other uplink. You have just achieved load sharing and reduced the number of STP instances from 100 to 2, thus conserving CPU resources. To ensure you optimize load sharing, you need to gather statistics about the traffic flowing through the networking on a VLAN-by-VLAN basis and make sure that you do not place heavily used VLANs in the same MSTP instance or you will not achieve optimal load sharing.

When deploying and troubleshooting MSTP, you have to remember these three very important rules for switches in the same region:

- The MSTP region name must match.
- The MSTP revision number must match.
- The MSTP instance to VLAN mappings must be the same on all the switches.

If any of the items listed do not match exactly, the digest that is sent within an MSTP BPDU will be different, and the switches will consider each other to be in a different MSTP region and therefore produce different spanning-tree topologies than the administrator envisioned. To verify the current region name, revision number, and VLAN to instance mappings on a switch, issue the **show spanning-tree mst configuration** command, as shown in Example 5-3.

Example 5-3 show spanning-tree mst configuration *Command Output*

```
SW3#show spanning-tree mst configuration
Name          TSHOOT
Revision      10    Instances configured 2

Instance      Vlans mapped
-----
0             1-9, 11-19, 21-99, 101-199, 201-4094
1             10, 100
2             20, 200
-----
```

STP Troubleshooting Issues

If STP fails to operate correctly, Layer 2 frames can endlessly circulate through a network because of the loop created. This behavior can lead to issues such as MAC address table corruption and broadcast storms. In this section we analyze the results of an STP failure.

Corruption of a Switch's MAC Address Table

Recall from Chapter 4, “Troubleshooting Layer 2 Trunks, VTP, and VLANs,” that the MAC address table determines what a switch will do with a frame. Therefore, this table needs to be accurate. A switch will dynamically learn what MAC addresses are reachable off its ports; however, in the event of an STP failure, the MAC address table of a switch can become corrupt. To illustrate, consider Figure 5-4. PC1 is transmitting traffic to PC2. When the frame sent from PC1 is transmitted on segment A, the frame is seen on the Gig 0/1 ports of switches SW1 and SW2, causing both switches to add an entry to their MAC address tables (AAAA.AAAA.AAAA is associated with port Gig 0/1). Because STP is not functioning, both switches then forward the frame out segment B. As a result, PC2 receives two copies of the frame. Also, switch SW1 sees the frame forwarded out the Gig 0/2 port of switch SW2. Because the frame has a source MAC address of AAAA.AAAA.AAAA, switch SW1 incorrectly updates its MAC address table indicating that a MAC address of AAAA.AAAA.AAAA resides off port Gig 0/2. Similarly, switch SW2 sees the frame forwarded onto segment B by switch SW1 on its Gig 0/2 port. Therefore, switch SW2 also incorrectly updates its MAC address table. As a result of this, all frames destined to AAAA.AAAA.AAAA will be forwarded out Gig0/2 and never reach PC1.

That was a simplified example of what would occur. In reality, as frames continue to propagate through the network, not only would the MAC address table be corrupt, it would be unstable. At one moment AAAA.AAAA.AAAA would be learned on Gig0/1, then Gig0/2, then back on Gig0/1, then Gig0/2.

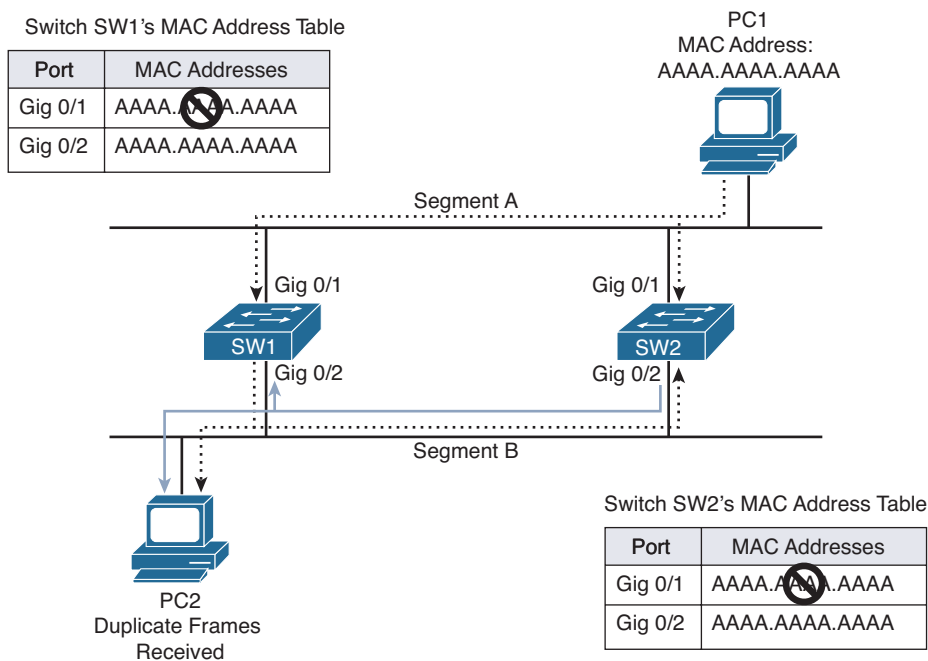


Figure 5-4 MAC Address Table Corruption

You will be able to recognize this issue because syslog messages will be generated identifying that you have MAC addresses flapping between different ports on the same switch. The following syslog messages show that the MAC addresses are being learned on Gi0/1 and Gi0/2, and this would occur only if there were a loop allowing the same frame to be seen on multiple interfaces:

```
%SW_MATM-4-MACFLAP_NOTIF: Host 0000.5e00.0114 in vlan 20 is flapping between port
Gi0/1 and port Gi0/2

%SW_MATM-4-MACFLAP_NOTIF: Host 8049.7111.7e05 in vlan 502 is flapping between port
Gi0/1 and port Gi0/2

%SW_MATM-4-MACFLAP_NOTIF: Host 0050.b60c.f21b in vlan 20 is flapping between port
Gi0/1 and port Gi0/2
```

Broadcast Storms

As previously mentioned, when a switch receives a broadcast frame (that is, a frame destined for a MAC address of FFFF.FFFF.FFFF), the switch floods the frame out all switch-ports except the port on which the frame was received. The same is true for unknown unicast and multicast frames. Because a Layer 2 frame does not have a TTL field, a broadcast frame endlessly circulates through the Layer 2 topology, consuming resources on both switches and attached devices (for example, user PCs).

Figure 5-5 illustrates how a broadcast storm can form in a Layer 2 topology when STP is not functioning correctly.

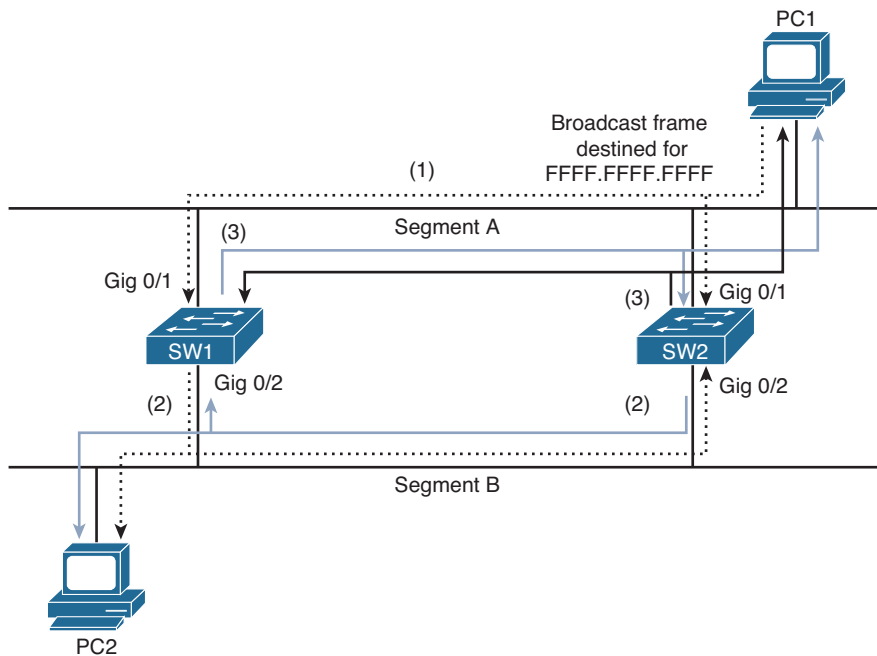


Figure 5-5 Broadcast Storm

1. PC1 sends a broadcast frame onto Segment A, and the frame enters each switch on port Gig 0/1.
2. Both switches flood a copy of the broadcast frame out of their Gig 0/2 ports (that is, on to Segment B), causing PC2 to receive two copies of the broadcast frame.
3. Both switches receive a copy of the broadcast frame on their Gig 0/2 ports (that is, from Segment B) and flood the frame out of their Gig 0/1 ports (that is, onto Segment A), causing PC1 to receive two copies of the broadcast frame.

This behavior continues, as the broadcast frame copies continue to loop through the network. The performance of PC1 and PC2 is impacted, because they also continue to receive copies of the broadcast frame that they must process.

A common complaint you will receive from multiple network users at the same time when there is an STP issue is, *the network/Internet is really slow*. This is because of the broadcast storm consuming the majority of the resources in the Layer 2 network. Therefore, the frames going to the resources that the users need to access are not making it to the destination or are taking a really long time because the network is congested.



Troubleshooting STP Features

STP relies on many features to protect the topology. These features are not enabled by default. Knowing how to troubleshoot these features is important to ensure the STP topology is functioning as it should. This section discusses these features and reviews the commands needed to troubleshoot them.

PortFast

The PortFast feature is used to transition a switchport to the forwarding state as soon as the switchport is enabled. (A device is plugged in, and the switchport is not shut down.) If you are using PortFast with PVST+, RPVST+, or MSTP, when a BPDU is received on a PortFast-enabled switchport, the switchport will immediately transition out of the PortFast state and become a normal switchport. This ensures that it transitions through the necessary states and processes before going to the forwarding state to ensure that a loop is not caused. You can enable PortFast on an interface-by-interface basis with the **spanning-tree portfast** interface command or globally with the **spanning-tree portfast default** command, which will enable it on all nontrunking switchports. Example 5-4 identifies three ways to verify PortFast is enabled on an interface.

Example 5-4 Verifying PortFast-Enabled Interfaces

```
SW3#show run interface fa0/1
Building configuration...

Current configuration : 108 bytes
!
interface FastEthernet0/1
  switchport access vlan 10
  switchport mode access
  spanning-tree portfast
end

SW3#show spanning-tree interface fastEthernet 0/1 portfast
VLAN0010          enabled

SW3#show spanning-tree interface fastEthernet 0/1 detail
Port 1 (FastEthernet0/1) of VLAN0010 is designated forwarding
  Port path cost 19, Port priority 128, Port Identifier 128.1.
  Designated root has priority 10, address 2893.fe3a.e300
  Designated bridge has priority 32778, address 081f.f34e.b800
  Designated port id is 128.1, designated path cost 4
  Timers: message age 0, forward delay 0, hold 0
  Number of transitions to forwarding state: 1
  The port is in the portfast mode
  Link type is point-to-point by default
  BPDU: sent 11, received 0
```

If you enabled PortFast globally, you can use another **show** command to verify that PortFast was enabled globally: **show spanning-tree summary**, as shown in Example 5-5. Notice that PortFast Default is enabled. Also notice how the output of the command **show spanning-tree interface fastEthernet 0/1 detail** in Example 5-5 is different when compared to Example 5-4. In Example 5-5, it states, “The port is in the portfast mode by default,” which indicates that PortFast was enabled globally.

Example 5-5 *Verifying Globally Enabled PortFast Interfaces*

```

SW3#show spanning-tree summary
Switch is in rapid-pvst mode
Root bridge for:
EtherChannel misconfig guard is enabled
Extended system ID          is enabled
Portfast Default             is enabled
PortFast BPDU Guard Default  is disabled
Portfast BPDU Filter Default is disabled
Loopguard Default           is disabled
UplinkFast                   is disabled
BackboneFast                 is disabled
Configured Pathcost method used is short

SW3#show spanning-tree interface fastEthernet 0/1 detail
Port 1 (FastEthernet0/1) of VLAN0010 is designated forwarding
  Port path cost 19, Port priority 128, Port Identifier 128.1.
  Designated root has priority 10, address 2893.fe3a.e300
  Designated bridge has priority 32778, address 081f.f34e.b800
  Designated port id is 128.1, designated path cost 4
  Timers: message age 0, forward delay 0, hold 0
  Number of transitions to forwarding state: 1
  The port is in the portfast mode by default
  Link type is point-to-point by default
  Bpdu filter is enabled by default
  BPDU: sent 11, received 0

```

One of the easiest ways to confirm that a switchport is indeed enabled for PortFast is to review the output of **show spanning-tree**. As shown in Example 5-6, Fa 0/1 is listed as an Edge port indicated that PortFast is enabled on the interface.

Example 5-6 *Using show spanning-tree to Verify PortFast Status*

```

SW3#show spanning-tree
...output omitted...

```

Interface	Role	Sts	Cost	Prio.	Nbr	Type
-----	-----	-----	-----	-----	-----	-----
Fa0/1	Desg	FWD	19	128.1	P2p	Edge
Fa0/2	Desg	FWD	19	128.2	P2p	Edge
Gi0/1	Root	FWD	4	128.25	P2p	
Gi0/2	Altn	BLK	4	128.26	P2p	

```

...output omitted...

```

BPDU Guard

BPDU Guard is used to enforce STP domain borders. This ensures that the STP topology remains predictable. When a BPDU is received on a switchport enabled with BPDU

Guard, the port will be disabled and placed in the err-disabled state. To verify which ports are in the err-disabled state, issue the command **show interfaces status**, as shown in Example 5-7. In this example, Fast Ethernet 0/1 is in the err-disabled state. In addition, if you are tracking syslog messages, you will receive the following:

```
%SPANTREE-2-BLOCK_BPDUGUARD: Received BPDU on port Fa0/1 with BPDU Guard enabled.
Disabling port.
```

```
%PM-4-ERR_DISABLE: bpduguard error detected on Fa0/1, putting Fa0/1 in err-disable
state
```

```
%LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to down
```

Example 5-7 show interfaces status Command Output

```
SW3#show interfaces status
```

Port	Name	Status	Vlan	Duplex	Speed	Type
Fa0/1		err-disabled	10	auto	auto	10/100BaseTX
Fa0/2		connected	10	a-full	a-100	10/100BaseTX
Fa0/3		notconnect	1	auto	auto	10/100BaseTX
Fa0/4		notconnect	1	auto	auto	10/100BaseTX
Fa0/5		notconnect	1	auto	auto	10/100BaseTX
Fa0/6		notconnect	1	auto	auto	10/100BaseTX

Like PortFast, BPDU Guard can be enabled on an interface-by-interface basis with the **spanning-tree bpduguard enable** interface command or globally with the **spanning-tree portfast bpduguard default** global configuration command. The global command will only enable it on PortFast-enabled interfaces.

You can verify whether BPDU Guard is enabled globally using the commands **show spanning-tree summary** and **show spanning-tree interface interface_type interface_number detail**, as depicted in Example 5-8.

Example 5-8 Verifying BPDU Guard Is Enabled Globally

```
SW3#show spanning-tree summary
Switch is in rapid-pvst mode
Root bridge for:
Extended system ID          is enabled
Portfast Default             is disabled
PortFast BPDU Guard Default is enabled
Portfast BPDU Filter Default is disabled
Loopguard Default           is disabled
EtherChannel misconfig guard is enabled
UplinkFast                   is disabled
BackboneFast                 is disabled
Configured Pathcost method used is short
...output omitted...

SW3#show spanning-tree interface fastethernet 0/1 detail
```

```

Port 1 (FastEthernet0/1) of VLAN0010 is designated forwarding
  Port path cost 19, Port priority 128, Port Identifier 128.1.
  Designated root has priority 10, address 2893.fe3a.e300
  Designated bridge has priority 32778, address 081f.f34e.b800
  Designated port id is 128.1, designated path cost 4
  Timers: message age 0, forward delay 0, hold 0
  Number of transitions to forwarding state: 1
  The port is in the portfast mode
  Link type is point-to-point by default
  Bpdu guard is enabled by default
  BPDU: sent 11, received 0

```

You can verify if BPDU Guard has been enabled on an interface basis with the **show spanning-tree interface interface_type interface_number detail** command and the **show run interface interface_type interface_number** command, as shown in Example 5-9.

Example 5-9 Verifying BPDU Guard Is Enabled on an Interface

```

SW3#show spanning-tree interface fastethernet 0/1 detail
Port 1 (FastEthernet0/1) of VLAN0010 is designated forwarding
  Port path cost 19, Port priority 128, Port Identifier 128.1.
  Designated root has priority 10, address 2893.fe3a.e300
  Designated bridge has priority 32778, address 081f.f34e.b800
  Designated port id is 128.1, designated path cost 4
  Timers: message age 0, forward delay 0, hold 0
  Number of transitions to forwarding state: 1
  The port is in the portfast mode
  Link type is point-to-point by default
  Bpdu guard is enabled
  BPDU: sent 4, received 0

SW3#show run interface fastethernet 0/1
Building configuration...

Current configuration : 140 bytes
!
interface FastEthernet0/1
  switchport access vlan 10
  switchport mode access
  spanning-tree portfast
  spanning-tree bpduguard enable
end

```

To recover from the err-disabled state, remove the device that is sending the rogue BPDUs, and then manually disable and enable the err-disabled interface with the **shutdown** and then **no shutdown** commands. Or, you can set up an err-disable recovery feature that will attempt to automatically enable the interface at defined intervals. If the

rogue BPDUs are still detected, the interface will go back into the err-disabled state. If the rogue BPDUs are not detected anymore, the interface will automatically recover. To enable the err-disable recovery feature for BPDU Guard, use the **errdisable recovery cause bpduguard** global configuration command.

BPDU Filter

BPDU Filter is designed to suppress the sending and receiving of BPDUs on an interface. This would be for security reasons. For example, there is no need to send BPDUs out an interface that is connected to an end station or a router. Doing so allows the end station to collect the data in the BPDUs and potentially launch an attack against the STP topology. How you enable it determines the extent of BPDUs that will be suppressed:

- If you enable it globally, with the **spanning-tree portfast bpdupfilter default** command, BPDU Filter will be enabled on all PortFast-enabled interfaces and will suppress the sending of BPDUs out an interface. However, if a BPDU is received on an interface, it will process it normally and, if necessary, transition the interface through the normal STP states/processes.
- If you enable BPDU Filter manually on an interface with the **spanning-tree bpdupfilter enable** command, it suppresses the sending and receiving of BPDUs. This is not recommended because any received BPDUs are ignored and may result in a Layer 2 loop because the interface is automatically in the forwarding state.

You can verify whether BPDU Filter is enabled globally with the **show spanning-tree summary** command and the **show spanning-tree interface interface_type interface_number detail** command, as shown in Example 5-10. If it is enabled on an interface-by-interface basis, which is not recommended, you can verify BPDU Filter with the **show spanning-tree interface interface_type interface_number detail** command and the **show run interface interface_type interface_number** command, as shown in Example 5-11.

Example 5-10 Verifying BPDU Filter Is Enabled Globally

```
SW3#show spanning-tree summary
Switch is in rapid-pvst mode
Root bridge for:
Extended system ID          is enabled
Portfast Default            is disabled
PortFast BPDU Guard Default is disabled
Portfast BPDU Filter Default is enabled
Loopguard Default           is disabled
EtherChannel misconfig guard is enabled
UplinkFast                  is disabled
BackboneFast                is disabled
Configured Pathcost method used is short

SW3#show spanning-tree interface fastethernet 0/1 detail
Port 1 (FastEthernet0/1) of VLAN0010 is designated forwarding
```



```

Port path cost 19, Port priority 128, Port Identifier 128.1.
Designated root has priority 10, address 2893.fe3a.e300
Designated bridge has priority 32778, address 081f.f34e.b800
Designated port id is 128.1, designated path cost 4
Timers: message age 0, forward delay 0, hold 0
Number of transitions to forwarding state: 1
The port is in the portfast mode
Link type is point-to-point by default
Bpdu guard is enabled
Bpdu filter is enabled by default
BPDU: sent 11, received 0

```

Example 5-11 *Verifying BPDU Filter Is Enabled on an Interface*

```

SW3#show spanning-tree interface fastethernet 0/1
Port 1 (FastEthernet0/1) of VLAN0010 is designated forwarding
  Port path cost 19, Port priority 128, Port Identifier 128.1.
  Designated root has priority 10, address 2893.fe3a.e300
  Designated bridge has priority 32778, address 081f.f34e.b800
  Designated port id is 128.1, designated path cost 4
  Timers: message age 0, forward delay 0, hold 0
  Number of transitions to forwarding state: 1
  The port is in the portfast mode
  Link type is point-to-point by default
  Bpdu guard is enabled
  Bpdu filter is enabled
  BPDU: sent 18, received 0

SW3#show run interface fastethernet 0/1
Building configuration...

Current configuration : 173 bytes
!
interface FastEthernet0/1
  switchport access vlan 10
  switchport mode access
  spanning-tree portfast
  spanning-tree bpduguard enable
  spanning-tree bpdufilter enable
end

```

If you are experiencing a Layer 2 loop in your topology, check whether BPDUFilter was enabled on an interface. If so, it would be suppressing the sending and receiving of BPDUs. As a result, a port within the topology is in the forwarding state causing a Layer 2 loop when it should be in the blocking state.

Root Guard

Root Guard is designed to protect the root bridge by ensuring that certain ports on non-root bridges are prevented from becoming root ports. If you recall, the root port on a switch points to the root bridge. If a rogue switch is introduced to the STP topology with a superior BID, it can become the root bridge, and root ports would change on all the other switches so that the new root ports point to the rogue root bridge.

Root Guard stops this from happening by ignoring superior BPDUs that are received on the Root Guard-enabled ports and placing the port in the spanning-tree inconsistent state. Because Root Guard is enabled on an interface-by-interface basis with the command **spanning-tree guard root**, the command **show spanning-tree interface *interface_type* *interface_number* detail**, as shown in Example 5-12, is used to verify its configuration. You can also verify which ports are inconsistent by issuing the **show spanning-tree inconsistentports** command, as shown in Example 5-13. Notice how Fast Ethernet 0/1 is in the root inconsistent state. This is a good indication that the interface is enabled for Root Guard and that it received a superior BPDU.

Example 5-12 Verifying That RootGuard Is Enabled on an Interface

```
SW3#show spanning-tree interface fastethernet 0/1
Port 1 (FastEthernet0/1) of VLAN0010 is designated forwarding
  Port path cost 19, Port priority 128, Port Identifier 128.1.
  Designated root has priority 10, address 2893.fe3a.e300
  Designated bridge has priority 32778, address 081f.f34e.b800
  Designated port id is 128.1, designated path cost 4
  Timers: message age 0, forward delay 0, hold 0
  Number of transitions to forwarding state: 2
  The port is in the portfast mode
  Link type is point-to-point by default
  Bpdu guard is enabled
  Bpdu filter is enabled by default
  Root guard is enabled on the port
  BPDU: sent 18, received 0
```

Example 5-13 Verifying Inconsistent Ports on a Switch

```
SW3#show spanning-tree inconsistent ports
```

Name	Interface	Inconsistency
VLAN0010	FastEthernet0/1	Root Inconsistent

```
Number of inconsistent ports (segments) in the system : 1
```

In addition, when a port goes into the root inconsistent state you will receive a syslog message indicating so as follows:

```
%SPANTREE-2-ROOTGUARD_BLOCK: Root guard blocking port FastEthernet0/1 on
VLAN0010.
```

When a switchport is in the inconsistent state, no manual intervention is required to recover the port from the inconsistent state. All you need to do is remove the device that is sending the superior BPDUs to that switchport from the network, and once the switchport no longer hears the superior BPDUs, the port is automatically taken out of the inconsistent state.

Loop Guard

Loop Guard is a feature designed to provide additional protection against Layer 2 loops. By default, if a nondesignated port ceases to receive BPDUs, it will transition to the forwarding state once the max age timer expires. However, what if the switch was not receiving the BPDUs because the switch that was sending the BPDUs had a software failure preventing it from sending BPDUs? That switch, would still be able to send and receive data on the interface. This would produce a loop because the nondesignated port is now sending and receiving data, as well, instead of blocking it. This is all because the BPDUs are no longer arriving on the interface. Loop Guard ensures that the nondesignated port does not erroneously transition to the forwarding state. Instead, it places it in the loop-inconsistent blocking state and generates the following syslog message:

```
%SPANTREE-2-LOOPGUARD_BLOCK: Loop guard blocking port GigabitEthernet0/2 on
VLAN0010.
```

To verify which ports are in the loop-inconsistent state, issue the command **show spanning-tree inconsistent** ports, as shown in Example 5-14.

Example 5-14 Verifying Loop-Inconsistent Ports on a Switch

```
SW3#show spanning-tree inconsistent ports
```

Name	Interface	Inconsistency
VLAN0010	GigabitEthernet0/2	Loop Inconsistent

```
Number of inconsistent ports (segments) in the system : 1
```

STP Trouble Tickets

This section presents various trouble tickets relating to the topics discussed earlier in the chapter. The purpose of these trouble tickets is to give a process that you can follow when troubleshooting in the real world or in an exam environment. All trouble tickets in this section are based on the topology depicted in Figure 5-6.

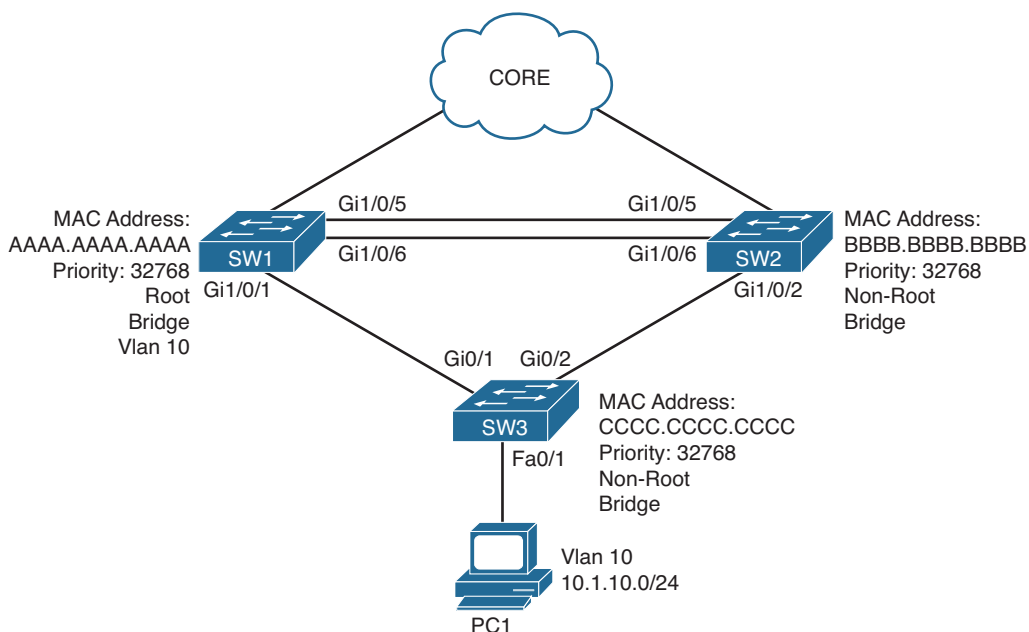


Figure 5-6 STP Trouble Ticket Topology

Trouble Ticket 5-1

Problem: Based on traffic analyzers, all traffic from the end stations in VLAN 10 destined to the core is flowing through SW2 when it should be flowing through SW1.

According to the topology, SW1 should be the root bridge for VLAN 10. Therefore, all traffic for VLAN 10 should be flowing through SW1 under normal conditions. With this in mind, check the placement of the root bridge using the **show spanning-tree vlan 10** command on SW1, as shown in Example 5-15. Notice that SW1 is not the root bridge for VLAN 10. According to the root ID section of the output, the switch with the MAC address bbbb.bbbb.bbbb is the root bridge.

Example 5-15 show spanning-tree vlan 10 Command Output for SW1

```
SW1#show spanning-tree vlan 10
VLAN0010
  Spanning tree enabled protocol ieee
  Root ID    Priority    10
             Address     bbbb.bbbb.bbbb
             Cost        4
             Port        5 (GigabitEthernet1/0/5)
             Hello Time  2 sec    Max Age 20 sec    Forward Delay 15 sec

  Bridge ID   Priority    32778 (priority 32768 sys-id-ext 10)
             Address     aaaa.aaaa.aaaa
```

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec					
Aging Time 300					
Interface	Role	Sts	Cost	Prio.Nbr	Type
Gi1/0/1	Desg	FWD	4	128.1	P2p
Gi1/0/5	Root	FWD	4	128.5	P2p
Gi1/0/6	Altn	BLK	4	128.6	P2p

Next you should check which switch is the root bridge. Figure 5-6 shows that bbbb.bbbb.bbbb is the MAC of SW2. However, without the diagram, how would you figure out who the root bridge is? You would follow the path. According to the output in Example 5-15, the port on SW1 to get to the root bridge is Gigabit Ethernet 1/0/5. At the bottom of the output, you can confirm that this is the root port. Therefore, using the **show cdp neighbors** command, you can confirm that SW2 is directly connected to SW1 on port Gi1/0/5, as shown in Example 5-16.

Example 5-16 show cdp neighbors Command Output on SW1

SW1#show cdp neighbors					
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge					
S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone,					
D - Remote, C - CVTA, M - Two-port Mac Relay					
Device ID	Local Intrfce	Holdtme	Capability	Platform	Port ID
SW2	Gig 1/0/6	138	S I	WS-C3750E	Gig 1/0/6
SW2	Gig 1/0/5	138	S I	WS-C3750E	Gig 1/0/5
SW3	Gig 1/0/1	141	S I	WS-C2960-	Gig 0/1

You should now verify if SW2 is the root bridge for VLAN 10 using the output of **show spanning-tree vlan 10**, as shown in Example 5-17. The output shows that SW2 is the root bridge for VLAN 10. It explicitly states *This bridge is the root*, and notice that all the ports are designated ports.

Example 5-17 show spanning-tree vlan 10 Command Output for SW2

```
SW2#show spanning-tree vlan 10
VLAN0010
    Spanning tree enabled protocol ieee
    Root ID    Priority      10
               Address      bbbb.bbbb.bbbb
               This bridge is the root
               Hello Time    2 sec    Max Age 20 sec    Forward Delay 15 sec

    Bridge ID   Priority      10          (priority 0 sys-id-ext 10)
               Address      bbbb.bbbb.bbbb
               Hello Time    2 sec    Max Age 20 sec    Forward Delay 15 sec
               Aging Time    300 sec
```

Interface	Role	Sts	Cost	Prio.Nbr	Type
-----	----	----	-----	-----	-----
Gi1/0/2	Desg	FWD	4	128.2	P2p
Gi1/0/5	Desg	FWD	4	128.5	P2p
Gi1/0/6	Desg	FWD	4	128.6	P2p

Upon further analysis of Example 5-17, you will notice that the priority of SW2 is 0 plus the extended system ID (which is the VLAN number), for a total value of 10, which is lower than the priority of SW1, which is 32768 plus 10 (32778), as shown in Example 5-15. It appears that the priority of SW2 was manually lowered. Using the command **show run | section spanning-tree** indicates that the command **spanning-tree vlan 10 priority 0** was executed on SW2, as shown in Example 5-18.

Example 5-18 show run Command Output for SW2

```
SW2#show run | section spanning-tree
...output omitted...
spanning-tree vlan 10 priority 0
...output omitted...
```

To solve this issue, we would need to remove this command by executing the **no spanning-tree vlan 10 priority 0** command. Once done, we can verify that SW1 is now the root bridge for VLAN 10 with the **show spanning-tree vlan 10** command, as shown in Example 5-19.

Example 5-19 show spanning-tree vlan 10 Command Output for SW1

```
SW1#show spanning-tree vlan 10
VLAN0010
  Spanning tree enabled protocol ieee
  Root ID    Priority    32778
             Address     aaaa.aaaa.aaaa
             This bridge is the root
  Hello Time  2 sec    Max Age 20 sec    Forward Delay 15 sec

  Bridge ID  Priority    32778    (priority 32768 sys-id-ext 10)
  Address     aaaa.aaaa.aaaa
  Hello Time  2 sec    Max Age 20 sec    Forward Delay 15 sec
  Aging Time  300 sec

Interface      Role Sts Cost      Prio.Nbr Type
-----
Gi1/0/1        Desg FWD 4         128.1    P2p
Gi1/0/5        Desg FWD 4         128.5    P2p
Gi1/0/6        Desg FWD 4         128.6    P2p
```

Trouble Ticket 5-2

Problem: Based on traffic analyzers, all traffic from the end stations in VLAN 10 destined to the core is flowing through SW2 when it should be flowing through SW1.

According to the topology, SW1 should be the root bridge for VLAN 10. Therefore, all traffic for VLAN 10 should be flowing through SW1 under normal conditions. With this in mind, check the placement of the root bridge using the **show spanning-tree vlan 10** command on SW1, as shown in Example 5-20. Notice that SW1 is the root bridge for VLAN 10.

Example 5-20 show spanning-tree vlan 10 Command Output for SW1

```
SW1#show spanning-tree vlan 10
VLAN0010
  Spanning tree enabled protocol ieee
  Root ID    Priority    32778
             Address     aaaa.aaaa.aaaa
             This bridge is the root
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32778      (priority 32768 sys-id-ext 10)
             Address     aaaa.aaaa.aaaa
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  300 sec

Interface                Role Sts Cost          Prio.Nbr Type
-----
Gil/0/1                  Desg FWD 4           128.1   P2p
Gil/0/5                  Desg FWD 4           128.5   P2p
Gil/0/6                  Desg FWD 4           128.6   P2p
```

We have confirmed that SW1 is the root bridge and this matches our diagram in Figure 5-6. If Figure 5-6 has been kept up to date, we can trust the information displayed.

According to Figure 5-6, we have a Gigabit Ethernet link between SW3 and SW1 as well as SW3 and SW2. These links should have a cost of 4 by default. Reviewing the output of **show spanning-tree vlan 10** on SW3, we can see that to reach the root bridge the total cost is 8 using Gigabit Ethernet 0/2, as shown in Example 5-21. If we look at Gig0/1, it is currently an alternate port in the blocking state with a cost of 10. This cost of 10 is larger than the total cost of 8 using Gig0/2. It appears that the cost of interface Gig0/1 has been modified.

Example 5-21 show spanning-tree vlan 10 Command Output for SW3

```
SW3#show spanning-tree vlan 10
VLAN0010
  Spanning tree enabled protocol ieee
  Root ID    Priority    32778
```

```

Address      aaaa.aaaa.aaaa
Cost         8
Port         2 (GigabitEthernet0/2)
Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

Bridge ID    Priority      32778      (priority 32768 sys-id-ext 10)
Address      cccc.cccc.cccc
Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
Aging Time   300 sec

Interface          Role Sts Cost          Prio.Nbr Type
-----
Gi0/1              Altn BLK 10          128.1    P2p
Gi0/2              Root FWD 4           128.2    P2p

```

The output of **show run interface gig 0/1** confirms that the cost was modified with the **spanning-tree vlan 10 cost 10** command, as shown in Example 5-22. To solve this issue, we need to execute the **no spanning-tree vlan 10 cost 10** command in interface configuration mode.

Example 5-22 show run interface gig 0/1 Command Output for SW3

```

SW3#show run interface gig 0/1
...output omitted...
spanning-tree vlan 10 cost 10
...output omitted...

```

After we remove the command, we can verify that SW3 is using Gi0/1 as the root port and that it has a cost of 4 by issuing the **show spanning-tree vlan 10** command shown in Example 5-23.

Example 5-23 show spanning-tree vlan 10 Command Output for SW3

```

SW3#show spanning-tree vlan 10
VLAN0010
Spanning tree enabled protocol ieee
Root ID    Priority      32778
Address    aaaa.aaaa.aaaa
Cost       4
Port       1 (GigabitEthernet0/1)
Hello Time 2 sec  Max Age 20 sec  Forward Delay 15 sec

Bridge ID  Priority      32778      (priority 32768 sys-id-ext 10)
Address    cccc.cccc.cccc
Hello Time 2 sec  Max Age 20 sec  Forward Delay 15 sec
Aging Time 300 sec

Interface          Role Sts Cost          Prio.Nbr Type
-----

```


Gi0/1	Root FWD 4	128.1	P2p
Gi0/2	Altn BLK 4	128.2	P2p

Trouble Ticket 5-3

Problem: It is Tuesday morning, and a user has indicated that he cannot connect to the network. He also indicates that he had no issues on Monday when he left work at 5:45 p.m.

You attempt to ping from the user's PC to its default gateway, but it fails. You attempt to ping from the user's PC to the Internet, but it fails. Your next task is to make sure that the PC is receiving an IP address from the Dynamic Host Configuration Protocol (DHCP) server in the network. Issuing the command **ipconfig /all** on the PC as depicted in Example 5-24 indicates that an Automatic Private IP Addressing (APIPA) address (169.254.x.x/16) is being used by the PC. Therefore, they are not able to contact a DHCP server. Also note the MAC address of PC1 at this point, as it will be useful later.

Example 5-24 ipconfig Output for PC

```
PC1>ipconfig /all
Windows Ip Configuration
<Output Omitted>
Ethernet adapter Local Area Connection:
<Output Omitted>
Physical Address. . . . . : 08-00-27-5D-06-D6
Link-local IPv6 Address . . . . . : fe80::444c:23b1:6e1e:de0c%16
Dhcp enabled. . . . . : Yes
Autoconfiguration enabled. . . . . : Yes
Autoconfiguration IP Address. . . . . : 169.254.180.166
Subnet Mask . . . . . : 255.255.0.0
<Output Omitted>
```

Issuing the command **show mac address-table dynamic** on SW3 will indicate whether SW3 is receiving any frames from PC1. Example 5-25 is displaying the MAC address table of SW3, and there is no entry in the table with PC1's MAC address. Therefore, something appears to be wrong at Layer 1 or Layer 2 of the OSI model.

Example 5-25 show mac address-table dynamic Output for SW3

```
SW3#show mac address-table dynamic
Mac Address Table
-----
Vlan    Mac Address      Type    Ports
----
10      0800.275d.1234   DYNAMIC Gi0/1
10      0800.275d.ac47   DYNAMIC Fa0/4
10      0800.275d.b3dd   DYNAMIC Fa0/3
```

```

10      0800.275d.ce47      DYNAMIC      Fa0/2
10      0800.275d.ed13      DYNAMIC      Gi0/1
Total Mac Addresses for this criterion: 6

```

You verify physical connectivity and everything is perfect. However, you notice that the LED of the switchport PC1 is connected to is amber rather than green, confirming that something is not right. According to Figure 5-6, PC1 should be in VLAN 10. Issuing the command **show vlan brief** will confirm this for us. Example 5-26 shows that interface Fa0/1, which is connected to PC1, is in VLAN 10. In addition, the output of **show interfaces status | include Fa0/1**, as shown in Example 5-27, does not indicate that anything is wrong.

Example 5-26 show vlan brief Output for SW3

```

SW3#show vlan brief
VLAN Name                Status    Ports
-----
1    default              active    Fa0/5, Fa0/6, Fa0/7, Fa0/8,
                                   Fa0/9, Fa0/10, Fa0/11, Fa0/12,
                                   Fa0/13, Fa0/14, Fa0/15, Fa0/16,
                                   Fa0/17, Fa0/18, Fa0/19, Fa0/20,
                                   Fa0/21, Fa0/22, Fa0/23, Fa0/24,
10   10.1.10.0/24          active    Fa0/1, Fa0/2, Fa0/3, Fa0/4,
20   10.1.20.0/24          active
1002 fddi-default          act/unsup
1003 trcrf-default        act/unsup
1004 fddinet-default       act/unsup
1005 trbrf-default        act/unsup

```

Example 5-27 show interfaces status | include Fa0/1 Output for SW3

```

SW3#show interfaces status | include Fa0/1
Port      Name                Status    Vlan    Duplex  Speed Type
Fa0/1     connected          10        a-full  a-10010/100BaseTX

```

No other users at this point have indicated that they are experiencing issues. You decide to check the SW3 logs on your syslog server and notice the following entry:

```

%SPANTREE-2-ROOTGUARD_BLOCK: Root guard blocking port FastEthernet0/1 on
VLAN0010.

```

It appears that BPDUs are being received by Fast Ethernet 0/1 from PC1. Issuing the command **show spanning-tree inconsistentports** on SW3 confirms that Fast Ethernet 0/1 is in the root-inconsistent state, as shown in Example 5-28.

Example 5-28 *show spanning-tree inconsistentports Output for SW3*

```
SW3#show spanning-tree inconsistentports
Name                Interface          Inconsistency
-----
VLAN0010            FastEthernet0/1    Root Inconsistent

Number of inconsistent ports (segments) in the system : 1
```

Upon further examination, beyond the scope of this book, an application was installed on PC1 after hours that mimics a switch and sends BPDUs. Further investigation will be needed to determine whether this was malicious or by accident.

To solve this issue, we remove the offending application from PC1, and the switch will recover the port automatically, as shown in Example 5-29.

Example 5-29 *SW3 show spanning-tree inconsistenetports Output After Application Removed from PC1*

```
SW3#
%SPANTREE-2-ROOTGUARD_UNBLOCK: Root guard unblocking port FastEthernet0/1 on
VLAN0010.
SW3#show spanning-tree inconsistentports

Name                Interface          Inconsistency
-----

Number of inconsistent ports (segments) in the system : 0
```

The output of **ipconfig** on PC1 in Example 5-30 verifies it has an IP address and a **ping** to 10.1.10.1, PC1's default gateway, is successful.

Example 5-30 *ipconfig and ping Output for PC After Issue Solved*

```
PC1>ipconfig
Windows IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . : domain.local
    Link-local IPv6 Address . . . . . : fe80::444c:23b1:6e1e:de0c%16
    IPv4 Address. . . . . : 10.1.10.10
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 10.1.10.1

PC1>ping 10.1.10.1
Pinging 10.1.10.1 with 32 bytes of data:
Reply from 10.1.10.1: bytes=32 time<1ms TTL=255
Reply from 10.1.10.1: bytes=32 time=1ms TTL=255
Reply from 10.1.10.1: bytes=32 time=3ms TTL=255
Reply from 10.1.10.1: bytes=32 time=1ms TTL=255
```

```
Ping statistics for 10.1.10.1:
  Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
  Minimum = 0ms, Maximum = 3ms, Average = 1ms
```

Troubleshooting Layer 2 EtherChannel

An exception to STP operation can be made if two switches are interconnected via multiple physical links and those links are configured as an *EtherChannel*. An EtherChannel logically combines the bandwidth of multiple physical interfaces into a logical connection between switches, as illustrated in Figure 5-7. Specifically, Figure 5-7 shows four Gigabit Ethernet links logically bonded into a single EtherChannel link.

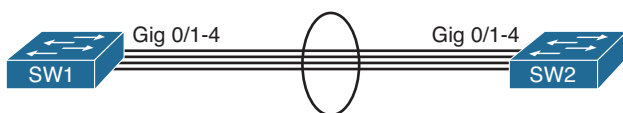


Figure 5-7 Layer 2 EtherChannel

This section reviews what is necessary to successfully form a Layer 2 EtherChannel bundle and the EtherChannel mode combinations that will successfully form the bundle.

Reviewing Layer 2 EtherChannel

When multiple ports are combined into a logical EtherChannel, STP treats the logical bundle (known as a *port channel*) as a single port for STP calculation purposes. Following are common troubleshooting targets to consider when troubleshooting an EtherChannel issue:



- **Mismatched port configurations:** The configurations of all ports making up an EtherChannel, on both switches, should be identical. For example, all ports should have the same speed, duplex, trunk mode, native VLAN configurations, allowed VLAN configurations, and port type (Layer 2 or Layer 3).
- **Mismatched EtherChannel configuration:** Both switches forming the EtherChannel should be configured with compatible modes. There are three options, Link Aggregation Control Protocol (LACP), Port Aggregation Protocol (PAgP), and ON. These modes are not compatible with each other. In addition, when using LACP or PAgP, you have to make sure that the modes within the protocol can successfully form the bundle with each other. Table 5-4 identifies which modes can be configured on each switch to successfully form an EtherChannel bundle.
- **Inappropriate EtherChannel distribution algorithm:** EtherChannel determines which physical link to use to transmit frames based on a hash calculation. The hashing approach selected should distribute the load fairly evenly across all physical links. For example, a hash calculation might be based only on the destination MAC

address of a frame. If the frames are destined for only a few different MAC addresses, the load distribution could be uneven. To verify the load-balancing algorithm in use, issue the **show etherchannel load-balance** command.



Table 5-4 *EtherChannel Modes That Will Successfully Form a Bundle*

	SW1					
	MODE	PAgP Desirable	PAgP Auto	LACP Active	LACP Passive	ON
SW2	PAgP Desirable	Yes	Yes	No	No	No
	PAgP Auto	Yes	No	No	No	No
	LACP Active	No	No	Yes	Yes	No
	LACP Passive	No	No	Yes	No	No
	ON	No	No	No	No	Yes

EtherChannel Trouble Tickets

This section presents various trouble tickets relating to the topics discussed earlier in the chapter. The purpose of these trouble tickets is to give a process that you can follow when troubleshooting in the real world or in an exam environment. All trouble tickets in this section are based on the topology depicted in Figure 5-8.

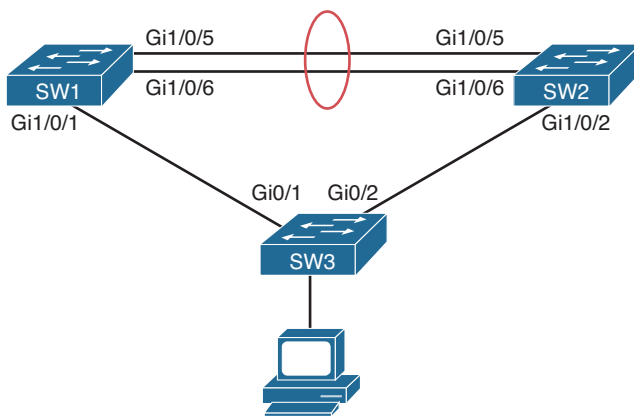


Figure 5-8 *Layer 2 EtherChannel Trouble Ticket Topology*

Trouble Ticket 5-4

Problem: A junior network administrator has approached you indicating that the EtherChannel bundle she is trying to form between SW1 and SW2 is not forming. You need to solve this issue for her.

You start by reviewing the output of **show etherchannel summary** for SW1 and SW2, as shown in Example 5-31. Notice that both switches are using LACP as their protocol; however, the ports are either standalone or suspended, and the port channel is down. This is a good indication that there is a conflict with the port configurations.

Example 5-31 show etherchannel summary Output for SW1 and SW2

```
SW1#show etherchannel summary
Flags:  D - down          P - bundled in port-channel
        I - stand-alone   s - suspended
        H - Hot-standby (LACP only)
        R - Layer3        S - Layer2
        U - in use        f - failed to allocate aggregator
        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port
```

Number of channel-groups in use: 1

Number of aggregators: 1

Group	Port-channel	Protocol	Ports
1	Po1 (SD)	LACP	Gi1/0/5 (I) Gi1/0/6 (s)

```
SW2#show etherchannel summary
```

```
Flags:  D - down          P - bundled in port-channel
        I - stand-alone   s - suspended
        H - Hot-standby (LACP only)
        R - Layer3        S - Layer2
        U - in use        f - failed to allocate aggregator
        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port
```

Number of channel-groups in use: 1

Number of aggregators: 1

Group	Port-channel	Protocol	Ports
1	Po1 (SD)	LACP	Gi1/0/5 (I) Gi1/0/6 (I)

To verify the port configuration you issue the **show run interface gigabitethernet 1/0/5** and **show run interface gigabitethernet 1/0/6** command on SW1 and SW2, as shown in Example 5-32. If you look closely, you will notice that the switchport modes do not match on the SW1 interfaces that are part of the EtherChannel bundle. To form the bundle, they have to match.

Example 5-32 *show run interface gigabitethernet Output for SW1 and SW2*

```
SW1#show run interface gigabitethernet 1/0/5
Building configuration...
Current configuration : 189 bytes
!
interface GigabitEthernet1/0/5
  switchport trunk encapsulation isl
  switchport mode access
  switchport nonegotiate
  channel-group 1 mode active
end
SW1#show run interface gigabitethernet 1/0/6
Building configuration...
Current configuration : 189 bytes
!
interface GigabitEthernet1/0/6
  switchport trunk encapsulation isl
  switchport mode trunk
  switchport nonegotiate
  channel-group 1 mode active
end

SW2#show run interface gigabitethernet 1/0/5
Building configuration...
Current configuration : 151 bytes
!
interface GigabitEthernet1/0/5
  switchport trunk encapsulation isl
  switchport mode trunk
  switchport nonegotiate
  channel-group 1 mode passive
end
SW2#show run interface gigabitethernet 1/0/6
Building configuration...

Current configuration : 151 bytes
!
interface GigabitEthernet1/0/6
  switchport trunk encapsulation isl
  switchport mode trunk
```

```
switchport nonegotiate
channel-group 1 mode passive
end
```

Once you change the switchport mode on SW1 Gigabit Ethernet 1/0/5 with the **switchport mode trunk** command, the port channel interface should come up, as shown with the following logging messages:

```
%LINK-3-UPDOWN: Interface Port-channel1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Port-channel1, changed state to up
```

In addition, the EtherChannel bundle should now be successfully formed. Reviewing the output of **show etherchannel summary** on SW1 and SW2 indicates that the ports are successfully bundled with the (P) flags and that the port channel is in use with the (U) flag, as shown in Example 5-33.

Example 5-33 *show etherchannel summary Output for SW1 and SW2 After Problem Solved*

```
SW1#show etherchannel summary
Flags:  D - down          P - bundled in port-channel
        I - stand-alone  s - suspended
        H - Hot-standby (LACP only)
        R - Layer3       S - Layer2
        U - in use       f - failed to allocate aggregator
        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port

Number of channel-groups in use: 1
Number of aggregators:          1

Group  Port-channel  Protocol    Ports
-----+-----+-----+-----
1      Po1(SU)       LACP        Gi1/0/5(P) Gi1/0/6(P)
```

```
SW2#show etherchannel summary
Flags:  D - down          P - bundled in port-channel
        I - stand-alone  s - suspended
        H - Hot-standby (LACP only)
        R - Layer3       S - Layer2
        U - in use       f - failed to allocate aggregator
        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port
```



```

Number of channel-groups in use: 1
Number of aggregators:          1

Group  Port-channel  Protocol    Ports
-----+-----+-----+-----
1      Po1 (SU)        LACP        Gi1/0/5 (P) Gi1/0/6 (P)

```

Trouble Ticket 5-5

Problem: A junior network administrator has approached you indicating that the EtherChannel bundle he is trying to form between SW1 and SW2 is not forming. You need to solve this issue for him.

You start by checking whether the port channel is up on SW1 and SW2, as shown in Example 5-34. According to the output, it is down/down.

Example 5-34 show ip interface brief | include Port Output for SW1 and SW2

```

SW1#show ip interface brief | include Port
Port-channel1          unassigned      YES unset  down      down

SW2#show ip interface brief | include Port
Port-channel1          unassigned      YES unset  down      down

```

Next you check the status of the EtherChannel bundle with the **show etherchannel summary** command, as shown in Example 5-35. Notice that the port channel is down and that all interfaces are standalone. However, if you look closer, you will see the issue. SW1 is using PAgP, and SW2 is using LACP. These EtherChannel protocols are not compatible. Therefore, to solve this issue, you will need to verify your documentation to determine which protocol should be used between SW1 and SW2 and make the appropriate adjustments.

Example 5-35 show etherchannel summary Output for SW1 and SW2

```

SW1#show etherchannel summary
Flags:  D - down          P - bundled in port-channel
        I - stand-alone  s - suspended
        H - Hot-standby (LACP only)
        R - Layer3       S - Layer2
        U - in use       f - failed to allocate aggregator
        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port

Number of channel-groups in use: 1
Number of aggregators:          1

```

```

Group  Port-channel  Protocol  Ports
-----+-----+-----+-----
1      Po1 (SD)        PAgP      Gi1/0/5 (I) Gi1/0/6 (I)

SW2#show etherchannel summary
Flags:  D - down          P - bundled in port-channel
        I - stand-alone   s - suspended
        H - Hot-standby (LACP only)
        R - Layer3        S - Layer2
        U - in use        f - failed to allocate aggregator
        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port

Number of channel-groups in use: 1
Number of aggregators:          1

Group  Port-channel  Protocol  Ports
-----+-----+-----+-----
1      Po1 (SD)        LACP      Gi1/0/5 (I) Gi1/0/6 (I)

```

Exam Preparation Tasks

As mentioned in the section “How to Use This Book” in the Introduction, you have a couple of choices for exam preparation: the exercises here; Chapter 22, “Final Preparation;” and the exam simulation questions on the CD-ROM.

Review All Key Topics

Review the most important topics in this chapter, noted with the Key Topic icon in the outer margin of the page. Table 5-5 lists a reference of these key topics and the page numbers on which each is found.



Table 5-5 *Key Topics for Chapter 5*

Key Topic Element	Description	Page Number
List	Describes root bridge election	173
Sentence	Identifies the golden rule of STP	173
Table 5-2	Identifies STP port types	174
Table 5-3	Identifies STP port costs	175
Section	Reviews how to determine root ports	175
Section	Reviews how to determine designated ports	176
List	Identifies STP port states	177
Section	Identifies show commands used for troubleshooting STP	178
Section	Reviews STP features and the show commands used for troubleshooting	182
List	Describes issues that could prevent an EtherChannel from forming	199
Table 5-4	Identifies the EtherChannel modes that will successfully form a bundle	200

Define Key Terms

Define the following key terms from this chapter and check your answers in the glossary:

Spanning Tree Protocol (STP), root bridge, root port, designated port, nondesignated port, blocking, listening, learning, forwarding, 802.1D, 802.1w, 802.1s, Layer 2 EtherChannel, PAgP, LACP

Complete Tables and Lists from Memory

Print a copy of Appendix C, “Memory Tables,” (found on the disc), or at least the section for this chapter, and complete the tables and lists from memory. Appendix D, “Memory Tables Answer Key,” also on the disc, includes completed tables and lists to check your work.

Command Reference to Check Your Memory

This section includes the **show** commands introduced in this chapter. It does not include the **show** commands that were used in this chapter but introduced in previous chapters. You will need to return to the previous chapters to review information relating to those **show** commands.

To test your memory of the commands, cover the right side of Table 5-6 with a piece of paper, read the description on the left side, and then see how much of the command you can remember.

The 300-135 TSHOOT exam focuses on practical, hands-on skills that are used by a networking professional. Therefore, you should be able to identify the show commands needed to successfully troubleshoot the topics covered in this chapter.

Table 5-6 *show Commands Introduced in Chapter 5*

Task	Command Syntax
Displays STP information about all VLANs	show spanning-tree
Displays STP information about a specific VLAN	show spanning-tree [vlan {vlan_id}]
Displays the STP interface role, cost, port priority, and type for each VLAN on the switch	show spanning-tree interface interface_type interface_number
Displays detailed STP information about an interface, including the number of BPDUs sent and received and the STP features that have been enabled specifically on the interface	show spanning-tree interface interface_type interface_number detail
Displays the MST region name, revision number, and the instance to VLAN mappings	show spanning-tree mst configuration
Displays ports configured with Root Guard that have received superior BPDUs and ports configured with Loop Guard that are in the loop inconsistent state	show spanning-tree inconsistentports
Displays which STP features have been enabled globally on the switch	show spanning-tree summary
Displays the status of port-channels as well as the status of the ports within the port channel	show etherchannel summary
Displays the EtherChannel load-balance algorithm configured on the switch	show etherchannel load-balance



This chapter covers the following topics:

- **Troubleshooting a Router-on-a-Trunk/Stick:** This section covers how to troubleshoot inter-VLAN routing issues when using the router-on-a-trunk scenario.
- **Router-on-a-Trunk/Stick Trouble Tickets:** This section provides trouble tickets that demonstrate how you can use a structured troubleshooting process to solve a reported problem.
- **Troubleshooting Switched Virtual Interfaces:** This section identifies what is necessary for an SVI to be up/up and provide inter-VLAN routing. You will also learn how to troubleshoot issues related to SVIs.
- **SVI Trouble Tickets:** This section provides trouble tickets that demonstrate how you can use a structured troubleshooting process to solve a reported problem.
- **Troubleshooting Routed Ports:** This section reviews what is necessary to convert a Layer 2 switchport into a routed port.
- **Routed Port Trouble Tickets:** This section provides trouble tickets that demonstrate how you can use a structured troubleshooting process to solve a reported problem.
- **Troubleshooting Layer 3 EtherChannel:** This section focuses on the steps needed to successfully troubleshoot a Layer 3 EtherChannel that relies on routed ports.
- **Layer 3 EtherChannel Trouble Tickets:** This section provides trouble tickets that demonstrate how you can use a structured troubleshooting process to solve a reported problem.

Troubleshooting Inter-VLAN Routing and Layer 3 EtherChannels

Chapters 4, “Troubleshooting Layer 2 Trunks, VTP, and VLANs,” and 5, “Troubleshooting STP and Layer 2 EtherChannel,” focused on Cisco Catalyst switches as Layer 2 switches. These switches operate at Layer 2 of the OSI model, forwarding or flooding frames based on the MAC addresses in the frame. However, many Cisco Catalyst switches are Layer 3 switches. These Layer 3 switches can perform both Layer 2 and Layer 3 services.

Of the Layer 3 services, routing is the most common that is implemented. Through the use of virtual Layer 3 interfaces (known as *switched virtual interfaces* [SVIs]) or by converting a Layer 2 switchport to a routed port, you can assign IP addresses to these interfaces and have the Layer 3 switch route data between VLANs and subnets. In addition, you can use routed ports to create Layer 3 EtherChannels.

This chapter focuses on how you can troubleshoot different inter-VLAN routing implementations, routed ports, and Layer 3 EtherChannel. You will also be exposed to a few different troubleshooting scenarios for each.

“Do I Know This Already?” Quiz

The “Do I Know This Already?” quiz allows you to assess whether you should read this entire chapter thoroughly or jump to the “Exam Preparation Tasks” section. If you are in doubt about your answers to these questions or your own assessment of your knowledge of the topics, read the entire chapter. Table 6-1 lists the major headings in this chapter and their corresponding “Do I Know This Already?” quiz questions. You can find the answers in Appendix A, “Answers to the ‘Do I Know This Already?’ Quizzes.”

Table 6-1 “Do I Know This Already?” Section-to-Question Mapping

Foundation Topics Section	Questions
Troubleshooting a Router-on-a-Trunk/Stick	1–2
Troubleshooting Switched Virtual Interfaces	3–5
Troubleshooting Routed Ports	6–7
Troubleshooting Layer 3 EtherChannel	8–9

Caution The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you do not know the answer to a question or are only partially sure of the answer, you should mark that question as wrong for purposes of the self-assessment. Giving yourself credit for an answer that you correctly guess skews your self-assessment results and might provide you with a false sense of security.

1. Which command enables you to associate a VLAN with a router subinterface?
 - a. encapsulation
 - b. interface
 - c. ip address
 - d. vlan
2. Which **show** command enables you to verify the VLAN that has been associated with a router subinterface?
 - a. show interface trunk
 - b. show vlan brief
 - c. show ip route
 - d. show vlans
3. What must be true for an SVI to be up/up? (Choose two answers.)
 - a. The VLAN associated with the SVI must exist on the switch.
 - b. The SVI must be disabled.
 - c. There must be at least one interface on the switch associated with the VLAN in the spanning-tree forwarding state.
 - d. IP routing must be enabled on the switch.
4. Which **show** command enables you to verify the status of the SVI for VLAN 10 and the MAC address associated with it?
 - a. show ip interface brief
 - b. show interfaces vlan 10
 - c. show ip interface vlan 10
 - d. show svi
5. Which command enables IPv4 unicast routing on a Layer 3 switch?
 - a. routing
 - b. ip route
 - c. ip routing
 - d. ip unicast-routing

6. Which command enables you to convert a Layer 2 switchport to a routed port?
 - a. no switchport
 - b. routed port
 - c. ip address
 - d. ip routing
7. Which **show** command enables you to verify whether interface Gigabit Ethernet 1/0/10 is a Layer 2 switchport or a routed port?
 - a. show gigabitethernet 1/0/10 switchport
 - b. show interfaces gigabitethernet 1/0/10
 - c. show interfaces gigabitethernet 1/0/10 switchport
 - d. show interfaces status
8. What flags in the **show etherchannel summary** output indicate that the EtherChannel is Layer 3 and in use?
 - a. SU
 - b. SD
 - c. RU
 - d. RD
9. Which EtherChannel modes will successfully form an LACP EtherChannel?
 - a. Active-auto
 - b. Desirable-auto
 - c. Passive-desirable
 - d. Active-passive
10. Which EtherChannel flag indicates that the port is bundled in the EtherChannel bundle?
 - a. R
 - b. S
 - c. P
 - d. H

Foundation Topics

Troubleshooting a Router-on-a-Trunk/Stick

For traffic to pass from one VLAN to another VLAN, it has to be routed. This is easy to remember if you recall that a VLAN = a subnet and to send traffic from one subnet to another you route it. Therefore, to send traffic from one VLAN to another VLAN, you also route it.

This section reviews how you can use an external router that is trunked to a switch to perform routing between VLANs. The section also covers the various issues that could cause this implementation to not function as expected.

Before Layer 3 switches existed, we relied on external routers to perform inter-VLAN routing. The external router was connected to the Layer 2 switch via a trunk, which created the *router-on-a-stick* or *router-on-a-trunk* topology, as shown in Figure 6-1.

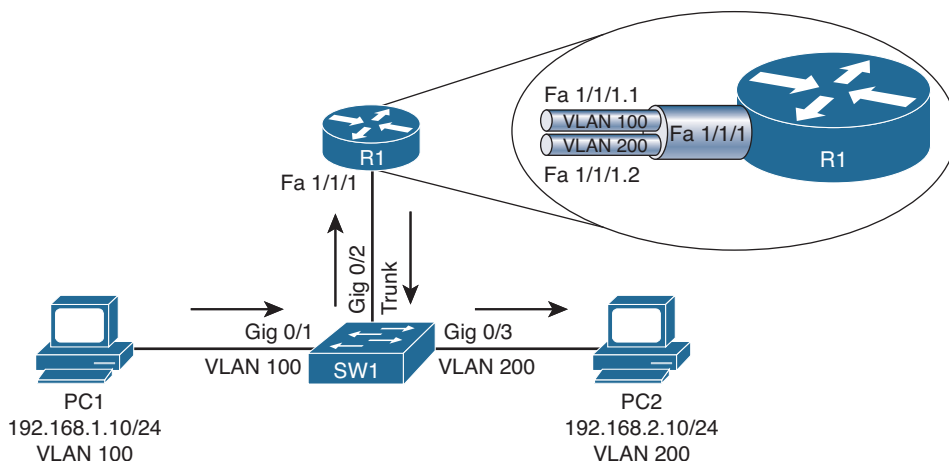


Figure 6-1 Router-on-a-Trunk / Router-on-a-Stick

In Figure 6-1, router R1's Fast Ethernet 1/1/1 interface has two subinterfaces as indicated by the period (.) in the interface identification. There is one for each VLAN, Fast Ethernet 1/1/1.1 for VLAN 100 and Fast Ethernet 1/1/1.2 for VLAN 200. Router R1 can route between VLANs 100 and 200, while simultaneously receiving and transmitting traffic over the trunk connection to the switch. Review Example 6-1 and Example 6-2, which outline the configurations needed to implement a router-on-a-trunk.

Key Topic

Example 6-1 show run Command Output from R1

```
R1#show run
...output omitted...
interface FastEthernet1/1/1.1
 encapsulation dot1Q 100
```

```

ip address 192.168.1.1 255.255.255.0
!
interface FastEthernet1/1/1.2
encapsulation dot1Q 200
ip address 192.168.2.1 255.255.255.0
...output omitted...

```

Key Topic

Example 6-2 show run Command Output from SW1

```

SW1#show run
...output omitted...
interface GigabitEthernet0/1
switchport mode access
switchport access vlan 100

interface GigabitEthernet0/2
switchport trunk encapsulation dot1q
switchport mode trunk
switchport nonegotiate

interface GigabitEthernet0/3
switchport mode access
switchport access vlan 200
...output omitted...

```

Key Topic

After reviewing Example 6-1 and Example 6-2, what are issues that could prevent inter-VLAN routing from being successful?

- Trunk encapsulation mismatch
- Incorrect VLAN assignment on routers' subinterfaces
- Incorrect IP address or subnet mask on routers' subinterfaces
- Incorrect IP address, subnet mask, or default gateway on PCs
- Switchport connected to router configured as an access port
- Switchport connected to router configured to use Dynamic Trunking Protocol (DTP), which is not supported by the router
- Switchports connected to PCs in wrong VLAN

Being able to identify these issues and correct them is important for any troubleshooter.

Router-on-a-Trunk/Stick Trouble Tickets

This section covers various trouble tickets relating to the topics discussed earlier in the chapter. The purpose of these trouble tickets is to give a process that you can follow when troubleshooting in the real world or in an exam environment. All trouble tickets in this section are based on the topology depicted in Figure 6-2.

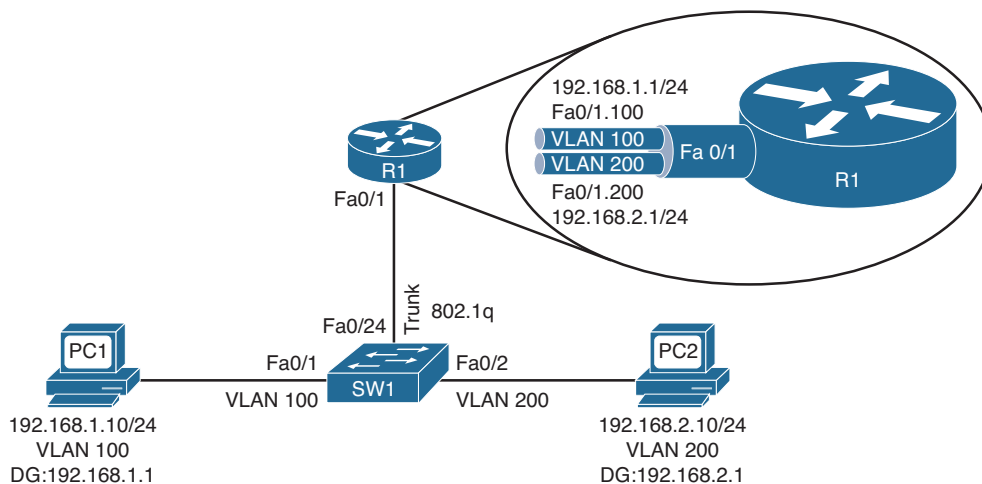


Figure 6-2 Router-on-a-Trunk Trouble Tickets

Trouble Ticket 6-1

Problem: PC1 is not able to access resources on PC2.

As you dive deeper into trouble tickets, everything covered in the previous chapters still applies because the PCs are still connected to the switches, there are still VLANs, and there are trunks. As a result, having a repeatable structured troubleshooting process in place will help you maintain focus and clarity as you troubleshoot.

The first item on the list of troubleshooting is to verify the problem. Issuing the **ping** command on PC1, as shown in Example 6-3, indicates that PC1 is not able to reach PC2, confirming the problem.

Example 6-3 Failed Ping from PC1 to PC2

```
C:\PC1>ping 192.168.2.10
Pinging 192.168.2.10 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.2.10:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Next you need to verify whether PC1 can get to its default gateway. This will help you narrow down where the issue may be. Pinging PC1's default gateway, as shown in Example 6-4, is not successful. This indicates that we have an issue between PC1 and the default gateway.

Example 6-4 *Failed Ping from PC1 to Default Gateway*

```

C:\PC1>ping 192.168.1.1
Pinging 192.168.1.1 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.1.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

```

Now is an excellent time to brainstorm the likely causes of the issue based on Figure 6-2 and the fact that PC1 is not able to ping its default gateway:

- PC1 may have an incorrect IP address, subnet mask, or default gateway configured.
- SW1 switchport FA0/1 may not be associated with the correct VLAN.
- VLAN 100 may not exist on SW1.
- PC1 may physically be connected to the wrong switchport.
- SW1 Fa0/24 may not be configured as a trunk.
- SW1 Fa0/24 may not be allowing VLAN 100 traffic on the trunk.
- SW1 Fa0/24 may be using the wrong trunk encapsulation.
- R1 may not have the appropriate subinterfaces configured with the correct IP addresses or subnet masks.
- R1's subinterfaces may be using the wrong trunk encapsulation.
- R1's subinterfaces may be disabled.

As you can see, the list is quite extensive, and it is not even a complete list. Let's start following the path from PC1 and work toward the router. Issuing **ipconfig** on PC1 indicates that it has the correct IP address, subnet mask, and default gateway configured, as shown in Example 6-5, when compared to Figure 6-2.

Example 6-5 *ipconfig Output on PC1*

```

C:\PC1>ipconfig
Windows IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . : 
    IP Address. . . . . : 192.168.1.10
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.1.1

```

Issuing the **show mac address-table dynamic** command on SW1 will identify which MAC address is being learned on Fa0/1 and which VLAN it is associated with. Example 6-6 is indicating that the MAC address of 0800.275d.06d6 is being learned on Fa0/1 and that it is associated with VLAN 100. Issuing the **ipconfig /all** command on PC1, as shown in Example 6-7, identifies PC1's MAC as 0800.275d.06d6, which is the same as the one outlined in the MAC address table. We can narrow our focus now because this proves that PC1 is connected to the correct switchport, VLAN 100 exists, and Fa0/1 is in the correct VLAN.

Example 6-6 show mac address-table dynamic Command Output on SW1

```
SW1#show mac address-table dynamic
      Mac Address Table
-----
Vlan    Mac Address      Type    Ports
----    -
100     0800.275d.06d6   DYNAMIC Fa0/1
200     0800.27a2.ce47   DYNAMIC Fa0/2
Total Mac Addresses for this criterion: 2
```

Example 6-7 ipconfig /all Output on PC1

```
C:\PC1>ipconfig
...output omitted...
Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . : 
    Description . . . . . : AMD PCNET Family PCI Ethernet Adapter
    Physical Address. . . . . : 08-00-27-5D-06-D6
    Dhcp Enabled. . . . . : No
    IP Address. . . . . : 192.168.1.10
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.1.1
...output omitted...
```

Focus on Example 6-6 again. If you look closely at the MAC address table on SW1, you will notice that no MAC addresses are being learned for VLAN 100 or VLAN 200 on Fa0/24. Why would this be? The link between R1 and SW1 should be an 802.1Q trunk according to Figure 6-2. If this trunk is not configured with the correct encapsulation, or the correct trunk mode, or the trunk is pruning VLAN 100 or 200 traffic, traffic for VLANs 100 and 200 would not pass over the link.

On SW1, start by issuing the **show interfaces trunk** command, as shown in Example 6-8. The output indicates that Fa0/24 is a trunk using mode on, which means the command **switchport mode trunk** was issued. It also indicates that Fa0/24 is using Inter-Switch Link (ISL) as the trunk encapsulation method. According to Figure 6-2, the trunk should be using 802.1Q.

Example 6-8 show interfaces trunk *Command Output on SW1*

```
SW1#show interfaces trunk
Port      Mode      Encapsulation  Status      Native vlan
Fa0/24    on        isl             trunking    1

Port      Vlans allowed on trunk
Fa0/24    1-4094

Port      Vlans allowed and active in management domain
Fa0/24    1,100,200

Port      Vlans in spanning tree forwarding state and not pruned
Fa0/24    1,100,200
```

Reviewing the output of **show vlans** on R1 in Example 6-9 confirms that R1 is using 802.1Q for its trunk encapsulation. As a result, we have a trunk encapsulation mismatch.

Example 6-9 show vlans *Output on R1*

```
R1#show vlans
...output omitted...
Virtual LAN ID: 100 (IEEE 802.1Q Encapsulation)

VLAN Trunk Interface:  FastEthernet0/1.100

Protocols Configured:  Address:              Received:             Transmitted:
      IP                192.168.1.1          4                    8
      Other              0                    5

4 packets, 298 bytes input
13 packets, 1054 bytes output

Virtual LAN ID: 200 (IEEE 802.1Q Encapsulation)

VLAN Trunk Interface:  FastEthernet0/1.200

Protocols Configured:  Address:              Received:             Transmitted:
      IP                192.168.2.1          4                    8
      Other              0                    5

4 packets, 298 bytes input
13 packets, 1054 bytes output
```

You need to fix SW1 so that Fa0/24 is using the correct trunk encapsulation method. On Fa0/24 of SW1, issue the **switchport trunk encapsulation dot1q** command. After you have implemented your solution, you need to confirm that it solved the problem by pinging from PC1 to PC2 again. Example 6-10 shows that the **ping** is successful.

Example 6-10 *Successful Ping from PC1 to PC2*

```

C:\PC1>ping 192.168.2.10

Reply from 192.168.2.10: bytes=32 time=1ms TTL=128
Reply from 192.168.2.10: bytes=32 time=1ms TTL=128
Reply from 192.168.2.10: bytes=32 time=1ms TTL=128
Reply from 192.168.2.10: bytes=32 time=1ms TTL=128

Ping statistics for 192.168.2.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

```

Trouble Ticket 6-2

Problem: PC1 is not able to access resources on PC2.

The problem reported in this trouble ticket is the exact same as the previous trouble ticket. However, do not jump to the conclusion that it is the same problem and solution. You always want to follow your structured troubleshooting approach to make sure that you efficiently solve the problem and waste little effort.

The first item on the list of troubleshooting is to verify the problem. Issuing the **ping** command on PC1, as shown in Example 6-11, indicates that PC1 is not able to reach PC2, confirming the problem.

Example 6-11 *Failed Ping from PC1 to PC2*

```

C:\PC1>ping 192.168.2.10

Pinging 192.168.2.10 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.2.10:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

```

Next you need to verify whether PC1 can get to its default gateway. This will help you narrow down where the issue may be. Pinging PC1's default gateway, as shown in Example 6-12, is successful. This indicates that we do not have an issue between PC1 and the default gateway.

Example 6-12 *Successful Ping from PC1 to Default Gateway*

```
C:\PC1>ping 192.168.1.1
Reply from 192.168.1.1: bytes=32 time 1ms TTL=128
Reply from 192.168.1.1: bytes=32 time 1ms TTL=128
Reply from 192.168.1.1: bytes=32 time 1ms TTL=128
Reply from 192.168.1.1: bytes=32 time 1ms TTL=128

Ping statistics for 192.168.1.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Now is a great time to check whether PC1 can ping the default gateway of VLAN 200 at 192.168.2.1. This will help you determine whether inter-VLAN routing is working on R1 between VLAN 100 and VLAN 200. The ping, as shown in Example 6-13, is successful.

Example 6-13 *Successful Ping from PC1 to Default Gateway of VLAN 200*

```
C:\PC1>ping 192.168.2.1
Reply from 192.168.2.1: bytes=32 time 1ms TTL=128
Reply from 192.168.2.1: bytes=32 time 1ms TTL=128
Reply from 192.168.2.1: bytes=32 time 1ms TTL=128
Reply from 192.168.2.1: bytes=32 time 1ms TTL=128

Ping statistics for 192.168.2.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

It is time to shift attention to R1 and PC2 because it appears everything is fine from PC1 to R1's subinterface Fa0/1.100. In this case, we will work our way backward from R1 to PC2. For VLAN 200 traffic to flow from R1 to PC2, the subinterface Fa0/1.200 needs to be using the correct encapsulation method (802.1Q), it needs to have the correct IP address and subnet mask assigned to it (192.168.2.1/24), and it needs to have the right VLAN assigned to it (VLAN 200). Using the command **show vlans** on R1 will help to verify the subinterface configuration on R1, as outlined in Example 6-14. Notice that subinterface Fa0/1.200 has the appropriate IP address and that it is also using 802.1Q as the trunk encapsulation. However, it is associated with VLAN 20, not VLAN 200. This appears to be the issue.

Example 6-14 *show vlans Command Output on R1*

```
R1#show vlans
...output omitted...
Virtual LAN ID: 20 (IEEE 802.1Q Encapsulation)

VLAN Trunk Interface: FastEthernet0/1.200
```


Protocols Configured:	Address:	Received:	Transmitted:
IP	192.168.2.1	0	0
0 packets, 0 bytes input			
0 packets, 0 bytes output			
...output omitted...			

In subinterface configuration mode for Fa0/1.200, you execute the command **encapsulation dot1q 200** to change the VLAN association from 20 to 200. Once done, you review the output of **show vlans** on R1, as shown in Example 6-15, to verify that subinterface Fa0/1.200 is associated with VLAN 200.

Example 6-15 *show vlans Command Output on R1 After Configuration Changes*

```
R1#show vlans
...output omitted...
Virtual LAN ID: 200 (IEEE 802.1Q Encapsulation)

VLAN Trunk Interface: FastEthernet0/1.200

Protocols Configured:  Address:          Received:      Transmitted:
IP                    192.168.2.1      0              0

0 packets, 0 bytes input
0 packets, 0 bytes output
```

You then confirm the issue is solved by pinging from PC1 to PC2 again. Example 6-16 shows that the ping is successful, and so you can now conclude that the problem is solved.

Example 6-16 *Successful Ping from PC1 to PC2*

```
C:\PC1>ping 192.168.2.10
Reply from 192.168.2.10: bytes=32 time 1ms TTL=128
Reply from 192.168.2.10: bytes=32 time 1ms TTL=128
Reply from 192.168.2.10: bytes=32 time 1ms TTL=128
Reply from 192.168.2.10: bytes=32 time 1ms TTL=128

Ping statistics for 192.168.2.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Troubleshooting Switched Virtual Interfaces

On a router, an interface has an IP address that defines the subnet the interface is part of. In addition, the IP address is usually acting as a default gateway to hosts residing off of that interface. However, if you have a Layer 3 switch with multiple ports (access or trunk) belonging to the same VLAN, as shown in Figure 6-3, which interface should the IP address be configured on?

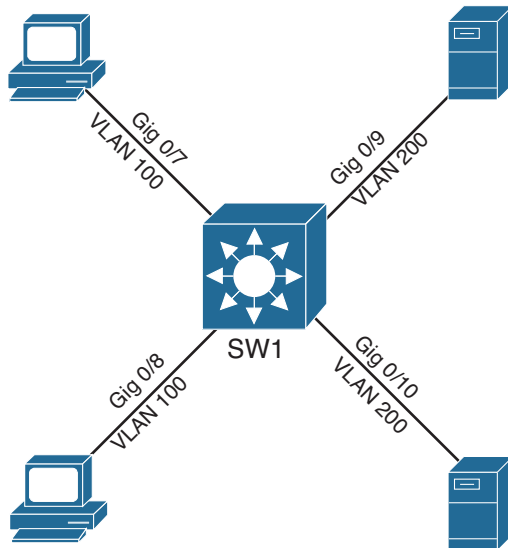


Figure 6-3 Layer 3 Switch Without IP addresses

Since Layer 2 switchports cannot be assigned an IP address; you need to create a logical Layer 3 interface known as a *switched virtual interface* (SVI). These SVIs can be assigned an IP address just like router interfaces. However, unlike router interfaces where an IP address is associated with one interface, the SVI represents all switchports that are part of the same VLAN the SVI is configured for. Therefore, any device connecting to the switch that is in VLAN 100 uses SVI 100, and any device in VLAN 200 uses SVI 200, and so on. This section explains how to configure SVIs on Layer 3 switches and the items that you should look out for when troubleshooting SVIs.

Reviewing SVIs

Figure 6-4 shows a topology using SVIs, and Example 6-17 shows the corresponding configuration. Notice that two SVIs are created: one for each VLAN. The SVI for VLAN 100 has the IP address 192.168.1.1/24, and the SVI for VLAN 200 has the IP address 192.168.2.1/24. Notice that these are two different subnets. As a result, devices that are members of VLAN 100 need to have an IP address in the 192.168.1.0/24 network and

have their default gateway pointing to the VLAN 100 SVI IP address of 192.168.1.1. Devices that are members of VLAN 200 need to have an IP address in the 192.168.2.0/24 network and have their default gateway pointing to the VLAN 200 SVI IP address of 192.168.2.1. An IP address is assigned to an SVI by going into interface configuration mode for a VLAN. For example, the global configuration command **interface vlan 10** enters interface configuration mode for SVI 10 and, if not previously created, will create SVI 10. In this example, because both SVIs are local to the switch, the switch's routing table knows how to forward traffic between members of the two VLANs. Also, IPv4 routing is not on by default on Layer 3 switches; therefore, you need to enable it with the **ip routing** global configuration command.

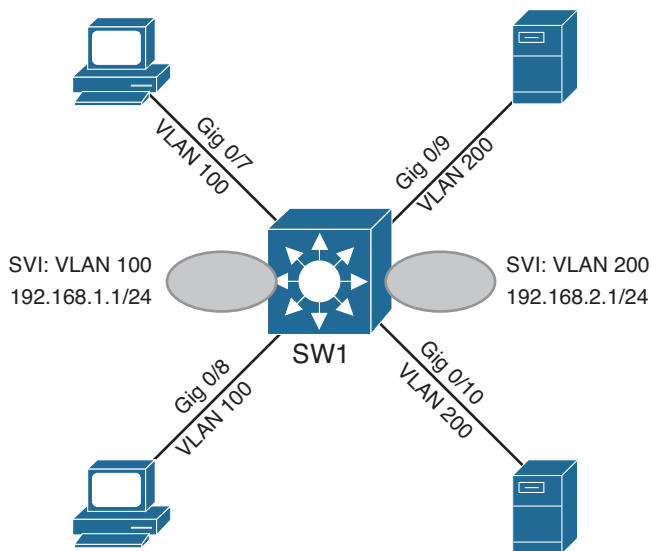


Figure 6-4 Layer 3 Switch with SVIs



Example 6-17 SW1 SVI Configuration

```
SW1#show run
...output omitted...
!
ip routing
!
...output omitted...
!
interface GigabitEthernet0/7
  switchport access vlan 100
  switchport mode access
```

```

!
interface GigabitEthernet0/8
  switchport access vlan 100
  switchport mode access
!
interface GigabitEthernet0/9
  switchport access vlan 200
  switchport mode access
!
interface GigabitEthernet0/10
  switchport access vlan 200
  switchport mode access
!
...output omitted...
!
interface Vlan100
  ip address 192.168.1.1 255.255.255.0
!
interface Vlan200
  ip address 192.168.2.1 255.255.255.0

```

Troubleshooting SVIs

For an SVI to function, the SVI status has to be up and the protocol has to be up. You can verify whether the SVI is up/up with a few different **show** commands, as shown in Example 6-18. In this case, the SVI for VLAN 100 is up/up, as shown in the output of **show ip interface brief**. The output of **show interfaces vlan 100** also displays the SVI as being up/up, but it provides the MAC (bia) address that will be used when devices need to communicate directly with the SVI. For example, when hosts on VLAN 100 need to send a frame to the default gateway (remember the SVI will be the default gateway), they need a destination MAC address for the IP address associated for the SVI. It is this MAC that will be used in this case. The command also provides the IP address of the SVI. Lastly, the **show ip interface vlan 100** command indicates that the SVI is up/up, in addition to providing us with the IP address.



Example 6-18 Verifying the Status of an SVI

SW1#show ip interface brief include Vlan Interface					
Interface	IP-Address	OK?	Method	Status	Protocol
Vlan1	unassigned	YES	NVRAM	administratively down	down
Vlan100	192.168.1.1	YES	manual	up	up
Vlan200	192.168.2.1	YES	manual	up	up

```

SW1#show interfaces vlan 100
Vlan100 is up, line protocol is up
  Hardware is EtherSVI, address is 000d.2829.0200 (bia 000d.2829.0200)
  Internet address is 192.168.1.1/24
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  ...output omitted...

SW1#show ip interface vlan 100
Vlan100 is up, line protocol is up
  Internet address is 192.168.1.1/24
  Broadcast address is 255.255.255.255
  Address determined by setup command
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Outgoing access list is not set
  Inbound access list is not set
  ...output omitted...

```



To successfully troubleshoot SVIs, you need to understand the circumstances that are necessary for an SVI to be up/up. The following list outlines what is needed for an SVI to be up/up:

- The VLAN the SVI is created for needs to exist locally on the switch.
- The SVI has to be enabled and not administratively shut down.
- At a minimum, there must be one switchport (access or trunk) that is up/up and in the spanning-tree forwarding state for that specific VLAN.

Note To route from one SVI to another SVI, IP routing must be enabled on the Layer 3 switch with the **ip routing** command.

SVI Trouble Tickets

This section presents various trouble tickets relating to the topics discussed earlier in the chapter. The purpose of these trouble tickets is to give a process that you can follow when troubleshooting in the real world or in an exam environment. All trouble tickets in this section are based on the topology depicted in Figure 6-5.

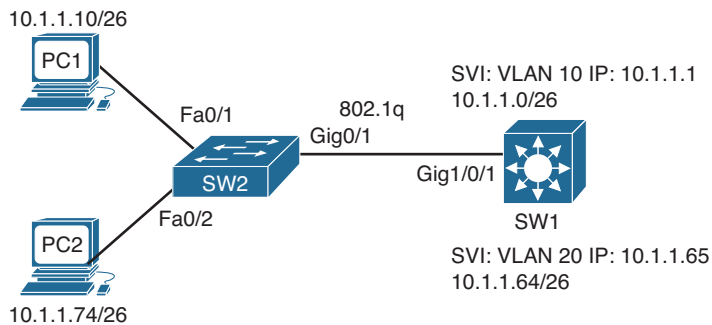


Figure 6-5 *SVI Trouble Ticket Topology*

Trouble Ticket 6-3

Problem: PC1 is not able to access resources on PC2.

Let's start this trouble ticket by verifying the problem. Example 6-19 verifies that PC1 cannot access resources on PC2 because the ping has failed.

Example 6-19 *Failed Ping from PC1 to PC2*

```

C:\PC1>ping 10.1.1.74
Pinging 10.1.1.74 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.1.1.74:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
  
```

Next, we ping the default gateway for PC1, and the result is not successful either, as shown in Example 6-20. This means that we have an issue from PC1 to the default gateway.

Example 6-20 *Failed Ping from PC1 to Default Gateway*

```

C:\PC1>ping 10.1.1.1
Pinging 10.1.1.1 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.1.1.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
  
```

Following a structured troubleshooting approach, you would verify the IP configuration on PC1 as well as its MAC address using the **ipconfig /all** command. Example 6-21 indicates that the IP address, subnet mask, and default gateway are all correct based on Figure 6-5. It also indicates that the MAC address is 0800:275d:06d6.

Example 6-21 *Verifying PC1s Configuration with ipconfig /all*

```
C:\PC1>ipconfig /all
...output omitted...
Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . : 
    Description . . . . . : AMD PCNET Family PCI Ethernet Adapter
    Physical Address. . . . . : 08-00-27-5D-06-D6
    Dhcp Enabled. . . . . : No
    IP Address. . . . . : 10.1.1.10
    Subnet Mask . . . . . : 255.255.255.192
    Default Gateway . . . . . : 10.1.1.1
...output omitted...
```

Next we verify that SW2 is learning the MAC address of PC1 on the correct interface and that it is associated with the correct VLAN. Example 6-22 shows that the MAC address of PC1 (0800:275d:06d6) is associated with Fa0/1 and VLAN 10 with the command **show mac address-table dynamic**.

Example 6-22 *Verifying SW2 Has Learned the MAC Address of PC1 on Fa0/1 and VLAN 10*

```
SW2#show mac address-table dynamic

    Mac Address Table
-----
Vlan    Mac Address      Type      Ports
----    -
10      0800.275d.06d6   DYNAMIC   Fa0/1
20      0800.27a2.ce47   DYNAMIC   Fa0/2
20      2893.fe3a.e342   DYNAMIC   Gi0/1
Total Mac Addresses for this criterion: 3
```

Next we issue the **show mac address-table dynamic** command on SW1, as shown in Example 6-23, to verify that the MAC address of PC1 is being learned on Gig1/0/1 and is associated with VLAN 10. In this case, it is not being learned at all. In addition, reviewing the output of Example 6-22 again concludes that there are no MAC addresses for VLAN 10 being learned on the Gig0/1 interface of SW2. We should see the MAC address of the default gateway for the 10.1.1.0/26 network associated with Gig0/1, but we don't.

Example 6-23 *Verifying SW1 Has Learned the MAC Address of PC1 on Gig1/0/1 and VLAN 10*

```
SW1#show mac address-table dynamic
      Mac Address Table
-----
Vlan    Mac Address      Type    Ports
----    -
20      0800.27a2.ce47    DYNAMIC Gi1/0/1
Total Mac Addresses for this criterion: 1
```

Because SW1 is a Layer 3 switch, it should have an SVI for VLAN 10 with an IP address associated with it in the up/up state. Issuing the command **show ip interface brief | include Vlan10**, as shown in Example 6-24, indicates that the SVI exists on SW1, it has the IP address 10.1.1.1, and it is up/down. Therefore, the issue in this trouble ticket is causing MAC addresses not to be learned for VLAN 10 on SW1's Gig1/0/1 and SW2's Gig0/1 interfaces and is causing the SVI on SW1 to be up/down.

Example 6-24 *Verifying SVI Exists on SW1 and Its Status*

```
SW1#show ip interface brief | include VLAN10|Interface
Interface          IP-Address      OK? Method Status          Protocol
Vlan10             10.1.1.1       YES NVRAM  up              down
```

What causes an SVI's protocol state to be down?

- The VLAN the SVI is created for does not exist locally on the switch.
- The SVI is administratively shut down.
- There is no switchport (access or trunk) that is up/up and in the spanning-tree forwarding state for that specific VLAN.

What would cause MAC addresses not to be learned on trunk interfaces?

- The trunk has mismatched encapsulations, modes, native VLANs.
- The trunk is manually or dynamically pruning traffic for the VLAN causing spanning tree to have no forwarding state for the VLAN.
- The VLAN does not exist on the switch.

Let's compare these two lists. What do they have in common?

- The VLAN does not exist.
- Spanning tree is not in the forwarding state for the VLAN on at least one interface.

On SW1, the **show interfaces trunk** command enables you to see the spanning-tree forwarding state for each VLAN on Gig1/0/1. Example 6-25 shows the output of the command **show interfaces trunk** on SW1 and highlights the fact that SW1 interface Gig1/0/1

is not in the spanning-tree forwarding state for VLAN 10, only for VLAN 1 and 20. If you look further at the output, you see that VLAN 10 is not even listed in the list of VLANs that are active in the management domain. This is a good indication that VLAN 10 does not exist on SW1.

Example 6-25 *Output of show interfaces trunk on SW1*

```
SW1#show interfaces trunk

Port                Mode                Encapsulation  Status        Native vlan
Gi1/0/1             on                  802.1q         trunking      99

Port                Vlans allowed on trunk
Gi1/0/1             1-4094

Port                Vlans allowed and active in management domain
Gi1/0/1             1,20

Port                Vlans in spanning tree forwarding state and not pruned
Gi1/0/1             1,20
```

Reviewing the output of **show vlan brief** on SW1 confirms that VLAN 10 does not exist, as shown in Example 6-26. Correcting this issue requires that you create the VLAN in global configuration mode using the **vlan 10** command.

Example 6-26 *Output of show vlan brief on SW1*

```
SW1#show vlan brief

VLAN Name                Status    Ports
-----
1    default              active    Gi1/0/2, Gi1/0/3, Gi1/0/4
                                           Gi1/0/5, Gi1/0/6, Gi1/0/7
                                           Gi1/0/8, Gi1/0/9, Gi1/0/10
                                           Gi1/0/11, Gi1/0/12, Gi1/0/13
                                           Gi1/0/14, Gi1/0/15, Gi1/0/16
                                           Gi1/0/17, Gi1/0/18, Gi1/0/19
                                           Gi1/0/20, Gi1/0/21, Gi1/0/22
                                           Gi1/0/23, Gi1/0/24, Te1/0/1,
                                           Te1/0/2
20   10.1.1.64/26         active
1002 fddi-default        act/unsup
1003 trcrf-default      act/unsup
1004 fddinet-default     act/unsup
1005 trbrf-default      act/unsup
```

After you have corrected the issue, you want to confirm that the VLAN exists, as shown in the **show vlan brief** output of Example 6-27. You want to confirm that the output of **show interfaces trunk** lists VLAN 10 in the active VLANs in the management domain and that it is in the spanning-tree forwarding state and not pruned for interface Gig1/0/1,

as shown in Example 6-28. In addition, you want to verify that the SVI for VLAN 10 is up/up by using the command **show ip interface brief | include Vlan10**, as shown in Example 6-29.

Example 6-27 *Output of show vlan brief on SW1 After Changes*

```
SW1#show vlan brief
```

VLAN	Name	Status	Ports
1	default	active	Gi1/0/2, Gi1/0/3, Gi1/0/4 Gi1/0/5, Gi1/0/6, Gi1/0/7 Gi1/0/8, Gi1/0/9, Gi1/0/10 Gi1/0/11, Gi1/0/12, Gi1/0/13 Gi1/0/14, Gi1/0/15, Gi1/0/16 Gi1/0/17, Gi1/0/18, Gi1/0/19 Gi1/0/20, Gi1/0/21, Gi1/0/22 Gi1/0/23, Gi1/0/24, Te1/0/1, Te1/0/2
10	10.1.1.0/26	active	
20	10.1.1.64/26	active	
1002	fddi-default	act/unsup	
1003	trcrf-default	act/unsup	
1004	fddinet-default	act/unsup	
1005	trbrf-default	act/unsup	

Example 6-28 *Output of show interfaces trunk on SW1 After Changes*

```
SW1#show interfaces trunk
```

Port	Mode	Encapsulation	Status	Native vlan
Gi1/0/1	on	802.1q	trunking	99

Port	Vlans allowed on trunk
Gi1/0/1	1-4094

Port	Vlans allowed and active in management domain
Gi1/0/1	1,10,20

Port	Vlans in spanning tree forwarding state and not pruned
Gi1/0/1	1,10,20

Example 6-29 *Output of show ip interface brief | include VLAN10 on SW1 After Changes*

```
SW1#show ip interface brief | include Vlan10
```

Interface	IP-Address	OK?	Method	Status	Protocol
Vlan10	10.1.1.1	YES	NVRAM	up	up

Finally, you want to verify that the problem is solved by successfully pinging from PC1 to PC2. Example 6-30 shows that the problem is solved and that the ping is successful.

Example 6-30 *Successful Ping from PC1 to PC2*

```
C:\PC1>ping 10.1.1.74
Reply from 10.1.1.74: bytes=32 time 1ms TTL=128
Reply from 10.1.1.74: bytes=32 time 1ms TTL=128
Reply from 10.1.1.74: bytes=32 time 1ms TTL=128
Reply from 10.1.1.74: bytes=32 time 1ms TTL=128

Ping statistics for 10.1.1.74:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Trouble Ticket 6-4

Problem: PC1 is not able to access resources on PC2.

You start by verifying the problem, as shown in Example 6-31, which confirms (because the ping has failed) that PC1 is unable to access resources on PC2. Next you verify that PC1 can reach the default gateway, as shown in Example 6-32, which it can since the ping was successful. This confirms that no issue exists between PC1 and the default gateway. Next you verify that PC1 can reach the default gateway of VLAN 20, which is 10.1.1.65. Example 6-33 confirms that PC1 is able to reach the default gateway of VLAN 20 since the ping was successful as well.

Example 6-31 *Failed Ping from PC1 to PC2*

```
C:\PC1>ping 10.1.1.74
Pinging 10.1.1.74 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.1.1.74:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Example 6-32 *Successful Ping from PC1 to VLAN 10 Default Gateway*

```
C:\PC1>ping 10.1.1.1
Reply from 10.1.1.1: bytes=32 time 1ms TTL=128
Reply from 10.1.1.1: bytes=32 time 1ms TTL=128
Reply from 10.1.1.1: bytes=32 time 1ms TTL=128
Reply from 10.1.1.1: bytes=32 time 1ms TTL=128
```

```
Ping statistics for 10.1.1.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Example 6-33 *Successful Ping from PC1 to VLAN 20 Default Gateway*

```
PC1#ping 10.1.1.65
Reply from 10.1.1.65: bytes=32 time 1ms TTL=128
Reply from 10.1.1.65: bytes=32 time 1ms TTL=128
Reply from 10.1.1.65: bytes=32 time 1ms TTL=128
Reply from 10.1.1.65: bytes=32 time 1ms TTL=128

Ping statistics for 10.1.1.65:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Because all the pings were successful, this might mean that we have a problem between SW1 and PC2. Let's ping from SW1 to PC2 to verify this. Example 6-34 provides the result of issuing the **ping 10.1.1.74** command on SW1. Notice that the ping is successful, which negates our hypothesis that a problem might exist between SW1 and PC2.

Example 6-34 *Successful Ping from SW1 to PC2*

```
SW1#ping 10.1.1.74
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.74, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/205/1015 ms
```

Let's recap. PC1 can ping SVI 10, and PC2 can ping SVI 20. We also concluded that PC1 can ping SVI 20, which should mean that PC2 can ping SVI 10. Let's double check by pinging from PC2 to the IP address 10.1.1.1. As shown in Example 6-35, it is successful as well.

Example 6-35 *Successful Ping from PC2 to SVI 10*

```
C:\PC2>ping 10.1.1.1
Reply from 10.1.1.1: bytes=32 time 1ms TTL=128
Reply from 10.1.1.1: bytes=32 time 1ms TTL=128
Reply from 10.1.1.1: bytes=32 time 1ms TTL=128
Reply from 10.1.1.1: bytes=32 time 1ms TTL=128

Ping statistics for 10.1.1.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

So, the pings are getting a little more than halfway to their destination. What is required for the ping from PC1 to fully reach PC2? Routing.

Remember how the SVIs work. They are equivalent to router interfaces. Therefore, when you create an SVI, give it an IP address, and it is up/up, an entry for the network that the SVI belongs gets placed in the routing table. Issuing the command **show ip interface brief** on SW1, as shown in Example 6-36, confirms that the SVIs for VLAN 10 and VLAN 20 exist, they have the correct IP addresses assigned to them, and they are up/up.

Example 6-36 *Output of show ip interface brief on SW1*

```
SW1#show ip interface brief | include Vlan|Interface
```

Interface	IP-Address	OK?	Method	Status	Protocol
Vlan1	unassigned	YES	NVRAM	administratively down	down
Vlan10	10.1.1.1	YES	NVRAM	up	up
Vlan20	10.1.1.65	YES	NVRAM	up	up

Let's check the routing table on SW1 with the command **show ip route**. The output of **show ip route**, as shown in Example 6-37, does not even look like a routing table. Therefore, SW1 cannot route traffic. It can only respond to pings that are sent to its local interfaces. The output of Example 6-37 should immediately lead you to the solution of this problem. The problem is that IP routing is not enabled on SW1. By default, on Layer 3 switches, IP routing is disabled. To enable it, you execute the **ip routing** command in global configuration mode.

Example 6-37 *Output of show ip route on SW1*

```
SW1#show ip route
```

Default gateway is not set

Host	Gateway	Last Use	Total Uses	Interface

ICMP redirect cache is empty

After you have enabled IP routing, you can issue the **show ip route** command again on SW1 to verify that directly connected entries have been added to the routing table for SVI VLAN 10 and SVI VLAN 20. Example 6-38 shows a routing table that we are familiar with and the directly connected entries for VLAN 10 and VLAN 20.

Example 6-38 *Output of show ip route on SW1*

```
SW1#show ip route
```

Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
+ - replicated route, % - next hop override

```

Gateway of last resort is not set

    10.0.0.0/8 is variably subnetted, 6 subnets, 3 masks
C       10.1.1.0/26 is directly connected, Vlan10
L       10.1.1.1/32 is directly connected, Vlan10
C       10.1.1.64/26 is directly connected, Vlan20
L       10.1.1.65/32 is directly connected, Vlan20

```

Finally, we need to confirm that our solution solved the original issue, which was that PC1 could not access resources on PC2. Pinging from PC1 to PC2, as shown in Example 6-39, is successful, proving that we solved the issue.

Example 6-39 Successful Ping from PC1 to PC2

```

C:\PC1>ping 10.1.1.74
Reply from 10.1.1.74: bytes=32 time=1ms TTL=128
Reply from 10.1.1.74: bytes=32 time=1ms TTL=128
Reply from 10.1.1.74: bytes=32 time=1ms TTL=128
Reply from 10.1.1.74: bytes=32 time=1ms TTL=128

Ping statistics for 10.1.1.74:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

```

Troubleshooting Routed Ports

Although SVIs can route between VLANs configured on a switch, a Layer 3 switch can be configured to act more as a router (for example, in an environment where you are replacing a router with a Layer 3 switch) by using *routed ports* on the switch. This section explains how to configure routed ports on Layer 3 switches so that you can identify potential problems during the troubleshooting process.

By default, the ports on many Layer 3 Cisco Catalyst switches operate as Layer 2 switchports. Therefore, you have to issue the **no switchport** command in interface configuration mode to convert a switchport to a routed port. Figure 6-6 and Example 6-40 illustrate a Layer 3 switch with its Gigabit Ethernet 0/9 and 0/10 ports configured as routed ports. You can verify whether a port is a routed port by using the **show interfaces interface_type interface_number switchport** command, as shown in Example 6-40 also. A routed port will state Switchport: Disabled.

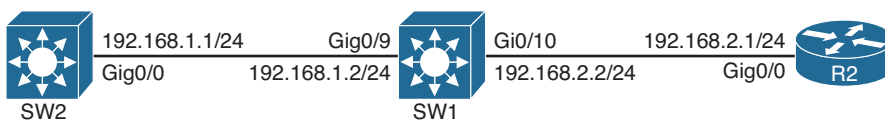


Figure 6-6 Routed Ports on a Layer 3 Switch

**Example 6-40** *Configuration for Routed Ports on a Layer 3 Switch*

```
SW1#show run
...output omitted...
!
interface GigabitEthernet0/9
  no switchport
  ip address 192.168.1.2 255.255.255.0
!
interface GigabitEthernet0/10
  no switchport
  ip address 192.168.2.2 255.255.255.0
!
...output omitted...
SW1#show interfaces gigabitEthernet 0/10 switchport
Name: Gi0/10
Switchport: Disabled
```

The following list outlines the characteristics of routed ports:

- Has no association with any VLAN.
- Physical switchport that has Layer 3 (routing) capabilities.
- Does not run switchport protocols such as Spanning Tree Protocol (STP) or Dynamic Trunking Protocol (DTP).
- Does not support subinterfaces like a router.
- Useful for uplinks between Layer 3 switches or when connecting a Layer 3 switch to a router.
- To route from one routed port to another or a routed port to an SVI and vice versa, IP routing needs to be enabled.

Routed Ports Trouble Tickets

This section presents various trouble tickets relating to the topics discussed earlier in the chapter. The purpose of these trouble tickets is to give a process that you can follow when troubleshooting in the real world or in an exam environment. All trouble tickets in this section are based on the topology depicted in Figure 6-7.

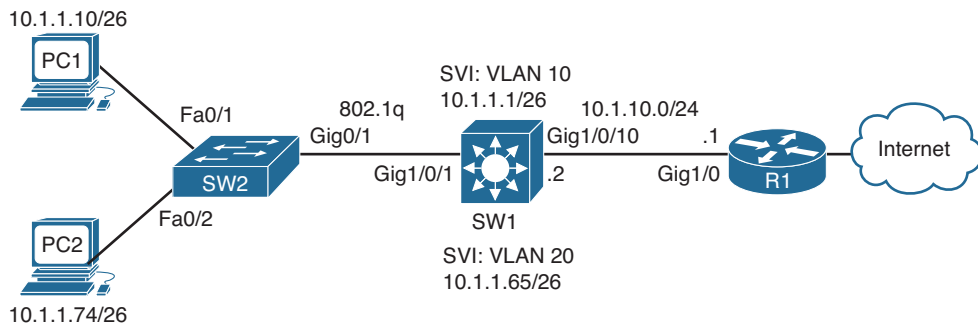


Figure 6-7 *Routed Ports Trouble Tickets Topology*

Trouble Ticket 6-5

Problem: PC1 and PC2 are not able to access resources outside their subnet.

You must always be sure that you fully understand the problem that is being submitted. Therefore, you always need to further define the problem to make sure that it is accurate. You ping from PC1 to the Internet, and it fails. You ping from PC2 to the Internet, and it fails. You ping from PC1 to its default gateway, and it is successful. You ping from PC2 to its default gateway, and it is successful. You ping from PC1 to PC2, and it is successful. Pinging from PC1 and PC2 to R1's Gig1/0 interface fails. Therefore, the problem statement can be changed to read as follows:

Problem: PC1 and PC2 are not able to access resources beyond SW1. They are able to access each other.

This clarification allows us to focus our attention from SW1 onward, skipping all the Layer 2 troubleshooting between the PCs and SW1.

On SW1, you **ping 10.1.10.1**, as shown in Example 6-41, and it fails.

Example 6-41 *Failed Ping from SW1 to R1*

```

SW1#ping 10.1.10.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.10.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
  
```

Next you issue the **show ip interface brief** command on SW1, as shown in Example 6-42, to verify that the correct IP address is configured on interface Gig1/0/10 and that it is up/up. The output shows that there is no IP address configured on Gig1/0/10 and that the interface is up/up.

Example 6-42 *Output of show ip interface brief on SW1*

```
SW1#show ip interface brief
```

Interface	IP-Address	OK?	Method	Status	Protocol
...output omitted...					
GigabitEthernet1/0/9	unassigned	YES	unset	down	down
GigabitEthernet1/0/10	unassigned	YES	unset	up	up
GigabitEthernet1/0/11	unassigned	YES	unset	down	down
...output omitted...					

You enter interface configuration mode for Gig1/0/10 and issue the command **ip address 10.1.10.2 255.255.255.0**, as shown in Example 6-43. You receive the error message displayed in Example 6-43.

Example 6-43 *Error message on SW1*

```
SW1#config t
SW1(config)#interface gig 1/0/10
SW1(config-if)#ip address 10.1.10.2 255.255.255.0
                        ^
% Invalid input detected at '^' marker.
```



As shown in Example 6-43, you are not able to configure an IP address on Gig1/0/10. This is a good indication that it is a Layer 2 switchport. You confirm this by issuing the **show interface Gig1/0/10 switchport** command. The output displayed in Example 6-44 indicates that it is indeed a Layer 2 switchport because the output states Switchport: Enabled. If it stated Switchport: Disabled, this would indicate that it is a routed port.

Example 6-44 *Output of the show interfaces gig1/0/10 switchport Command on SW1*

```
SW1#show interfaces gig1/0/10 switchport
Name: Gil/0/10
Switchport: Enabled
Administrative Mode: dynamic auto
Operational Mode: static access
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: native
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
...output omitted...
```

To assign an IP address to a switchport on a Layer 3 switch, you need to convert it to a routed port using the **no switchport** command in interface configuration mode, as shown in Example 6-45. Also in Example 6-45, you can see that the IP address command was successfully executed after the **no switchport** command was entered.

Example 6-45 *Configuring a Routed Port on SW1*

```
SW1#config t
SW1(config)#interface gig 1/0/10
SW1(config-if)#no switchport
SW1(config-if)#ip address 10.1.10.2 255.255.255.0
```

Now the ping from SW1 to R1 is successful, as displayed in Example 6-46. Also, the pings from PC1 and PC2 to the Internet are successful (not displayed).

Example 6-46 *Successful Ping from SW1 to R1*

```
SW1#ping 10.1.10.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.10.1, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 9/14/17 ms
```

Troubleshooting Layer 3 EtherChannel

Chapter 5 discussed how to troubleshoot Layer 2 EtherChannels between Layer 2 switchports on Cisco Catalyst switches. When you have multiple routed ports on Layer 3 switches, you can bundle them together to create Layer 3 EtherChannels. This section focuses on the Layer 3 EtherChannel requirements and how you can successfully troubleshoot issues relating to it.

An EtherChannel logically combines the bandwidth of multiple physical interfaces into a logical connection between switches, as illustrated in Figure 6-8. Specifically, Figure 6-8 shows four Gigabit Ethernet routed ports logically bonded into a single EtherChannel link known as a *port channel*.

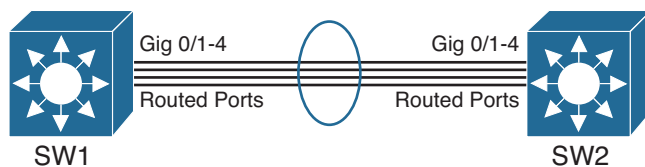


Figure 6-8 *Layer 3 EtherChannel*



Following are common troubleshooting targets to consider when troubleshooting a Layer 3 EtherChannel issue:

- **Mismatched port configurations:** The configurations of all ports making up an EtherChannel, on both switches, should be identical. For example, all ports should have the same speed and duplex and port type (Layer 2 or Layer 3). With Layer 3 EtherChannel, there is no need to worry about trunk mode, native VLAN configurations, and allowed VLAN configurations because we use routed ports, which are Layer 3 ports that do not care about those parameters.

- **Port type during configuration:** Creating an EtherChannel with the **channel-group** command before the port channel is created will automatically create the port channel with the same state as the physical ports bundled in the channel group. For example, if the physical interfaces are Layer 2 switchports, the port channel will be a Layer 2 port channel. If the physical interfaces are Layer 3 interfaces, the port channel will be a Layer 3 port channel. Therefore, it is imperative that you either make the physical interfaces routed ports with the **no switchport** command before creating the bundle or create the Layer 3 port channel with the **interface port-channel interface_number** command and issue the **no switchport** command in interface configuration mode before you configure the physical interfaces with the **channel-group** command. Order of operations is more important with Layer 3 EtherChannel than with Layer 2 EtherChannel.
- **Mismatched EtherChannel configuration:** Both switches forming the EtherChannel should be configured for the same EtherChannel negotiation protocol. The options are Link Aggregation Control Protocol (LACP) and Port Aggregation Protocol (PAgP). If you prefer to statically configure EtherChannel, there is the **on** option as well. Table 6-2 identifies which options can be configured on each switch to successfully form an EtherChannel.
- **Inappropriate EtherChannel distribution algorithm:** EtherChannel determines which physical link to use to transmit frames based on a hash calculation. The hashing approach selected should distribute the load fairly evenly across all physical links. For example, a hash calculation might be based only on the destination MAC address of a frame. If the frames are destined for only a few different MAC addresses, the load distribution could be uneven.

**Table 6-2** *Options for Successfully Forming an EtherChannel*

		SW1				
	MODE	PAgP Desirable	PAgP Auto	LACP Active	LACP Passive	On
SW2	PAgP Desirable	Yes	Yes	No	No	No
	PAgP Auto	Yes	No	No	No	No
	LACP Active	No	No	Yes	Yes	No
	LACP Passive	No	No	Yes	No	No
	On	No	No	No	No	Yes

Verifying an EtherChannel bundle is done with the **show etherchannel summary** command, as shown in Example 6-47. With this output, you can verify the group number, the logical port channel number for the group, the status of the port channel, the protocol that was used, the ports in the bundle, and the status of the ports. In this example, the logical port channel is port channel 1, it is a Layer 3 port channel, and it is in use (as indicated by the RU). This is what you want to see; if you see any other combination, it means that you have a misconfiguration that is preventing the port channel from going up. Link Aggregation Control Protocol (LACP) was used as the protocol in this example, and Gig1/0/5 and 1/0/6 are bundled in the port channel, as indicated by the P. Again, you want to see P listed by the ports; if you see anything else, it means that you have a configuration issue that is preventing the port from being bundled. The only other option I would like to see beside the ports is H, which is used with LACP when you have more than eight ports in the bundle. When you have more than eight ports with LACP, the additional ports are placed in the standby state and used only if one of the main eight go down.

Example 6-47 Output of show etherchannel summary

```
SW1#show etherchannel summary
Flags:  D - down          P - bundled in port-channel
        I - stand-alone  s - suspended
        H - Hot-standby (LACP only)
        R - Layer3       S - Layer2
        U - in use       f - failed to allocate aggregator
        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port

Number of channel-groups in use: 1
Number of aggregators:          1

Group  Port-channel  Protocol    Ports
-----+-----+-----+-----
 1      Po1 (RU)      LACP        Gi1/0/5 (P) Gi1/0/6 (P)
```

Layer 3 EtherChannel Trouble Tickets

This section presents various trouble tickets relating to the topics discussed earlier in the chapter. The purpose of these trouble tickets is to give a process that you can follow when troubleshooting in the real world or in an exam environment. All trouble tickets in this section are based on the topology depicted in Figure 6-9.

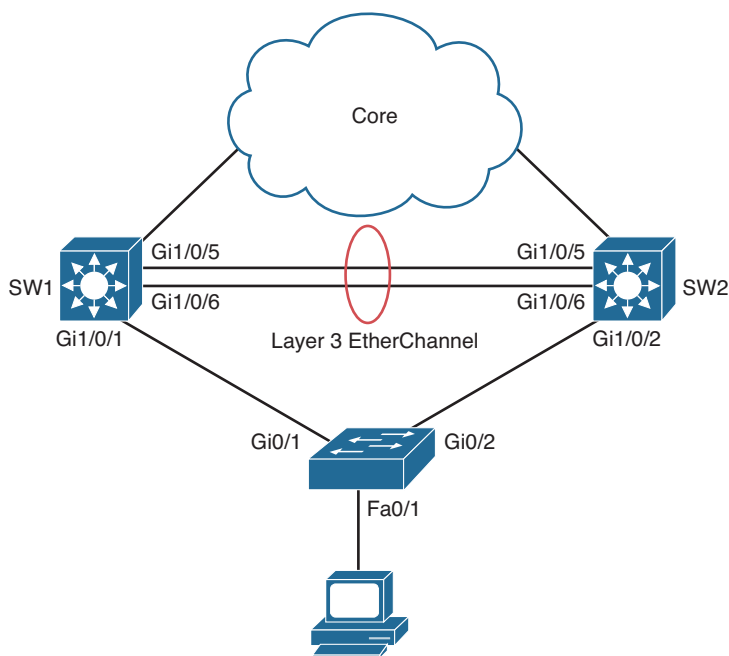


Figure 6-9 *EtherChannel Trouble Tickets Topology*

Trouble Ticket 6-6

Problem: A junior network administrator has approached you indicating that the Layer 3 EtherChannel they are trying to form between SW1 and SW2 is not forming. You need to solve this issue for them.

Your first step is to verify the EtherChannel configuration on SW1 and SW2 using the **show etherchannel summary** command, as shown in Example 6-48 and Example 6-49. Reviewing the flags on SW1 in Example 6-48 indicates that the ports are in standalone and that the port channel is Layer 2 down. Reviewing the flags on SW2 in Example 6-49 indicates that ports are suspended and that the port channel is Layer 3 down. Do you see the issue?

Example 6-48 SW1 show etherchannel summary Output

```
SW1#show etherchannel summary
Flags:  D - down          P - bundled in port-channel
        I - stand-alone  s - suspended
        H - Hot-standby (LACP only)
        R - Layer3       S - Layer2
        U - in use       f - failed to allocate aggregator
        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
```

```

d - default port

Number of channel-groups in use: 1
Number of aggregators:          1

Group  Port-channel  Protocol    Ports
-----+-----+-----+-----
1      Po1(SD)         LACP        Gi1/0/5(I) Gi1/0/6(I)

```

Example 6-49 SW2 show etherchannel summary Output

```

SW2#show etherchannel summary

Flags:  D - down          P - bundled in port-channel
        I - stand-alone  S - suspended
        H - Hot-standby (LACP only)
        R - Layer3       S - Layer2
        U - in use       f - failed to allocate aggregator
        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port

Number of channel-groups in use: 1
Number of aggregators:          1

Group  Port-channel  Protocol    Ports
-----+-----+-----+-----
1      Po1(RD)         LACP        Gi1/0/5(S) Gi1/0/6(S)

```

It appears that our junior network administrator failed to create a Layer 3 EtherChannel on SW1. If you recall, to create a Layer 3 EtherChannel, the physical ports and the port channel must be routed ports. Therefore, the junior network administrator forgot the **no switchport** command on SW1, as shown in Example 6-50.

Example 6-50 SW1 show run interface Output

```

SW1#show run int gig 1/0/5
!
interface GigabitEthernet1/0/5
 switchport trunk encapsulation dot1q
 switchport mode trunk
 switchport nonegotiate
 channel-group 1 mode active
end
SW1#show run int gig 1/0/6
!
interface GigabitEthernet1/0/6

```

```

switchport trunk encapsulation dot1q
switchport mode trunk
switchport nonegotiate
channel-group 1 mode active
end
SW1#show run int port-channel 1
!
interface Port-channel1
end

```

To solve this issue, you need to remove the port channel and channel group configuration from SW1, convert Gig1/0/5 and Gig1/0/6 to routed ports with the **no switchport** command, and then issue the **channel-group mode** command on Gig1/0/5 and Gig1/0/6, which will create the bundle and the Layer 3 port channel. Example 6-51 confirms that the Layer 3 EtherChannel bundle is now formed. Notice how the ports are bundled in the port channel and that the port channel is Layer 3 in use.

Example 6-51 SW1 and SW2 show etherchannel summary Output

```

SW1#show etherchannel summary
Flags:  D - down          P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3        S - Layer2
        U - in use        f - failed to allocate aggregator
        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port

Number of channel-groups in use: 1
Number of aggregators:          1

Group  Port-channel  Protocol    Ports
-----+-----+-----+-----
1      Po1(RU)        LACP       Gi1/0/5(P) Gi1/0/6(P)
!
SW2#show etherchannel summary
Flags:  D - down          P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3        S - Layer2
        U - in use        f - failed to allocate aggregator
        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port

```

```
Number of channel-groups in use: 1
```

```
Number of aggregators: 1
```

```
Group  Port-channel  Protocol  Ports
```

```
-----+-----+-----+-----
```

```
1      Po1 (RU)      LACP      Gi1/0/5 (P)  Gi1/0/6 (P)
```


Exam Preparation Tasks

As mentioned in the section “How to Use This Book” in the Introduction, you have a couple of choices for exam preparation: the exercises here; Chapter 22, “Final Preparation;” and the exam simulation questions on the CD-ROM.

Review All Key Topics

Review the most important topics in this chapter, noted with the Key Topic icon in the outer margin of the page. Table 6-3 lists a reference of these key topics and the page numbers on which each is found.



Table 6-3 *Key Topics for Chapter 6*

Key Topic Element	Description	Page Number
Example 6-1	show run command output from R1	212
Example 6-2	show run command output from SW1	213
List	Describes issues that prevent inter-VLAN routing from functioning with the router-on-a-stick approach	213
Example 6-17	SW1 SVI configuration	222
Example 6-18	Verifying the status of an SVI	223
List	Identifies the elements that must be true for an SVI to be up	224
Example 6-40	Configuration for routed ports on a Layer 3 switch	234
Paragraph	Identifies how to verify whether the port is a Layer 2 switchport or a routed port	236
List	Describes the common Layer 3 EtherChannel troubleshooting targets	237
Table 6-2	Options for successfully forming an EtherChannel	238

Define Key Terms

Define the following key terms from this chapter and check your answers in the glossary:

Layer 3 switch, router-on-a-trunk/router-on-a-stick, switched virtual interface (SVI), routed port, Layer 3 EtherChannel

Complete Tables and Lists from Memory

Print a copy of Appendix C, “Memory Tables,” (found on the disc), or at least the section for this chapter, and complete the tables and lists from memory. Appendix D, “Memory Tables Answer Key,” also on the disc, includes completed tables and lists to check your work.

Show Command Reference to Check Your Memory

This section includes the **show** commands introduced in this chapter. It does not include the **show** commands that were used in this chapter but introduced in previous chapters. You will need to return to the previous chapters to review information relating to those **show** commands.

To test your memory of the commands, cover the right side of Table 6-4 with a piece of paper, read the description on the left side, and then see how much of the command you can remember.

The 300-135 TSHOOT exam focuses on practical, hands-on skills that are used by a networking professional. Therefore, you should be able to identify the show commands needed to successfully troubleshoot the topics presented in this chapter.

Table 6-4 *show Commands Introduced in Chapter 6*

Task	Command Syntax
Displays the VLANs that are associated with a router's subinterfaces, in addition to the trunk encapsulation method used on router's subinterfaces.	show vlans
Displays the Layer 1 and Layer 2 status of an SVI on an MLS along with the IP address, subnet mask, and MAC address associated with it.	show interfaces [vlan {vlan-id}]
If IPv4 routing is enabled on a Layer 3 switch, it displays the contents of the IPv4 routing table.	show ip route
Identifies if a switchport is operating as a Layer 2 switchport or a Layer 3 routed port.	show interfaces interface_type interface_number switchport
Displays the status of port channels, in addition to the status of the ports within the port channel.	show etherchannel summary



This chapter covers the following topics:

- **Troubleshooting Port Security:** This section covers the various reasons why port security might not be performing as expected and how you can troubleshoot them.
- **Port Security Trouble Tickets:** This section provides trouble tickets that demonstrate how you can use a structured troubleshooting process to solve a reported problem.
- **Troubleshooting Spoof-Prevention Features:** This section explains the purpose of DHCP Snooping, Dynamic ARP Inspection, and IP Source Guard. In addition, you will learn what could cause these features not to perform as expected and how to troubleshoot them.
- **Spoof-Prevention Features Trouble Tickets:** This section provides trouble tickets that demonstrate how you can use a structured troubleshooting process to solve a reported problem.
- **Troubleshooting Layer 2 Access Control:** This section examines how to troubleshoot misconfigurations related to protected ports, private VLANs, and VLAN Access Control Lists.

Troubleshooting Switch Security Features

By default, switches are designed to provide connectivity. Therefore, out of the box, minimal security is applied. You can improve switch security by implementing features such as port security, DHCP snooping, dynamic Address Resolution Protocol (ARP) inspection, and IP Source Guard. In addition, by default, all traffic within a VLAN is free to flow between the switchports in the same VLAN. This might not be desired. Therefore, you can control the flow of traffic within the same VLAN with features such as protected ports, private VLANs, and VLAN access control lists (ACLs).

However, with these added features comes additional issues related to them that you will need to be able to troubleshoot. This chapter covers all these features and explores the various reasons why you may be experiencing issues and how you can troubleshoot them.

“Do I Know This Already?” Quiz

The “Do I Know This Already?” quiz allows you to assess whether you should read this entire chapter thoroughly or jump to the “Exam Preparation Tasks” section. If you are in doubt about your answers to these questions or your own assessment of your knowledge of the topics, read the entire chapter. Table 7-1 lists the major headings in this chapter and their corresponding “Do I Know This Already?” quiz questions. You can find the answers in Appendix A, “Answers to the ‘Do I Know This Already?’ Quizzes.”

Table 7-1 “Do I Know This Already?” Section-to-Question Mapping

Foundation Topics Section	Questions
Troubleshooting Port Security	1–3
Troubleshooting Spoof-Prevention Features	4–8
Troubleshooting Layer 2 Access Control	9–10

Caution The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you do not know the answer to a question or are only partially sure of the answer, you should mark that question as wrong for purposes of the self-assessment. Giving yourself credit for an answer that you correctly guess skews your self-assessment results and might provide you with a false sense of security.

1. Which command enables you to verify the port status of a port security-enabled port?
 - a. `show port-security`
 - b. `show port-security interface interface_type interface_number`
 - c. `show port-security address`
 - d. `show running-configuration`
2. Which two of the following port security violation modes will generate a log message when a violation occurs?
 - a. Protect
 - b. Restrict
 - c. Shutdown
 - d. Disabled
3. Which two commands identify the ports that are in the err-disabled state if the err-disable recovery feature has not been enabled for port security?
 - a. `show running-configuration`
 - b. `show interfaces`
 - c. `show interfaces status`
 - d. `show port-security address`
4. What must be true for DHCP snooping to operate successfully? (Choose two.)
 - a. It must be enabled globally.
 - b. It must be enabled for specific VLANs.
 - c. The ports going to end stations must be configured as trusted.
 - d. The ports going to the DHCP servers need to be configured as untrusted.
5. Which command enables you to verify the IP address that has been given to each client from the DHCP server along with the interface they are connected to and the VLAN the interface is a member of?
 - a. `show ip dhcp snooping`
 - b. `show ip dhcp snooping binding`
 - c. `show ip dhcp snooping database`
 - d. `show ip dhcp snooping statistics`

6. What must be true for dynamic ARP inspection to operate successfully? (Choose two answers.)
 - a. DHCP snooping must be enabled globally.
 - b. DHCP snooping must be enabled for specific VLANs.
 - c. IP ARP inspection must be enabled for specific VLANs.
 - d. All interfaces, except for upstream interfaces, need to be configured as trusted interfaces.
7. How does IP Source Guard learn where valid source IPs are in the network?
 - a. ARP cache
 - b. MAC address table
 - c. DHCP snooping database
 - d. Routing table
8. Which command enables you to verify which interfaces have been configured with IP Source Guard?
 - a. `show ip arp`
 - b. `show ip verify source`
 - c. `show interfaces status`
 - d. `show ip dhcp snooping binding`
9. Which two of the following statements are true about PVLANs?
 - a. Community ports cannot communicate with other community ports in the same community.
 - b. Community ports can communicate with other community ports in a different community.
 - c. Community ports cannot communicate with isolated ports and vice versa.
 - d. Isolated ports cannot communicate with other isolated ports.
10. Which of the following has the ability to deny only FTP traffic between two devices in the same VLAN?
 - a. IP Source Guard
 - b. Protected ports
 - c. Private VLANs
 - d. VLAN ACL

Foundation Topics

Troubleshooting Port Security

The port security feature is designed to control a specific set/number of MAC addresses that will be learned on an interface. This helps to eliminate CAM table flooding attacks, where a malicious user attempts to overflow the CAM table by populating it with a large number of bogus MAC addresses. In addition, it ensures that only specific devices (based on MAC address) can connect to certain switchports. Therefore, port security is a must for all organizations to implement. However, as with all services and features, if something goes wrong, you will be troubleshooting. This section shows you how to identify and troubleshoot port security issues.

Common Port Security Issues

Usually, port security will perform as expected with minimal issues. If an attack occurs, port security kicks in; if not, port security keeps waiting. Most issues arise from misconfigurations. The following is a listing of issues that may occur when working with port security:



- Port security is configured but not enabled.
- A static MAC address was not configured correctly.
- The maximum number of MAC addresses has been reached, preventing access.
- Legitimate users are being blocked because of a violation.
- Running configuration not saved to startup configuration.

Port Security Configured but Not Enabled

Example 7-1 provides a port security configuration on interface Fast Ethernet 0/1 of an access layer switch. Notice that all commands start with **switchport port-security**. However, if you fail to include the command **switchport port-security** (which is highlighted), port security is not enabled on the interface regardless of the rest of the configuration specified.

Example 7-1 Sample Port Security Configuration

```
SW1#show running-config interface fastEthernet 0/1
Building configuration...

Current configuration : 456 bytes
!
interface FastEthernet0/1
  switchport access vlan 10
```

```

switchport mode access
switchport port-security maximum 2
switchport port-security
switchport port-security violation restrict
switchport port-security mac-address sticky
switchport port-security mac-address sticky 0050.b607.657a
switchport port-security mac-address 0800.275d.06d6

```

Use the commands **show port-security** and **show port-security interface *interface_type interface_number*** to verify whether port security is enabled on an interface, as shown in Example 7-2. In this case, Fast Ethernet 0/1 is enabled for port security.

Key Topic

Example 7-2 Verifying Port Security Is Enabled on an Interface

```

SW1#show port-security
Secure Port    MaxSecureAddr  CurrentAddr  SecurityViolation  Security Action
      (Count)          (Count)          (Count)
-----
Fa0/1          2              2              0              Restrict
-----

Total Addresses in System (excluding one mac per port)    : 1
Max Addresses limit in System (excluding one mac per port) : 8192

ASW1#show port-security interface fastEthernet 0/1
Port Security      : Enabled
Port Status        : Secure-up
Violation Mode      : Restrict
Aging Time          : 0 mins
Aging Type          : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses : 2
Total MAC Addresses : 2
Configured MAC Addresses : 0
Sticky MAC Addresses : 2
Last Source Address:Vlan : 0800.275d.06d6:10
Security Violation Count : 0

```

Static MAC Address Not Configured Correctly

If you have implemented port security by defining MAC addresses statically, it is imperative that they are accurate. If a user complains that he cannot access the network after receiving a new computer and your network relies on static port security addresses, you more than likely forgot to change the port security static MAC address. Example 7-3 identifies the static MAC address configuration for 0800.275d.06d6.

Example 7-3 *Sample Static MAC Address Port Security Configuration*

```
SW1#show running-config interface fastEthernet 0/1
Building configuration...

Current configuration : 456 bytes
!
interface FastEthernet0/1
  switchport access vlan 10
  switchport mode access
  switchport port-security maximum 2
  switchport port-security
  switchport port-security violation restrict
  switchport port-security mac-address sticky
  switchport port-security mac-address sticky 0050.b607.657a
  switchport port-security mac-address 0800.275d.06d6
```

Using the **show port-security address** command reveals the static MAC address configured for the interfaces, as shown in Example 7-4. In this example, the MAC address 0800.275d.06d6 is a statically configured (SecureConfigured) port security MAC address for Fa0/1 and VLAN 10. You need to compare this to the MAC address of the PC connected to the port with the **ipconfig /all** command, as shown in Example 7-5. (This is where accurate documentation is helpful.) The **show port-security address** command will also identify the dynamically learned port security MAC addresses and the sticky secure MAC addresses.

**Example 7-4** *Verifying Static Addresses Associated with Interfaces*

```
SW1#show port-security address

Secure Mac Address Table

-----
Vlan    Mac Address      Type                Ports    Remaining Age
      (mins)
-----
  10    0050.b607.657a   SecureSticky        Fa0/1    -
  10    0800.275d.06d6   SecureConfigured    Fa0/1    -
-----

Total Addresses in System (excluding one mac per port)    : 1
Max Addresses limit in System (excluding one mac per port) : 8192
```

Example 7-5 *Verifying MAC Address of PC.*

```
PC1#ipconfig /all

Windows IP Configuration

Host Name . . . . . : pc1
Primary Dns Suffix . . . . . :
Node Type . . . . . : Broadcast
```

```

IP Routing Enabled. . . . . : No
WINS Proxy Enabled. . . . . : No

Ethernet adapter PCI Lab:

    Connection-specific DNS Suffix  . :
    Description . . . . . : AMD PCNET Family PCI Ethernet Adapter
    Physical Address. . . . . : 08-00-27-5D-06-D6
    Dhcp Enabled. . . . . : No
...output omitted...

```

Maximum Number of MAC Addresses Reached

By default, when port security is enabled, only one MAC address will be allowed. Therefore, if you need more than one MAC address, you have to specify the number with the **switchport port-security maximum *number*** command, as shown in Example 7-6. In this case, the maximum number was set to 2 so that two devices could communicate through the interface.

Example 7-6 Identifying the Maximum Number of MAC Addresses Allowed

```

SW1#show running-config interface fastEthernet 0/1
Building configuration...

Current configuration : 456 bytes
!
interface FastEthernet0/1
 switchport access vlan 10
 switchport mode access
 switchport port-security maximum 2
 switchport port-security
 switchport port-security violation restrict
 switchport port-security mac-address sticky
 switchport port-security mac-address sticky 0050.b607.657a
 switchport port-security mac-address 0800.275d.06d6

```

You can verify the maximum number of MAC addresses allowed on an interface with the **show port-security** and **show port-security interface *interface_type interface_number*** commands. As shown in Example 7-7, two MACs are allowed, and two have been learned.

Example 7-7 Identifying the Maximum Number of MAC Addresses Allowed

```

SW1#show port-security
Secure Port    MaxSecureAddr  CurrentAddr    SecurityViolation  Security Action
              (Count)      (Count)        (Count)
-----
Fa0/1          2              2              0                  Restrict

```

```
-----
Total Addresses in System (excluding one mac per port)      : 1
Max Addresses limit in System (excluding one mac per port) : 8192
```

```
SW1#show port-security interface fastEthernet 0/1
```

```
Port Security          : Enabled
Port Status            : Secure-up
Violation Mode         : Restrict
Aging Time             : 0 mins
Aging Type             : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses  : 2
Total MAC Addresses    : 2
Configured MAC Addresses : 1
Sticky MAC Addresses   : 1
Last Source Address:Vlan : 0800.275d.06d6:10
Security Violation Count : 0
```

Legitimate Users Being Blocked Because of Violation

You need to make sure that you have the correct number of MAC addresses specified. If the number is not correct, a violation will occur if more than the specified number of MAC addresses are seen on the port. The violation will occur regardless of the additional MAC addresses being accidental or malicious. Three different violations exist:



- **Protect:** Any frame from the MAC addresses in violation is dropped without a notification, and the violation count is not incremented.
- **Restrict:** Any frame from the MAC addresses in violation is dropped, and log messages are generated.
- **Shutdown:** When a violation occurs, the port is placed in the err-disabled state, and any frame from any MAC address will be dropped. In addition, log messages will be generated.

Tip You can remember that these get more severe in alphabetic order (P/R/S) (drop/drop&alert/shutdown&alert).

You can verify whether there is a violation by using the **show port-security** and **show port-security interface *interface_type interface_number*** commands, as shown in Example 7-8. In this case, there is currently no violation. However, if there were, the security violation count would increment, and because the violation mode is Restrict, any frame from the MAC addresses in violation is dropped, and log messages are generated.

Example 7-8 *Identifying Security Violations*

```

SW1#show port-security
Secure Port    MaxSecureAddr  CurrentAddr  SecurityViolation  Security Action
          (Count)          (Count)          (Count)
-----
Fa0/1          2              2              0              Restrict
-----

Total Addresses in System (excluding one mac per port)    : 1
Max Addresses limit in System (excluding one mac per port) : 8192

SW1#show port-security interface fastEthernet 0/1
Port Security          : Enabled
Port Status            : Secure-up
Violation Mode         : Restrict
Aging Time             : 0 mins
Aging Type             : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses  : 2
Total MAC Addresses    : 2
Configured MAC Addresses : 1
Sticky MAC Addresses   : 1
Last Source Address:Vlan : 0800.275d.06d6:10
Security Violation Count : 0

```

If the violation mode is set to **shutdown**, as shown in Example 7-9, and a violation occurs, the port status is *Secure-shutdown* and placed in the *err-disable* state, as displayed in the following syslog messages:

```
%PM-4-ERR_DISABLE: psecure-violation error detected on Fa0/1, putting Fa0/1 in err-disable state
```

```
%PORT_SECURITY-2-PSECURE_VIOLATION: Security violation occurred, caused by MAC address 0800.27a2.ce47 on port FastEthernet0/1.
```

```
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to down
```

```
%LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to down
```

Example 7-9 *Example Port That Has Been Shut Down and Placed in the Err-Disable State*

```

SW1#show port-security interface fastEthernet 0/1
Port Security          : Enabled
Port Status            : Secure-shutdown
Violation Mode         : Shutdown
Aging Time             : 0 mins
Aging Type             : Absolute

```

```
SecureStatic Address Aging : Disabled
Maximum MAC Addresses      : 2
Total MAC Addresses        : 2
Configured MAC Addresses   : 1
Sticky MAC Addresses       : 1
Last Source Address:Vlan   : 0800.27a2.ce47:10
Security Violation Count    : 1
```



To verify ports that are in the err-disabled state, use the command **show interfaces status**, as shown in Example 7-10. You can also use the **show interface *interface_type* *interface_number*** command. As you can see, Fa0/1 is in the err-disabled state. However, it does not tell you what caused the err-disabled state. Example 7-11 displays all the different services that can cause a port to go into the err-disabled state. Notice that they are all enabled by default and that port security is one of them (psecure-violation).

Example 7-10 Identifying Ports in the Err-Disabled State

```
SW1#show interfaces status

Port      Name      Status      Vlan      Duplex  Speed  Type
Fa0/1     err-disabled 10          auto      auto    10/100BaseTX
Fa0/2     connected   10          a-full    a-100   10/100BaseTX
Fa0/3     notconnect  1           auto      auto    10/100BaseTX
Fa0/4     notconnect  1           auto      auto    10/100BaseTX
Fa0/5     notconnect  1           auto      auto    10/100BaseTX
Fa0/6     notconnect  1           auto      auto    10/100BaseTX
...output omitted...

SW1#show interfaces fastEthernet 0/1
FastEthernet0/1 is down, line protocol is down (err-disabled)
Hardware is Fast Ethernet, address is 081f.f34e.b801 (bia 081f.f34e.b801)
```

Example 7-11 Identifying Which Services Are Enabled for Err-Disable

```
SW1#show errdisable detect

ErrDisable Reason      Detection      Mode
-----
arp-inspection          Enabled       port
bpduguard               Enabled       port
channel-misconfig (STP) Enabled       port
community-limit         Enabled       port
dhcp-rate-limit         Enabled       port
dtp-flap                Enabled       port
gbic-invalid            Enabled       port
iif-reg-failure          Enabled       port
inline-power             Enabled       port
invalid-policy           Enabled       port
```

link-flap	Enabled	port
loopback	Enabled	port
lsgroup	Enabled	port
mac-limit	Enabled	port
pagp-flap	Enabled	port
port-mode-failure	Enabled	port
pppoe-ia-rate-limit	Enabled	port
psecure-violation	Enabled	port/vlan
security-violation	Enabled	port
sfp-config-mismatch	Enabled	port
sgacl_limitation	Enabled	port
small-frame	Enabled	port
storm-control	Enabled	port
udld	Enabled	port
vmpls	Enabled	port
psp	Enabled	port



The best way to determine why a port is in the err-disabled state is to review syslog messages. They are listed as severity level 4, and the mnemonic is ERR-DISABLE. In this case, the message text clearly states it was caused by a port security violation.

```
%PM-4-ERR_DISABLE: psecure-violation error detected on Fa0/1, putting Fa0/1 in
err-disable state
```

Tip If for some reason you do not have access to the syslog messages, bounce (shut/noshut) the interface that is err-disabled. By doing so, after the interface is enabled, the error will be detected again, which will generate a syslog message. Make sure that logging to the console or terminal lines is enabled, and do not forget about the terminal monitor command if you are using Telnet or Secure Shell (SSH). This process is shown in Example 7-12, and you can see that the port was err-disabled due to a port security violation.

Example 7-12 Bouncing the Interface to Determine Why It Is Err-Disabled

```
SW1#config t
Enter configuration commands, one per line. End with CNTL/Z.
SW1(config)#interface fastEthernet 0/1
SW1(config-if)#shut
%LINK-5-CHANGED: Interface FastEthernet0/1, changed state to administratively down
SW1(config-if)#no shut
%LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up
%PM-4-ERR_DISABLE: psecure-violation error detected on Fa0/1, putting Fa0/1 in err-
disable state
SW1(config-if)#
%PORT_SECURITY-2-PSECURE_VIOLATION: Security violation occurred, caused by MAC
address 0800.27a2.ce47 on port FastEthernet0/1.
SW1(config-if)#
%LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to down
```



If you are relying on the err-disable recovery feature to enable interfaces once the violation is no longer detected, you can verify the status of the feature with the **show errdisable recovery** command, as shown in Example 7-13. Notice that the err-disable recovery feature is disabled by default for all the different services and features. Therefore, if you need to use it, it has to be manually enabled by you.

Example 7-13 Verifying the Err-Disable Recovery Feature

```
SW1#show errdisable recovery
ErrDisable Reason          Timer Status
-----
arp-inspection             Disabled
bpduguard                 Disabled
channel-misconfig (STP)   Disabled
dhcp-rate-limit           Disabled
dtp-flap                  Disabled
gbic-invalid              Disabled
inline-power              Disabled
link-flap                 Disabled
mac-limit                 Disabled
loopback                  Disabled
pagp-flap                 Disabled
port-mode-failure         Disabled
pppoe-ia-rate-limit       Disabled
psecure-violation         Disabled
security-violation        Disabled
sfp-config-mismatch       Disabled
small-frame               Disabled
storm-control             Disabled
udld                     Disabled
vmps                     Disabled
psp                      Disabled

Timer interval: 300 seconds

Interfaces that will be enabled at the next timeout:
```

To enable err-disable recovery for a specific feature or service, issue the **errdisable recovery cause service/feature** global configuration command, as shown in Example 7-14. This example displays all the different options available on a Catalyst 2960 switch.

Example 7-14 Enabling the Err-Disable Recovery Feature

```
SW1(config)#errdisable recovery cause ?
all                  Enable timer to recover from all error causes
arp-inspection       Enable timer to recover from arp inspection error
                    disable state
bpduguard            Enable timer to recover from BPDU Guard error
```

channel-misconfig (STP)	Enable timer to recover from channel misconfig error
dhcp-rate-limit	Enable timer to recover from dhcp-rate-limit error
dtp-flap	Enable timer to recover from dtp-flap error
gbic-invalid	Enable timer to recover from invalid GBIC error
inline-power	Enable timer to recover from inline-power error
link-flap	Enable timer to recover from link-flap error
loopback	Enable timer to recover from loopback error
mac-limit	Enable timer to recover from mac limit disable state
pagp-flap	Enable timer to recover from pagp-flap error
port-mode-failure	Enable timer to recover from port mode change failure
pppoe-ia-rate-limit	Enable timer to recover from PPPoE IA rate-limit error
psecure-violation	Enable timer to recover from psecure violation error
psp	Enable timer to recover from psp
security-violation	Enable timer to recover from 802.1x violation error
sfp-config-mismatch	Enable timer to recover from SFP config mismatch error
small-frame	Enable timer to recover from small frame error
storm-control	Enable timer to recover from storm-control error
udld	Enable timer to recover from udld error
vmpls	Enable timer to recover from vmpls shutdown error

When using the err-disable recovery feature, you have an extra piece of information you can use. Suppose, for instance, that you enable it for port security. At the bottom of the **show errdisable recovery** output, information identifies what interface is err-disabled and why, as shown in Example 7-15. This makes it easier for you to troubleshoot what caused the port to be err-disabled. It also indicates how much time is left until the port is automatically enabled. If the violation still exists at that point, it will be err-disabled again.

Example 7-15 *Verifying the Err-Disable Reason*

```
SW1#show errdisable recovery
ErrDisable Reason      Timer Status
-----
arp-inspection         Disabled
bpduguard              Disabled
channel-misconfig (STP) Disabled
dhcp-rate-limit        Disabled
dtp-flap               Disabled
gbic-invalid           Disabled
inline-power           Disabled
link-flap              Disabled
mac-limit              Disabled
loopback               Disabled
pagp-flap              Disabled
```



```

port-mode-failure           Disabled
pppoe-ia-rate-limit         Disabled
psecure-violation           Enabled
security-violation          Disabled
sfp-config-mismatch         Disabled
small-frame                 Disabled
storm-control               Disabled
udld                       Disabled
vmps                       Disabled
psp                        Disabled

```

Timer interval: 300 seconds

Interfaces that will be enabled at the next timeout:

Interface	Errdisable reason	Time left(sec)
-----	-----	-----
Fa0/1	psecure-violation	85

Running Configuration Not Saved to Startup Configuration

This is pretty obvious: If you fail to save the running configuration to the NVRAM, the port security configuration will no longer be available when the switch reboots. However, many administrators who use the port security sticky feature forget about saving the configuration when a new PC is added. The sticky feature allows the switch to dynamically learn MAC addresses and then place the MAC address in the configuration just like they had been statically configured. Example 7-16 displays the port security sticky configuration on a switch. Notice how the sticky feature was enabled with the **switchport port-security mac-address sticky** command. Once the MAC address 0050.b607.657a was learned by the switch on interface Fast Ethernet 0/1, the switch placed it in the configuration with the **switchport port-security mac-address sticky 0050.b607.657a** command. You now need to save the configuration; otherwise, the sticky-learned MAC address will not be in the configuration if the switch reboots.

Example 7-16 Port Security Sticky Configuration

```

SW1#show running-config interface fastEthernet 0/1
Building configuration...

Current configuration : 456 bytes
!
interface FastEthernet0/1
  switchport access vlan 10
  switchport mode access
  switchport port-security maximum 2
  switchport port-security

```

```

switchport port-security violation restrict
switchport port-security mac-address sticky
switchport port-security mac-address sticky 0050.b607.657a
switchport port-security mac-address 0800.275d.06d6

```

Port Security Trouble Tickets

This section presents various trouble tickets relating to the topics discussed earlier in the chapter. The purpose of these trouble tickets is to give a process that you can follow when troubleshooting in the real world or in an exam environment. All trouble tickets in this section are based on the topology depicted in Figure 7-1.

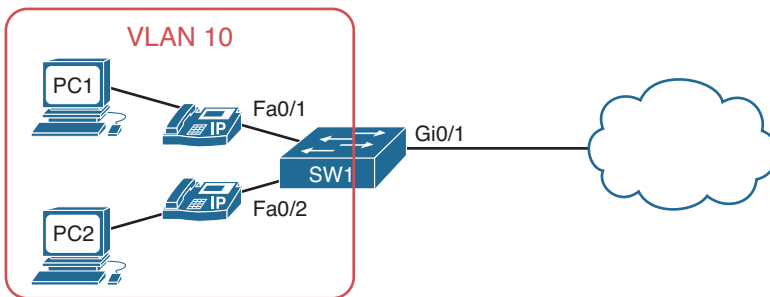


Figure 7-1 Port Security Trouble Ticket Topology

Trouble Ticket 7-1

Problem: It is Monday morning, and the user on PC1 has called you indicating that she is not able to access any network resources.

You ask her when the last time it was that she was able to access resources. She indicates that it was 2 weeks ago, before she went on vacation. This leads you to examine the change control documentation to determine whether any configuration changes were done in the past 2 weeks. You notice that port security was added to all access ports on SW1. Therefore, you decide to start your troubleshooting process by examining the port security configuration on SW1.

According to documentation, PC1 is connected to Fa0/1. You issue the command **show port-security**, as shown in Example 7-17, and notice that Fa0/1 is enabled for port security and that there is a security violation count of 1.

Example 7-17 Verifying Port Security on Fa0/1

```

SW1#show port-security

```

Secure Port	MaxSecureAddr (Count)	CurrentAddr (Count)	SecurityViolation (Count)	Security Action
Fa0/1	2	2	1	Shutdown
Fa0/2	2	2	0	Shutdown

Fa0/3	2	0	0	Shutdown
Fa0/4	2	0	0	Shutdown
Fa0/5	2	0	0	Shutdown
Fa0/6	2	0	0	Shutdown
Fa0/7	2	0	0	Shutdown
Fa0/8	2	0	0	Shutdown
Fa0/9	2	0	0	Shutdown
Fa0/10	2	0	0	Shutdown
Fa0/11	2	0	0	Shutdown
Fa0/12	2	0	0	Shutdown
Fa0/13	2	0	0	Shutdown
Fa0/14	2	0	0	Shutdown
Fa0/15	2	0	0	Shutdown
Fa0/16	2	0	0	Shutdown
Fa0/17	2	0	0	Shutdown
Fa0/18	2	0	0	Shutdown
Fa0/19	2	0	0	Shutdown
Fa0/20	2	0	0	Shutdown
Fa0/21	2	0	0	Shutdown
Fa0/22	2	0	0	Shutdown
Fa0/23	2	0	0	Shutdown
Fa0/24	2	0	0	Shutdown

Total Addresses in System (excluding one mac per port) : 2
 Max Addresses limit in System (excluding one mac per port) : 8192

To verify the status of port security for Fa0/1 you issue the command **show port-security interface fastEthernet 0/1**, as shown in Example 7-18. Port security is enabled but it is in the *Secure-shutdown* state. The last MAC address that was received on the interface was 0800.275d.06d6 for VLAN 10.

Example 7-18 Verifying Port Security Status on Fa0/1

```
SW1#show port-security interface fastEthernet 0/1
Port Security          : Enabled
Port Status            : Secure-shutdown
Violation Mode         : Shutdown
Aging Time             : 0 mins
Aging Type             : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses  : 2
Total MAC Addresses    : 2
Configured MAC Addresses : 2
Sticky MAC Addresses   : 0
Last Source Address:Vlan : 0800.275d.06d6:10
Security Violation Count : 1
```

Next you issue the **show run interface fa0/1** command to verify the port security configuration on Fa0/1. As shown in Example 7-19, it has been enabled, the maximum MAC addresses is set to 2, and there are 2 MAC addresses configured (one for the phone and one for PC1).

Example 7-19 *Verifying Port Security Configuration on Fa0/1*

```
SW1#show run interface fa0/1
Building configuration...

Current configuration : 352 bytes
!
interface FastEthernet0/1
  switchport access vlan 10
  switchport mode access
  switchport port-security maximum 2
  switchport port-security
  switchport port-security mac-address 0050.b607.657a
  switchport port-security mac-address 0800.275d.06d7
  no lldp transmit
  spanning-tree portfast
  spanning-tree bpduguard enable
  spanning-tree guard root
end
```

You decide to confirm the MAC addresses of the IP Phone and PC1. Starting with the PC, you issue the **ipconfig /all** command, as shown in Example 7-20. The MAC address of PC1 is 08-00-27-5D-06-D6, which happens to be the same MAC address that caused the violation shown in Example 7-18. Comparing the MAC address of PC1 to the addresses statically configured on Fa0/1, as shown in Example 7-19, confirms that PC1's MAC address is not one of the addresses configured.

Example 7-20 *Reviewing the MAC Address on PC1*

```
C:\>ipconfig /all

Windows IP Configuration

Host Name . . . . . : pc1
Primary Dns Suffix . . . . . :
Node Type . . . . . : Broadcast
IP Routing Enabled. . . . . : No
WINS Proxy Enabled. . . . . : No

Ethernet adapter PC1 Lab:

Connection-specific DNS Suffix . :
Description . . . . . : AMD PCNET Family PCI Ethernet Adapter
```

```
Physical Address. . . . . : 08-00-27-5D-06-D6
Dhcp Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . : Yes
Autoconfiguration IP Address. . . : 169.254.180.166
Subnet Mask . . . . . : 255.255.0.0
...output omitted...
```

After confirming that the IP Phone's MAC address is 0050.b607657a, you conclude that the command **switchport port-security mac-address 0050.b607657a** is correct but that the command **switchport port-security mac-address 0800.275d.06d7** is not correct. It appears that the static MAC address was misconfigured with a 7 at the end rather than a 6.

You proceed to remove the incorrect static MAC address with the **no switchport port-security mac-address 0800.275d.06d7** command and replace it with the MAC address of PC1. Example 7-21 provides the configuration that is needed to solve the issue.

Example 7-21 *Solving the Issue by Configuring the Correct Static MAC Address*

```
SW1#config t
Enter configuration commands, one per line. End with CNTL/Z.
SW1(config)#interface fastEthernet 0/1
SW1(config-if)#no switchport port-security mac-address 0800.275d.06d7
SW1(config-if)#switchport port-security mac-address 0800.275d.06d6
```

You confirm the port is still in the err-disabled state with the **show interfaces status** command. The output shown in Example 7-22 confirms it is. To recover from the err-disabled state, you bounce the interface by issuing the **shutdown** and then **no shutdown** commands.

Example 7-22 *Confirming Fa0/1 is in the Err-Disabled State*

```
SW1#show interfaces status
```

Port	Name	Status	Vlan	Duplex	Speed	Type
Fa0/1		err-disabled	10	auto	auto	10/100BaseTX
Fa0/2		connected	10	a-full	a-100	10/100BaseTX
Fa0/3		notconnect	1	auto	auto	10/100BaseTX
Fa0/4		notconnect	1	auto	auto	10/100BaseTX

...output omitted...

The interface successfully goes up/up, and you receive the following syslog messages:

```
%LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to up
```

You confirm the problem is solved by accessing PC1 and pinging the default gateway at 10.1.1.1. It is successful, as shown in Example 7-23. The issue has been solved.

Example 7-23 *Successful Ping from PC1 to Default Gateway*

```

C:\>ping 10.1.1.1

Reply from 10.1.1.1: bytes=32 time 1ms TTL=128
Reply from 10.1.1.1: bytes=32 time 1ms TTL=128
Reply from 10.1.1.1: bytes=32 time 1ms TTL=128
Reply from 10.1.1.1: bytes=32 time 1ms TTL=128

Ping statistics for 10.1.1.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

```

Troubleshooting Spoof-Prevention Features

Features such as DHCP snooping, dynamic ARP inspection, and IP Source Guard are designed to protect your network from spoofing attacks against the Dynamic Host Configuration Protocol (DHCP) service, ARP, and IP addressing. This section explains what you should look for while troubleshooting these three security features.

DHCP Snooping

To prevent rogue DHCP servers from handing out IP addresses in your network, you can implement DHCP snooping. With DHCP snooping, you can define which interfaces will accept all DHCP messages and which interfaces will accept only Discover and Request DHCP messages. DHCP snooping also creates a binding table that keeps track of which devices are connected to which interfaces based on the IP addresses that were handed out by the DHCP server. This comes in handy with DAI and IP Source Guard, as you will see later.

Take a moment to examine Example 7-24, which displays a sample DHCP snooping configuration. What is required for DHCP snooping to operate successfully? Let's make a list:



- DHCP snooping is enabled globally with the **ip dhcp snooping** command.
- DHCP snooping is enabled for specific VLANs with the **ip dhcp snooping vlan** command.
- Interfaces that need to accept all DHCP message types are configured as trusted with the **ip dhcp snooping trust** command.
- All other interfaces need to be untrusted, which is the default.
- If the DHCP server does not support option 82 it needs to be disabled on the switch with the **no ip dhcp snooping information option** command.

Example 7-24 *Sample DHCP Snooping Configuration*

```
SW1#show run
...output omitted...
ip dhcp snooping vlan 10
no ip dhcp snooping information option
ip dhcp snooping
...output omitted...
interface GigabitEthernet0/1
  ip dhcp snooping trust
interface GigabitEthernet0/2
  ip dhcp snooping trust
...output omitted...
```

To verify DHCP snooping, use the **show ip dhcp snooping** command, as shown in Example 7-25. You can verify whether it is enabled globally with the line that states *Switch DHCP snooping is enabled*. You can verify which VLANs are enabled and operational for DHCP snooping. In this case, it is only VLAN 10. You can verify whether option 82 is enabled or disabled. Finally, you can verify which interfaces are trusted, which interfaces are untrusted, and which interfaces have a DHCP rate limit applied. In this case, Gigabit Ethernet 0/1 and 0/2 are trusted interfaces, and all other interfaces that are not listed are automatically untrusted.

**Example 7-25** *Verifying DHCP Snooping*

```
SW1#show ip dhcp snooping
Switch DHCP snooping is enabled
DHCP snooping is configured on following VLANs:
10
DHCP snooping is operational on following VLANs:
10
DHCP snooping is configured on the following L3 Interfaces:

Insertion of option 82 is disabled
  circuit-id default format: vlan-mod-port
  remote-id: 081f.f34e.b800 (MAC)
Option 82 on untrusted port is not allowed
Verification of hwaddr field is enabled
Verification of giaddr field is enabled
DHCP snooping trust/rate is configured on the following Interfaces:
```

Interface	Trusted	Allow option	Rate limit (pps)
-----	-----	-----	-----
GigabitEthernet0/1	yes	yes	unlimited
Custom circuit-ids:			
GigabitEthernet0/2	yes	yes	unlimited
Custom circuit-ids:			

To verify the bindings in the DHCP snooping database, issue the **show ip dhcp snooping bindings** command, as shown in Example 7-26. In this example, the PC with the MAC address 08:00:27:5D:06:D6 is located out Fast Ethernet 0/1, which is part of VLAN 10, and has been assigned the IP address 10.1.1.10 from a DHCP server.

**Key
Topic**

Example 7-26 *Verifying DHCP Snooping Bindings*

```
SW1#show ip dhcp snooping binding
```

MacAddress	IpAddress	Lease(sec)	Type	VLAN	Interface
-----	-----	-----	-----	----	-----
--					
08:00:27:5D:06:D6	10.1.1.10	67720	dhcp-snooping	10	FastEthernet0/1
Total number of bindings: 1					

Dynamic ARP Inspection

Dynamic ARP inspection (DAI) is used to prevent ARP spoofing attacks. It relies on DHCP snooping and the binding table that is created by it. Because of this, you need to be able to troubleshoot DHCP snooping issues when dealing with DAI issues. In addition, you have to be able to troubleshoot the commands related to DAI. Refer to Example 7-27. For DAI to function, it needs to be enabled per VLAN with the **ip arp inspection vlan** command. In addition, interfaces where DAI should not be performed (where there are no DHCP snooping bindings) need to be configured as trusted interfaces with the **ip arp inspection trust** command.

**Key
Topic**

Example 7-27 *Sample DAI Configuration*

```
SW1#show run
...output omitted...
ip dhcp snooping vlan 10
ip arp inspection vlan 10
no ip dhcp snooping information option
ip dhcp snooping
...output omitted...
interface GigabitEthernet0/1
 ip dhcp snooping trust
 ip arp inspection trust
interface GigabitEthernet0/2
 ip dhcp snooping trust
 ip arp inspection trust
...output omitted...
```

When DAI detects an invalid ARP request or response on an untrusted interface it will generate syslog messages with a severity level of 4 with the mnemonic of *DHCP_SNOOPING_DENY*. This is because DAI relies on the DHCP snooping binding table to identify appropriate IP address to MAC address bindings. In these syslog messages

a device with the IP address 10.1.1.10 and a MAC of 0050.b607.657a is being denied because its ARPs are invalid since the addresses do not match the addresses in the binding table.

```
%SW_DAI-4-DHCP_SNOOPING_DENY: 1 Invalid ARPs (Req) on Fa0/1, vlan 10.([0050.
b607.657a/10.1.1.10/2893.fe3a.e345/10.1.1.1/18:42:55 UTC Mon Mar 1 1993])
```

```
%SW_DAI-4-DHCP_SNOOPING_DENY: 1 Invalid ARPs (Res) on Fa0/1, vlan 10.([0050.
b607.657a/10.1.1.10/2893.fe3a.e345/10.1.1.1/18:43:15 UTC Mon Mar 1 1993])
```

IP Source Guard

IP Source Guard is used to prevent IP address spoofing. It relies on DHCP snooping and the binding table that is created by it. Because of this, you need to be able to troubleshoot DHCP snooping issues when dealing with IP Source Guard issues. In addition, you have to be able to identify issues related to IP Source Guard configurations. Notice in Example 7-28 that the same DHCP snooping configuration example is listed; however, on interface Fast Ethernet 0/1 (which connects to an end station), the **ip verify source** command has been added. This enables IP Source Guard on the interface.

Example 7-28 Sample IP Source Guard Configuration

```
SW1#show run
...output omitted...
ip dhcp snooping vlan 10
no ip dhcp snooping information option
ip dhcp snooping
...output omitted...
interface FastEthernet0/1
  ip verify source
interface GigabitEthernet0/1
  ip dhcp snooping trust
interface GigabitEthernet0/2
  ip dhcp snooping trust
...output omitted...
```



You can verify which interfaces have IP Source Guard enabled with the **show ip verify source** command, as shown in Example 7-29. In this case, Fa0/1 on SW1 has been enabled with IP Source Guard, and the packets with the source IP address 10.1.1.10 are the only ones allowed inbound on interface Fa0/1.

Notice how the Mac-address column is blank and the Filter-type is IP. With the **ip verify source** command, you are filtering based on IP address only. If you want to include the MAC address with the IP address when verifying the source of packets, you issue the **ip verify source port-security** command. In Example 7-30, you can see that the MAC address is included now and the filter type is ip-mac.

Example 7-29 *Verifying IP Source Guard (only IP)*

```
SW1#show ip verify source
```

Interface	Filter-type	Filter-mode	IP-address	Mac-address	Vlan
-----	-----	-----	-----	-----	----
Fa0/1	ip	active	10.1.1.10		10

Example 7-30 *Verifying IP Source Guard (IP and MAC)*

```
SW1#show ip verify source
```

Interface	Filter-type	Filter-mode	IP-address	Mac-address	Vlan
-----	-----	-----	-----	-----	----
Fa0/1	ip-mac	active	10.1.1.10	08:00:27:5D:06:D6	10

If you are using the *ip-mac* filter type, you need to have port security enabled on the interface, because the secure MAC addresses are used. If port security is not enabled, the specific MAC address will not be learned, and all MAC addresses will be permitted as a result, as shown in Example 7-31.

Example 7-31 *IP MAC Filtering Without Port Security Enabled on Interface*

```
SW1#show ip verify source
```

Interface	Filter-type	Filter-mode	IP-address	Mac-address	Vlan
-----	-----	-----	-----	-----	----
Fa0/1	ip-mac	active	10.1.1.10	08:00:27:5D:06:D6	10
Fa0/2	ip-mac	active	10.1.1.20	permit-all	10

Also, remember that IP Source Guard relies on DHCP snooping. Therefore, if there is no binding in the DHCP snooping database for the port, all traffic will be blocked for all IPs, as shown in Example 7-32. In this example, there is no DHCP snooping binding for Fa0/2 because it has a static IP configured. However, IP Source Guard is enabled on the interface. Because IP Source Guard relies on DHCP snooping and there is no binding in the table, all ingress traffic on Fa0/2 will be denied.

Example 7-32 *Fa0/2 Sourced Traffic Denied Because There Is No Binding*

```
SW1#show ip verify source
```

Interface	Filter-type	Filter-mode	IP-address	Mac-address	Vlan
-----	-----	-----	-----	-----	----
Fa0/1	ip-mac	active	10.1.1.10	08:00:27:5D:06:D6	10
Fa0/2	ip-mac	active	deny-all	permit-all	10


```
SW1#show ip dhcp snooping binding
```

MacAddress	IpAddress	Lease (sec)	Type	VLAN	Interface
-----	-----	-----	-----	----	-----
--					
08:00:27:5D:06:D6	10.1.1.10	70453	dhcp-snooping	10	FastEther-
net0/1					
Total number of bindings: 1					

Spoof-Prevention Features Trouble Tickets

This section presents various trouble tickets relating to the topics discussed earlier in the chapter. The purpose of these trouble tickets is to give a process that you can follow when troubleshooting in the real world or in an exam environment. All trouble tickets in this section are based on the topology depicted in Figure 7-2.

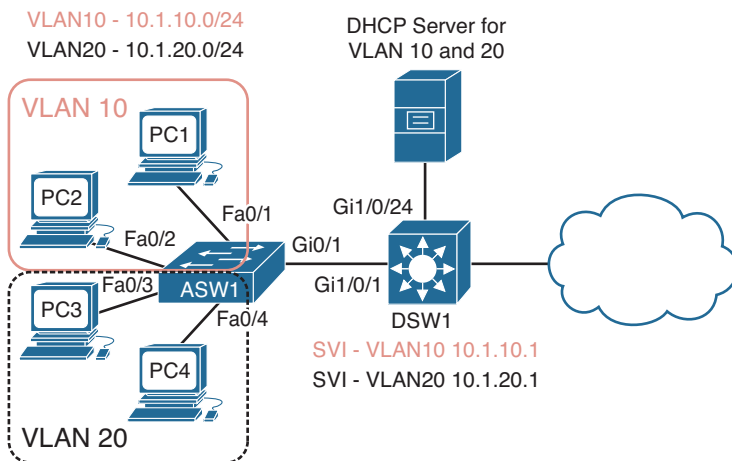


Figure 7-2 *Spoof-Prevention Features Trouble Ticket Topology*

Trouble Ticket 7-2

Problem: A junior administrator has approached you for assistance with a trouble ticket that she is having an issue with. The trouble ticket indicates that users in VLAN 10 are not able to access any resources outside their own subnet. They have verified that the clients receive their IP addressing information via a DHCP server. However, they are confused as to why they would be receiving the default gateway address of 10.1.10.100 when documentation shows that the default gateway should be configured as 10.1.10.1. They also indicate that they verified the DHCP pool on the DHCP server and that the default gateway address for the VLAN 10 pool is configured for 10.1.10.1.

To assist with the issue, you decide to connect your laptop to Fast Ethernet 0/24 on ASW1. This is the port on ASW1 that is used as the Switched Port Analyzer (SPAN) destination port. You configure ASW1, as shown in Example 7-33, so that all traffic sent or received by Fa0/1 is captured and sent to Fa0/24, where your laptop is connected and running packet-capturing software.

Example 7-33 *Configuring a SPAN Session on ASW1*

```
ASW1#config t
Enter configuration commands, one per line. End with CNTL/Z.
ASW1(config)#monitor session 1 source interface fastEthernet 0/1 both
ASW1(config)#monitor session 1 destination interface fastEthernet 0/24
```

You access PC1 and issue the **ipconfig /renew** command to trigger the DHCP process so that you can identify who is providing the IP addressing. The DHCP packets between the server and PC1 are successfully copied by SPAN to your laptop running packet-capturing software, which is connected to Fa0/24.

You review the DHCP offer message in your packet-capture software and notice that it is sourced from IP 10.1.10.34 and MAC 28:93:fe:3a:e3:45. Using the **show mac address-table dynamic address 28:93:fe:3a:e3:45** command to follow the path, as shown in Example 7-34, you verify that the device with that MAC address is reachable out Fa0/17, which is part of VLAN 10. You review your network documentation and trace the port to a PC that is being used for study purposes by an employee that currently enabled DHCP and just happened to use the same network that VLAN 10 is using in the production network. You ask the employee to disable the DHCP server, and she does.

Example 7-34 *Renewing a DHCP Address*

```
ASW1#show mac address-table dynamic address 28:93:fe:3a:e3:45
Mac Address Table
-----
Vlan    Mac Address      Type    Ports
----    -
10      28:93:fe:3a:e3:45 DYNAMIC Fa0/17
Total Mac Addresses for this criterion: 1
```

The issue is solved. To update all the client PCs, you issue the **ipconfig /renew** command on all of them. They receive the correct default gateway of 10.1.10.1 now. However, you decide to dig deeper. Your network is configured with DHCP snooping, DAI, and IP Source Guard. As a result, this issue should have never happened. You decide to issue the **show ip dhcp snooping** command on ASW1 to verify the DHCP snooping configuration, as shown in Example 7-35. Based on the output, DHCP snooping is enabled globally, it is enabled for VLAN 20, information option 82 is disabled, and Gig0/1 is trusted. You have identified the problem. DHCP snooping has not been enabled for VLAN 10. Therefore, the DHCP server that was configured on Fa0/17 is able to hand out DHCP addresses on the network. By your enabling of DHCP snooping for VLAN 10, Fa0/17 would become an untrusted port by default and prevent DHCP Offer and Acks from being accepted inbound.

Example 7-35 *Reviewing the DHCP Snooping Configuration*

```
ASW1#show ip dhcp snooping
Switch DHCP snooping is enabled
DHCP snooping is configured on following VLANs:
20
DHCP snooping is operational on following VLANs:
20
DHCP snooping is configured on the following L3 Interfaces:
```

```

Insertion of option 82 is disabled
  circuit-id default format: vlan-mod-port
  remote-id: 001c.57fe.f600 (MAC)
Option 82 on untrusted port is not allowed
Verification of hwaddr field is enabled
Verification of giaddr field is enabled
DHCP snooping trust/rate is configured on the following Interfaces:

Interface                Trusted    Allow option    Rate limit (pps)
-----
GigabitEthernet0/1      yes      yes             unlimited
Custom circuit-ids:

```

To fix the DHCP snooping configuration, you issue the **ip dhcp snooping vlan 10** command in global configuration mode, as shown in Example 7-36.

Example 7-36 Configuring DHCP Snooping for VLAN 10

```

ASW1#config t
Enter configuration commands, one per line. End with CNTL/Z.
ASW1(config)#ip dhcp snooping vlan 10

```

You verify the configuration with the **show ip dhcp snooping** command again and confirm that VLAN 10 is now enabled for DHCP snooping, as shown in Example 7-37.

Example 7-37 Verifying DHCP Snooping Is Enabled for VLAN 10

```

ASW1#show ip dhcp snooping
Switch DHCP snooping is enabled
DHCP snooping is configured on following VLANs:
10,20
DHCP snooping is operational on following VLANs:
10,20
DHCP snooping is configured on the following L3 Interfaces:

Insertion of option 82 is disabled
  circuit-id default format: vlan-mod-port
  remote-id: 001c.57fe.f600 (MAC)
Option 82 on untrusted port is not allowed
Verification of hwaddr field is enabled
Verification of giaddr field is enabled
DHCP snooping trust/rate is configured on the following Interfaces:

Interface                Trusted    Allow option    Rate limit (pps)
-----
GigabitEthernet0/1      yes      yes             unlimited
Custom circuit-ids:

```

Troubleshooting Access Control

Access control between devices within the same VLAN/subnet can be implemented using features such as protected ports, private VLANs, and VLAN access control lists (VACLs). Because the devices are in the same VLAN/subnet that you are trying to filter traffic to or from, regular router-based ACLs that are applied to router interfaces will not filter this traffic. This is because that traffic is never sent to the router interface. It stays within the local subnet/VLAN between the Layer 2 switchports.

This section explains what is involved when troubleshooting issues related to protected ports, private VLANs, and VACLs, which are used to filter traffic between devices within the same subnet/VLAN.

Protected Ports

Key Topic

The purpose of a protected port is to deny all traffic from flowing between devices connected to two interfaces in the same VLAN on the same switch. Therefore, when troubleshooting protected ports, you are usually dealing with the following issues:

- Traffic is flowing between two interfaces when it should not be.
- Traffic is not flowing between two interfaces when it should be.

When dealing with protected ports, both these issues would be the result of a misconfiguration. Keep in mind that a protected port can only communicate with ports that are not protected ports. If traffic arrives inbound on a protected port, it will not be forwarded if the egress port is also a protected port. Therefore, if two devices are able to communicate when they should not, it might be because one port is a protected port and the other is not a protected port when it should be.

Figure 7-3 displays an access layer switch with PC1 and PC2 connected to it on Fa0/1 and Fa0/2. Both ports are members of VLAN 10. However, for security reasons, traffic is not allowed to flow between Fa0/1 and Fa0/2. Example 7-38 displays the interface configuration command **switchport protected** that is used to configure the ports as protected.

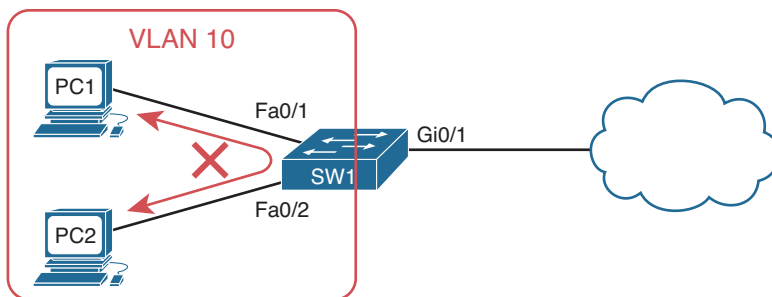


Figure 7-3 Protected Ports

Example 7-38 *Sample Protected Port Configuration*

```

SW1#show run interface fastEthernet 0/1
...output omitted...
interface FastEthernet0/1
  switchport access vlan 10
  switchport mode access
  switchport protected
end

SW1#show run interface fastEthernet 0/2
...output omitted...
interface FastEthernet0/2
  switchport access vlan 10
  switchport mode access
  switchport protected
end

```

Besides using the running configuration to verify protected ports, you can use the command **show interfaces interface_type interface_number switchport** to verify whether a port is configured as a protected port, as shown in Example 7-39. In the output for Fa0/1, it states *Protected: true*, which means Fa0/1 is a protected port.

Example 7-39 *Verifying Protected Ports*

```

SW1#show interfaces fastEthernet 0/1 switchport
Name: Fa0/1
Switchport: Enabled
Administrative Mode: static access
Operational Mode: static access
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: native
Negotiation of Trunking: Off
Access Mode VLAN: 10 (10.1.1.0/26)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: enabled
Voice VLAN: none
Administrative private-vlan host-association: none
Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: none
Administrative private-vlan trunk Native VLAN tagging: enabled
Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: none
Administrative private-vlan trunk associations: none
Administrative private-vlan trunk mappings: none
Operational private-vlan: none
Trunking VLANs Enabled: ALL

```

```

Pruning VLANs Enabled: 2-1001
Capture Mode Disabled
Capture VLANs Allowed: ALL

Protected: true
Unknown unicast blocked: disabled
Unknown multicast blocked: disabled
Appliance trust: none

```

Private VLANs

Private VLANs (PVLAN) take the protected port concept further by enabling you to control which ports in the same VLAN can communicate with each other and which ports cannot. This is accomplished by grouping ports together in secondary VLANs that are members of a Private VLAN. Just like protected ports, when troubleshooting PVLANS, you are usually dealing with the following issues:

- Traffic is flowing between two interfaces when it should not be.
- Traffic is not flowing between two interfaces when it should be.

When dealing with PVLANS, both these issues would be the result of a misconfiguration. Refer to Figure 7-4, which will be used for our PVLAN examples. DNS1 and DNS2 are in the secondary community VLAN of 501, which is within the primary VLAN 200. FS1 and FS2 are in the secondary isolated VLAN 502, which is within the primary VLAN 200. Therefore, based on the rules of PVLANS, the following are true:

- DNS1 and DNS2 are able to communicate with each other because they are members of the same community VLAN.
- DNS1 and DNS2 are not able to communicate with FS1 and FS2 because DNS1 and DNS2 are members of a community VLAN and FS1 and FS2 are members of an isolated VLAN.
- FS1 and FS2 are not able to communicate with each other because they are members of an Isolated VLAN.
- DNS1, DNS2, FS1, and FS2 are able to communicate out to the cloud because Gi1/0/10 is the promiscuous port.

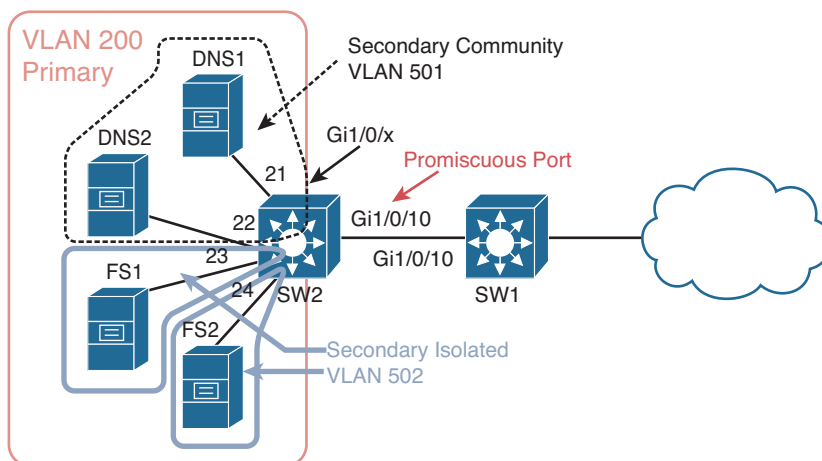


Figure 7-4 PVLANs

To successfully troubleshoot PVLANs, you need to remember the following PVLAN rules:



- Community ports can communicate with other community ports in the same community.
- Community ports cannot communicate with other community ports in a different community.
- Community ports cannot communicate with isolated ports and vice versa.
- Isolated ports cannot communicate with other isolated ports.
- Community and isolated ports can communicate with the promiscuous port.

Example 7-40 displays the commands required to successfully implement the PVLANs in Figure 7-4. First, unless you are using Virtual Trunking Protocol (VTP) Version 3, the VTP mode has to be transparent or off. VTP Versions 1 and 2 cannot carry PVLAN information like VTPv3. The primary VLAN needs to be identified with the **private-vlan primary** command and associated with the secondary VLANs with the **private-vlan association** command. In addition, the secondary community VLAN needs to be identified with the **private-vlan community** command, and the secondary isolated VLAN needs to be identified with the **private-vlan isolated** command. After the VLANs have been identified, you can associate the ports on the switch with the appropriate VLANs. In this example, Gig1/0/10 is the promiscuous port for the secondary VLANs 501 and 502 that are mapped to the primary VLAN 200, as identified by the commands **switchport private-vlan mapping 200 501-502** and **switchport mode private-vlan promiscuous**. To associate a port with a secondary VLAN, you use the **switchport private-vlan host-association primary_vlan secondary_vlan** command in interface configuration mode along with the command **switchport mode private-vlan host**. The only way to determine from this output that the interface is in the correct secondary VLAN is to examine the **switchport private-vlan host-association primary_vlan secondary_vlan** command and compare the secondary VLAN ID to the VLAN configuration information.

For example, if you compare the secondary VLAN ID of 502 in the command **switchport private-vlan host-association 200 502** of interface Gig1/0/23 with the VLAN 502 configuration, you will notice that VLAN 502 is an isolated VLAN.



Example 7-40 PVLAN Configuration Example

```
SW2#show run
...output omitted...
!
vtp mode transparent
!
vlan 200
  private-vlan primary
  private-vlan association 501-502
!
vlan 501
  private-vlan community
!
vlan 502
  private-vlan isolated
!
...output omitted...
!
interface GigabitEthernet1/0/10
  switchport private-vlan mapping 200 501-502
  switchport mode private-vlan promiscuous
!
...output omitted...
!
interface GigabitEthernet1/0/21
  switchport private-vlan host-association 200 501
  switchport mode private-vlan host
!
interface GigabitEthernet1/0/22
  switchport private-vlan host-association 200 501
  switchport mode private-vlan host
!
interface GigabitEthernet1/0/23
  switchport private-vlan host-association 200 502
  switchport mode private-vlan host
!
interface GigabitEthernet1/0/24
  switchport private-vlan host-association 200 502
  switchport mode private-vlan host
!
...output omitted...
end
```

As you can see, with all the different parameters, it is very easy to misconfigure PVLANS. Therefore, it is imperative that you can read a PVLAN configuration, compare it to a topological diagram, and determine where the misconfiguration is that is causing traffic to be forwarded to ports it should not be forwarded to or causing traffic to not be forwarded to ports it should be forwarded to.

In addition, you can verify the private VLANs and the ports associated with each private VLAN using the **show vlan private-vlan** command, as shown in Example 7-41. You can see in this output the primary VLAN 200 and its associated community VLAN 501 and isolated VLAN 502. The ports associated with the community VLAN are Gi1/0/10, Gi1/0/21, and Gi1/0/22. The ports associated with the isolated VLAN are Gi1/0/10, Gi1/0/23, and Gi1/0/24. The first port, Gi1/0/10, is the promiscuous port in both cases.



Example 7-41 Verifying Private VLANs and Associated Ports

```
SW2#show vlan private-vlan
```

Primary	Secondary	Type	Ports
200	501	community	Gi1/0/10, Gi1/0/21, Gi1/0/22
200	502	isolated	Gi1/0/10, Gi1/0/23, Gi1/0/24

You can also use the command **show interfaces interface_type interface_number switchport** to verify the PVLAN status and configuration of a specific interface. As shown in Example 7-42, the administrative mode and operational mode is *private-vlan host*, indicating that it is either a member of a community vlan or isolated vlan. If it stated *private-vlan promiscuous*, it is the promiscuous port. The primary VLAN in this case is VLAN 200, as indicated by the line *Access Mode VLAN: 200 (primary)*. Further down, you can see the host association, which indicates that the primary VLAN is VLAN 200 and that this specific port is a member of the secondary VLAN 501. In addition, the *Operational private-vlan* output states the same.

Example 7-42 Verifying Private VLAN Information for a Specific Port

```
SW2#show interfaces gigabitEthernet 1/0/22 switchport
```

```
Name: Gi1/0/22
Switchport: Enabled
Administrative Mode: private-vlan host
Operational Mode: private-vlan host
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: native
Negotiation of Trunking: Off
Access Mode VLAN: 200 (primary)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: enabled
Voice VLAN: none
Administrative private-vlan host-association: 200 (10.1.200.0/24) 501 (VLAN0501)
Administrative private-vlan mapping: none
```

```

Administrative private-vlan trunk native VLAN: none
Administrative private-vlan trunk Native VLAN tagging: enabled
Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: none
Administrative private-vlan trunk associations: none
Administrative private-vlan trunk mappings: none
Operational private-vlan:
  200 (10.1.200.0/24) 501 (VLAN0501)
...output omitted...

```

VACLs

Protected ports and PVLANS are excellent features that help you control the traffic that can flow between ports in the same subnet/VLAN. However, they lack granular control. Therefore, it is all traffic or no traffic that is being forwarded between the ports. You cannot pick which type of traffic to control. If you do need to control the type of traffic that is flowing between ports in the same VLAN/subnet on a switch, you can implement VLAN access control lists (VACLs). Because you are able to control traffic on a more granular level, when troubleshooting VACLs you need to examine a few different components that make up the VACL:



- **ACLs:** Used to define the traffic that will be examined by the VLAN access map (IP or MAC). Use the **show access-lists** command to verify the configured ACLs.
- **VLAN access map:** Used to define the action that will be taken on the traffic that is matched in the ACLs. Use the **show run | section vlan access-map** command or the **show vlan access-map** command to verify the configured VLAN access maps.
- **VLAN filter list:** Used to define which VLANs the VLAN access map will apply to. Use the **show run | include vlan filter** command or the **show vlan filter** command to verify the configured VLAN filter list.

Refer to the sample VACL in Example 7-43, which was used to configure SW1 in Figure 7-5. This VACL is designed to prevent PC1 from being able to ping or telnet to PC2, which is in the same VLAN. However, PC1 will be able to access other resources and services on PC2. Notice all the different configurations that could cause the VACL to not function as expected.



- **The ACL could be misconfigured:** Permit versus deny, wrong protocol, wrong addresses, wrong ports.
- **The VLAN access map could be in the wrong sequence order:** Just like an ACL, route map, and prefix list, it uses top-down processing, will immediately execute the actions upon a match, and there is an implicit **deny** all at the end.
- **The VLAN access map could be misconfigured:** Matching the wrong ACL, the action could be incorrect, such as drop versus forward.

- **The VLAN filter could be misconfigured:** The filter may be referencing the wrong VLAN access map, it could be configured with the wrong VLAN list, or it may be missing completely.

Example 7-43 Sample VLAN ACL Configuration

```
SW1#show access-lists
Extended IP access list 100
    10 permit icmp host 10.1.1.10 host 10.1.1.20
    20 permit tcp host 10.1.1.10 host 10.1.1.20 eq telnet

SW1#show run | section vlan access-map
vlan access-map TSHOOT 10
    match ip address 100
    action drop
vlan access-map TSHOOT 20
    action forward

SW1#show run | include vlan filter
vlan filter TSHOOT vlan-list 10
```

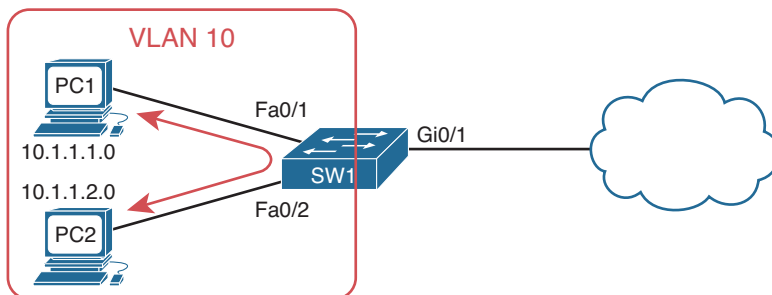


Figure 7-5 VACL

Exam Preparation Tasks

As mentioned in the section “How to Use This Book” in the Introduction, you have a couple of choices for exam preparation: the exercises here; Chapter 22, “Final Preparation;” and the exam simulation questions on the CD-ROM.

Review All Key Topics

Review the most important topics in this chapter, noted with the Key Topic icon in the outer margin of the page. Table 7-2 lists a reference of these key topics and the page numbers on which each is found.



Table 7-2 *Key Topics for Chapter 7*

Key Topic Element	Description	Page Number
List	Identifies issues that may be the reason why port security is not behaving as expected	250
Example 7-2	Verifying port security	251
Example 7-4	Verifying static addresses associated with interfaces	252
List	Outlines the different port security violation modes	254
Paragraph	Describes how to verify a port is in the err-disable state	256
Paragraph	Describes how to determine why a port is in the err-disable state and provides a valuable tip	257
Paragraph	Describes the error disable recovery feature and the commands used for verification purposes	258
List	Provides a listing of items that must be true for DHCP snooping to operate correctly	265
Example 7-25	Verifying DHCP snooping	266
Example 7-26	Verifying DHCP snooping bindings	267
Example 7-27	Sample DAI configuration	267
Paragraph	Describes how to verify that IP Source Guard has been configured correctly	268
Section	Protected ports	273
List	Outlines the PVLAN rules that are required when troubleshooting PVLANS	276
Example 7-40	PVLAN configuration example	277
Example 7-41	Verifying Private VLANs and associated ports	278

Key Topic Element	Description	Page Number
List	Identifies the components involved with VACLs that you may have to troubleshoot	279
List	Identifies what could be misconfigured with a VACL that could be causing issues	279

Define Key Terms

Define the following key terms from this chapter and check your answers in the glossary:

port security, protect violation mode, restrict violation mode, shutdown violation mode, err-disabled, sticky secure MAC address, DCHP snooping, DHCP snooping (trusted port), DHCP snooping (untrusted port), dynamic ARP inspection, IP Source Guard, protected ports, private VLANs, primary VLAN, community VLAN, isolated VLAN, promiscuous port, VLAN access control list

Command Reference to Check Your Memory

This section includes the most important **show** commands covered in this chapter. It might not be necessary to memorize the complete syntax of every command, but you should be able to remember the basic keywords that are needed.

To test your memory of the commands, cover the right side of Table 7-3 with a piece of paper, read the description on the left side, and then see how much of the command you can remember.

The 300-135 TSHOOT exam focuses on practical, hands-on skills that are used by a networking professional. Therefore, you should be able to identify the commands needed to successfully verify and troubleshoot the topics covered within this chapter.

Table 7-3 *show Commands Used for Verification and Troubleshooting*

Task	Command Syntax
Displays the ports that have port security enabled, the maximum number of MAC addresses allowed, the current number learned, whether there is a security violation, and the action that is taken if a violation occurs.	show port-security

Task	Command Syntax
Displays the secure MAC addresses that have been learned on each port security enabled port. It displays the port and associated VLAN, the MAC address, and the type (SecureDynamic, SecureSticky, and SecureConfigured).	<code>show port-security address</code>
Displays detailed port security information for the interface. It identifies whether port security is enabled or disabled, the port security status, the violation mode that is configured, and the aging type and time. It also displays the maximum max addresses allowed, the current number of MAC addresses, the number of statically configured addresses, the number of sticky addresses, and whether a violation has occurred. In addition, it displays the last seen MAC on the port, which is helpful for troubleshooting.	<code>show port-security interface <i>interface_type</i> <i>interface_number</i></code>
Displays the configuration within the running configuration for a specific interface. You can verify configurations related to port security, DHCP snooping, DAI, IP Source Guard, protected ports, and PVLANS.	<code>show running-config interface <i>interface_type</i> <i>interface_number</i></code>
Displays the Layer 1 and Layer 2 status of an interface. Also helps identify which ports are in the err-disable state.	<code>show interface status</code>
Displays which features are able to use the error disable recovery feature on the switch and the mode they will use.	<code>show errdisable detect</code>
Displays which features are enabled and disabled for the error disable recovery feature, the timer that has been set, and any ports that are currently in the err-disable state (along with the reason why).	<code>show errdisable recovery</code>
Displays the status of DHCP snooping, including whether it is enabled or disabled globally, the VLANs it is enabled for, whether option 82 is enabled or disabled, and the trusted ports.	<code>show ip dhcp snooping</code>
Displays the MAC address to IP address DHCP snooping mappings, along with the port and VLAN they are mapped to.	<code>show ip dhcp snooping binding</code>

Task	Command Syntax
Displays the interfaces that have been enabled with IP Source Guard, the filter type being used, along with the IP address, MAC address, and VLAN number that source packets and frames will need to match.	show ip verify source
Displays VLAN, trunking, PVLAN, and protected port information related to an interface.	show interfaces <i>interface_type</i> <i>interface_number</i> switchport
Displays the primary and secondary PVLAN mappings along with the member interfaces.	show vlan private-vlan
Displays all access lists, including IP and MAC, that are configured on the switch.	show access-list
Displays the VLAN access map configuration on the switch.	show run section vlan access-map show vlan access-map
Displays the VLAN access map to VLAN mapping on the switch.	show run include vlan filter show vlan filter

This page intentionally left blank



This chapter covers the following topics:

- **Troubleshooting HSRP:** This section focuses on the Cisco Hot Standby Router Protocol (HSRP). It reviews the HSRP features and functions and how you can verify HSRP configurations and troubleshoot HSRP issues.
- **HSRP Trouble Tickets:** This section provides trouble tickets that demonstrate how you can use a structured troubleshooting process to solve a reported problem.
- **Troubleshooting VRRP:** This section focuses on the industry standard Virtual Router Redundancy Protocol (VRRP). It reviews the VRRP features and functions as well as how you can verify VRRP configurations and troubleshoot VRRP issues.
- **VRRP Trouble Tickets:** This section provides trouble tickets that demonstrate how you can use a structured troubleshooting process to solve a reported problem.
- **Troubleshooting GLBP:** This section focuses on the Cisco Gateway Load Balancing Protocol (GLBP). It reviews the GLBP features and functions and how you can verify GLBP configurations and troubleshoot GLBP issues.
- **GLBP Trouble Tickets:** This section provides trouble tickets that demonstrate how you can use a structured troubleshooting process to solve a reported problem.
- **Comparing HSRP, VRRP, and GLBP:** This section provides a close-up comparison of the different first-hop redundancy protocols (FHRPs) covered in the chapter.

Troubleshooting First-Hop Redundancy Protocols

Many devices, such as PCs, are configured with a default gateway. The default gateway parameter identifies the IP address of a next-hop router on the local-area network (LAN) that serves as the exit point for the LAN. As a result, if that router were to become unavailable, devices that relied on the default gateway's IP address would be unable to send traffic off their local subnet.

Fortunately, Cisco devices such as routers and Layer 3 switches offer technologies known as first-hop redundancy protocols (FHRPs) that provide next-hop gateway redundancy. These technologies include HSRP, VRRP, and GLBP, which allow clients to continue to reach their default gateway's IP address, even if the Layer 3 switch or router that had been servicing that IP address becomes unavailable.

This chapter reviews HSRP, VRRP, and GLBP, and provides a collection of Cisco IOS commands you can use to troubleshoot issues related to them.

“Do I Know This Already?” Quiz

The “Do I Know This Already?” quiz allows you to assess whether you should read this entire chapter thoroughly or jump to the “Exam Preparation Tasks” section. If you are in doubt about your answers to these questions or your own assessment of your knowledge of the topics, read the entire chapter. Table 8-1 lists the major headings in this chapter and their corresponding “Do I Know This Already?” quiz questions. You can find the answers in Appendix A, “Answers to the ‘Do I Know This Already?’ Quizzes.”

Table 8-1 “Do I Know This Already?” Section-to-Question Mapping

Foundation Topics Section	Questions
Troubleshooting HSRP	1–4
Troubleshooting VRRP	5–6
Troubleshooting GLBP	7–9
Comparing HSRP, VRRP, GLBP	10

Caution The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you do not know the answer to a question or are only partially sure of the answer, you should mark that question as wrong for purposes of the self-assessment. Giving yourself credit for an answer that you correctly guess skews your self-assessment results and might provide you with a false sense of security.

1. What is the default priority for an HSRP interface?
 - a. 0
 - b. 100
 - c. 256
 - d. 32768
2. How many active forwarders can be in an HSRP group?
 - a. 1
 - b. 2
 - c. 4
 - d. No limit
3. What command enables you to verify the virtual MAC address of an HSRP group?
 - a. `show hsrp`
 - b. `show hsrp brief`
 - c. `show standby`
 - d. `show standby brief`
4. Which two of the following are true about HSRP?
 - a. Preemption is on by default.
 - b. Preemption is off by default.
 - c. The virtual router IP address can be an unused IP in the LAN or an IP associated with a router's LAN interface.
 - d. The virtual router IP address has to be an unused IP in the LAN.
5. What is the name for the router in a VRRP virtual router group that is actively forwarding traffic on behalf of the virtual router group?
 - a. Virtual forwarder
 - b. Active virtual gateway
 - c. Virtual router master
 - d. Active virtual forwarder

6. Which two of the following are true about VRRP? (Choose two answers.)
 - a. Preemption is on by default.
 - b. Preemption is off by default.
 - c. The virtual router IP address can be an unused IP in the LAN or an IP associated with a router's LAN interface.
 - d. The virtual router IP address has to be an unused IP in the LAN.
7. Which **show** commands enable you to verify the virtual MAC addresses that an AVF is responsible for? (Choose two answers.)
 - a. **show run**
 - b. **show arp**
 - c. **show glbp**
 - d. **show glbp brief**
8. Which of the following is the default GLBP method for load balancing?
 - a. Weighted
 - b. Host dependent
 - c. Server dependent
 - d. Round-robin
9. Which of the following statements is true concerning GLBP?
 - a. GLBP is an industry-standard FHRP.
 - b. GLBP allows multiple routers to simultaneously forward traffic.
 - c. The active virtual forwarder in a GLBP group is responsible for responding to ARP requests with different MAC addresses.
 - d. A GLBP group has multiple active virtual gateways.
10. Which of the following are Cisco proprietary FHRPs? (Choose two answers.)
 - a. HSRP
 - b. VRRP
 - c. GLBP
 - d. IRDP

Foundation Topics

Troubleshooting HSRP

Hot Standby Router Protocol (HSRP) is a Cisco Proprietary FHRP that was designed to provide default gateway redundancy. HSRP operates on both Cisco routers and Cisco multilayer switches. When implemented, it allows multiple physical layer 3 gateways to appear as a single virtual layer 3 gateway. It is this virtual layer 3 gateway that the clients point to as their default gateway.

As a troubleshooter you will need to have a very solid understanding of how HSRP functions in order to resolve any issues related to HSRP. In this section you will review the concepts of HSRP as well as how to verify and troubleshoot HSRP configurations.

Reviewing HSRP

HSRP uses a virtual IP address and MAC address to represent a virtual router within an HSRP group. The end-stations' default gateway IP address is the IP address of the virtual router. When the end-stations ARP for the MAC address of the default gateway IP address, they are given the virtual MAC address. Under no circumstances should the end-stations ever be given the real MAC address of the device that is acting as the default gateway when they are ARPing for the MAC of the virtual IP address.

Within an HSRP group, one router is the *active router*. This router is responsible for forwarding data sent to the MAC address of the default gateway and responding to ARP requests asking for the MAC associated with the IP address of the default gateway. Another router in the HSRP group is known as the *standby router*. This router is waiting for the active router to fail or experience a link/reachability failure so that it can take over the active router role and forward traffic and respond to ARP requests. You can have additional routers in an HSRP group, but they will not be active or standby. They will simply sit and wait for the active or standby to fail so they can elect a replacement among them. Figure 8-1 illustrates a basic HSRP topology.

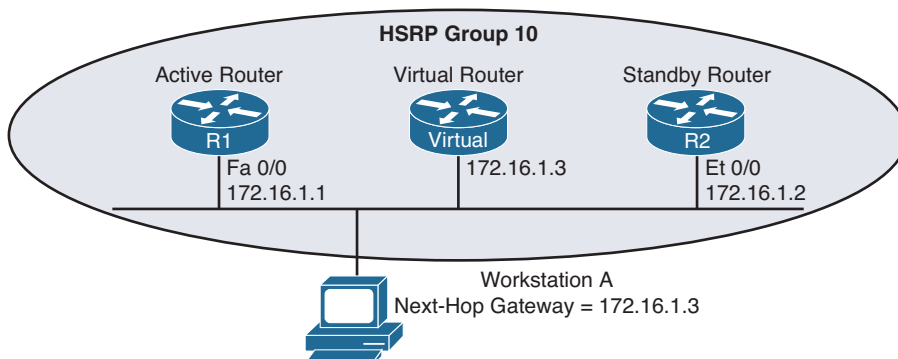


Figure 8-1 Basic HSRP Operation

Examples 8-1 and 8-2 show the HSRP configuration for routers R1 and R2.

Example 8-1 HSRP Configuration on Router R1

```
R1#show run
...OUTPUT OMITTED...
interface FastEthernet0/0
 ip address 172.16.1.1 255.255.255.0
 standby 10 ip 172.16.1.3
 standby 10 priority 150
 standby 10 preempt
...OUTPUT OMITTED...
```

Example 8-2 HSRP Configuration on Router R2

```
R2#show run
...OUTPUT OMITTED...
interface Ethernet0/0
 ip address 172.16.1.2 255.255.255.0
 standby 10 ip 172.16.1.3
...OUTPUT OMITTED...
```



Notice that both routers R1 and R2 have been configured with the same virtual IP address of 172.16.1.3 for an HSRP group of 10. Router R1 is configured with a higher priority using the **standby 10 priority 150** command. Router R2 has a default HSRP priority of 100 for group 10, and with HSRP, higher priority values are more preferable. Also, notice that router R1 is configured with the **standby 10 preempt** command, which means that if router R1 loses its active status, perhaps because it is powered off, it will regain its active status when it again becomes available.

HSRP Converging After a Failure

By default, HSRP sends hello messages every three seconds. Also, if the standby router does not hear a hello message within ten seconds by default, the standby router considers the active router to be down. The standby router then assumes the active role.

Although this ten-second convergence time applies for a router becoming unavailable for a reason such as a power outage or a link failure, convergence happens more rapidly if an interface is administratively shut down. Specifically, an active router sends a *resign* message if its active HSRP interface is shut down.

Also, consider the addition of another router to the network segment whose HSRP priority for group 10 is higher than 150. If it were configured for preemption, the newly added router would send a *coup* message, to inform the active router that the newly added router was going to take on the active role. If, however, the newly added router were not configured for preemption, the currently active router would remain the active router.

HSRP Verification and Troubleshooting

When verifying an HSRP configuration or troubleshooting an HSRP issue, you should begin by determining the following information about the HSRP group under inspection:



- Which router is the active router?
- Which routers, if any, are configured with the preempt option?
- What is the virtual IP address?
- What is the virtual MAC address?
- Is interface or object tracking on?

The **show standby brief** command can be used to show which interface is participating in an HSRP group. It identifies the HSRP group number, the priority of the interface, and if preemption is enabled or not. Additionally, this command identifies the router that is currently the active router, the router that is currently the standby router, and the virtual IP address for the HSRP group. Examples 8-3 and 8-4 show the output from the **show standby brief** command issued on routers R1 and R2, where router R1 is currently the active router for group 10 with a virtual IP of 172.16.1.3. It also has a priority of 150 with preemption enabled. In this case, the router with the IP address 172.16.1.2 is the standby router, which happens to be R2, as shown in Example 8-4.



Example 8-3 show standby brief Command Output on Router R1

```
R1#show standby brief
                P indicates configured to preempt.
                |
Interface  Grp  Prio P State   Active      Standby      Virtual IP
Fa0/0      10   150 P Active  local       172.16.1.2   172.16.1.3
```

Example 8-4 show standby brief Command Output on Router R2

```
R2#show standby brief
                P indicates configured to preempt.
                |
Interface  Grp  Prio P State   Active      Standby      Virtual IP
Et0/0      10   100  Standby  172.16.1.1  local        172.16.1.3
```

In addition to an interface's HSRP group number, the interface's state, and the HSRP group's virtual IP address, the **show standby interface_type interface_number** command also displays the HSRP group's virtual MAC address, the HSRP timers, the standby routers priority, and if the current local priority is different than the configured local priority. Issuing this command on router R1, as shown in Example 8-5, shows that the virtual MAC address for HSRP group 10 is **0000.0c07.ac0a**, the timers are default at **3** and **10**, the standby routers priority is **100**, and the local routers current priority is the same as the configured priority.

Example 8-5 *show standby fastethernet 0/0 Command Output on Router R1*

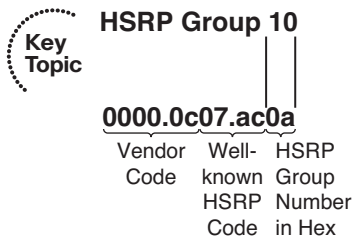
```

R1#show standby fastethernet 0/0
FastEthernet0/0 - Group 10
  State is Active
    1 state change, last state change 01:20:00
  Virtual IP address is 172.16.1.3
  Active virtual MAC address is 0000.0c07.ac0a
    Local virtual MAC address is 0000.0c07.ac0a (v1 default)
  Hello time 3 sec, hold time 10 sec
    Next hello sent in 1.044 secs
  Preemption enabled
  Active router is local
  Standby router is 172.16.1.2, priority 100 (expires in 8.321 sec)
  Priority 150 (configured 150)
  IP redundancy name is "hsrp-Fa0/0-10" (default)

```

Virtual Router MAC Address

The default virtual MAC address for an HSRPv1 group, as shown in Figure 8-2, is based on the HSRP group number. Specifically, the virtual MAC address for an HSRP group begins with a vendor code of 0000.0c, followed with a well-known HSRPv1 code of 07.ac. The last two hexadecimal digits are the hexadecimal representation of the HSRP group number. Therefore, you can have up to 256 HSRPv1 groups. For example, an HSRP group of 10 yields a default virtual MAC address of 0000.0c07.ac0a, because 10 in decimal equates to 0a in hexadecimal.

**Figure 8-2** *HSRP Virtual MAC Address*

The default virtual MAC address for an HSRPv2 group begins with a vendor code of 0000.0c, followed with a well-known HSRPv2 code of 9F.F, and then the last three hexadecimal digits represent the HSRPv2 group. Therefore, you can have a total of 4096 HSRPv2 groups.

Interface Tracking

Key Topic

HSRP interface tracking is a feature that most organizations will deploy. By default, HSRP will only detect a failure of the device itself or the path that is used by the hello packets. What about the uplinks from the routers running HSRP? If they fail, hello packets are

still exchanged successfully, and the active router is still available. However, if the uplink is down, packets are dropped at the active router because it cannot forward them. This is where interface tracking comes into play. Interface tracking allows you to control the priority of a router in an HSRP group based on the status of an interface. If the interface is anything but up/up, you can decrement the priority of the router to a value that is lower than the standby router, and if preemption is enabled on the standby router, it will take over as the active forwarder because it now has the higher priority. You implement interface tracking with the **standby group_number track interface_type interface_number decrement_value** command. You can use the **show standby** command to verify whether interface tracking is configured and the state of the tracked interface, as shown in Example 8-6.

Example 8-6 show standby Command Output on Router R1

```
R1#show standby fa 0/0
FastEthernet0/0 - Group 10
  State is Standby
    2 state changes, last state change 00:02:16
  Virtual IP address is 172.16.1.3
  Active virtual MAC address is 0000.0c07.ac0a
    Local virtual MAC address is 0000.0c07.ac0a (v1 default)
  Hello time 3 sec, hold time 10 sec
    Next hello sent in 0.784 secs
  Preemption enabled
  Active router is 172.16.1.2, priority 100 (expires in 9.312 sec)
  Standby router is local
  Priority 99 (configured 110)
    Track interface FastEthernet2/0 state Down decrement 11
  Group name is "hsrp-Gi0/0-10" (default)
```

In the case of Example 8-6, you can see that the tracked interface state is down. When it is down, the priority will be decremented by 11. Therefore, reviewing the configured priority of 110 and the current priority of 99 indicates why this router is not the active router at the moment. Its priority has been lowered to 99 from 110 because the interface state is down. Now you would have to troubleshoot why the interface is down, which is beyond the scope of our HSRP discussion.

In addition to interface tracking, you can use object tracking, which allows you to track IP-related information such as a route, a group of objects, the status of a service level agreement (SLA), and the status of an interface. We discuss this type of tracking in the “Troubleshooting VRRP” section.

Verifying First Hop



Once you know the current HSRP configuration, you might then check to see whether a host on the HSRP virtual IP address’s subnet can ping the virtual IP address. Based on the topology previously shown in Figure 8-1, Example 8-7 shows a successful ping from Workstation A.

Example 8-7 *Ping Test from Workstation A to the HSRP Virtual IP Address*

```

C:\>ping 172.16.1.3

Pinging 172.16.1.3 with 32 bytes of data:

Reply from 172.16.1.3: bytes=32 time=2ms TTL=255
Reply from 172.16.1.3: bytes=32 time=1ms TTL=255
Reply from 172.16.1.3: bytes=32 time=1ms TTL=255
Reply from 172.16.1.3: bytes=32 time=1ms TTL=255

Ping statistics for 172.16.1.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 2ms, Average = 1ms

```

A client could also be used to verify the appropriate virtual MAC address learned by the client corresponding to the virtual MAC address reported by one of the HSRP routers. Example 8-8 shows Workstation A's Address Resolution Protocol (ARP) cache entry for the HSRP virtual IP address of 172.16.1.3. Notice in the output that the MAC address learned via ARP does match the HSRP virtual MAC address reported by the active HSRP router.

Example 8-8 *Workstation A's ARP Cache*

```

C:\>arp -a

Interface: 172.16.1.4 --- 0x4

    Internet Address      Physical Address      Type
    -----
    172.16.1.3           00-00-0c-07-ac-0a    dynamic

```

However, one of the best tools to use with FHRPs to verify the path is `tracert`. With `tracert`, you can identify the physical first-hop router that the packets are traversing. Example 8-9 displays the `tracert` command executed on a PC. Notice that it states that the first hop is 172.16.1.1. This is the IP address of R1's LAN interface. Therefore, we can conclude the R1 is the active forwarder at the moment. However, suppose that a failure happened and R2 became the active forwarder. The ARP cache would still be the same on the PC. However, the output of `tracert` on the PC would now display that the first hop is 172.16.1.2, as shown in Example 8-10.

Example 8-9 *A Trace from Workstation A Confirming That R1 Is the First Hop (Active Forwarder)*

```

C:\>tracert 192.0.2.1

Tracing route to 192.0.2.1 over a maximum of 30 hops

  1    7 ms    <1 ms    2 ms  172.16.1.1
...output omitted...
Trace complete.

```

Example 8-10 *A Trace from Workstation A Confirming That R2 Is the First Hop (Active Forwarder)*

```
C:\PC1>tracert 192.0.2.1
Tracing route to 192.0.2.1 over a maximum of 30 hops

  1      3 ms      2 ms      4 ms  172.16.1.2
...output omitted...
Trace complete.
```

Debug

You can also use the **debug standby terse** command to view important HSRP changes, such as a state change. Example 8-11 shows this **debug** output on router R2 when router R1's Fast Ethernet 0/0 interface is shut down; notice that router R2's state changes from standby to active.

Example 8-11 *debug standby terse Command Output on Router R2: Changing to Active*

```
R2#
*Mar  1 01:25:45.930: HSRP: Et0/0 Grp 10 Standby: c/Active timer expired
(172.16.1.1)
*Mar  1 01:25:45.930: HSRP: Et0/0 Grp 10 Active router is local, was 172.16.1.1
*Mar  1 01:25:45.930: HSRP: Et0/0 Grp 10 Standby router is unknown, was local
*Mar  1 01:25:45.930: HSRP: Et0/0 Grp 10 Standby -> Active
*Mar  1 01:25:45.930: %HSRP-6-STATECHANGE: Ethernet0/0 Grp 10 state Standby ->
Active
*Mar  1 01:25:45.930: HSRP: Et0/0 Grp 10 Redundancy "hsrp-Et0/0-10" state Standby
-> Active
*Mar  1 01:25:48.935: HSRP: Et0/0 Grp 10 Redundancy group hsrp-Et0/0-10 state
Active -> Active
*Mar  1 01:25:51.936: HSRP: Et0/0 Grp 10 Redundancy group hsrp-Et0/0-10 state
Active -> Active
```

When router R1's Fast Ethernet 0/0 interface is administratively enabled, router R1 resumes its previous role as the active HSRP router for HSRP group 10, because router R1 is configured with the preempt option. The output shown in Example 8-12 demonstrates how router R2 receives a coup message, letting router R2 know that router R1 is taking back its active role.

Example 8-12 *debug standby terse Command Output on Router R2: Changing HSRP to Standby*

```
R2#
*Mar  1 01:27:57.979: HSRP: Et0/0 Grp 10 Coup   in  172.16.1.1 Active  pri 150
vIP 172.16.1.3
*Mar  1 01:27:57.979: HSRP: Et0/0 Grp 10 Active: j/Coup rcvd from higher pri
router (150/172.16.1.1)
```

```

*Mar 1 01:27:57.979: HSRP: Et0/0 Grp 10 Active router is 172.16.1.1, was local
*Mar 1 01:27:57.979: HSRP: Et0/0 Grp 10 Active -> Speak
*Mar 1 01:27:57.979: %HSRP-6-STATECHANGE: Ethernet0/0 Grp 10 state Active -> Speak
*Mar 1 01:27:57.979: HSRP: Et0/0 Grp 10 Redundancy "hsrp-Et0/0-10" state Active
-> Speak
*Mar 1 01:28:07.979: HSRP: Et0/0 Grp 10 Speak: d/Standby timer expired (unknown)
*Mar 1 01:28:07.979: HSRP: Et0/0 Grp 10 Standby router is local
*Mar 1 01:28:07.979: HSRP: Et0/0 Grp 10 Speak -> Standby
*Mar 1 01:28:07.979: HSRP: Et0/0 Grp 10 Redundancy "hsrp-Et0/0-10" state Speak
-> Standby

```

HSRP Trouble Tickets

This section presents various trouble tickets relating to the topics discussed earlier in the chapter. The purpose of these trouble tickets is to give a process that you can follow when troubleshooting in the real world or in an exam environment. All trouble tickets in this section are based on the topology depicted in Figure 8-3.

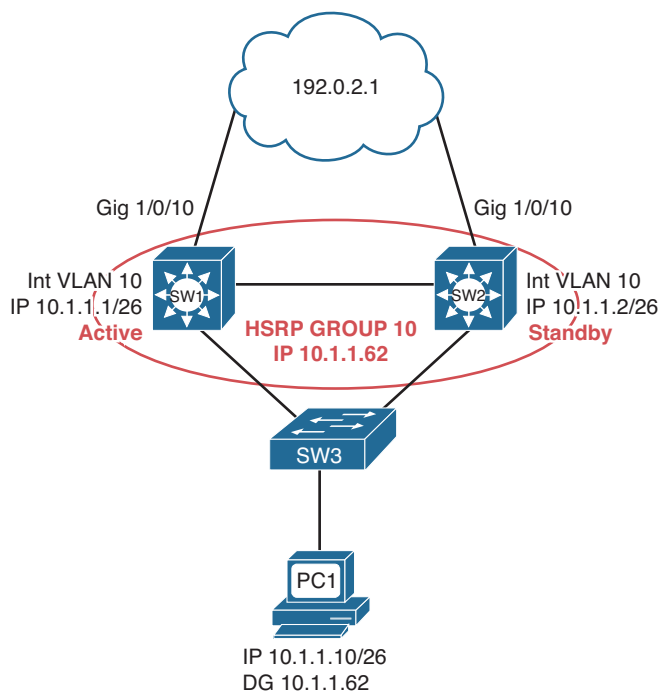


Figure 8-3 HSRP Trouble Ticket Topology

Trouble Ticket 8-1

Problem: According to traffic statistics, all traffic for VLAN 10 is flowing through SW2 to reach the core instead of SW1.

You start by verifying the problem from PC1 on VLAN 10. In this case, the best tool is traceroute because it will identify the router hops (real IPs) along the path. All you care about is the first hop; is it 10.1.1.1 or 10.1.1.2? This will identify whether traffic is flowing through SW1 or SW2 to reach the core. Example 8-13 indicates that SW2 is in fact the HSRP active forwarder for the 10.1.1.0/26 network because it was the first hop returned for the **tracert** command output.

Example 8-13 *A Trace from PC1 Confirming That SW2 Is the First Hop (Active Forwarder)*

```
C:\PC1>tracert 192.0.2.1
Tracing route to 192.0.2.1 over a maximum of 30 hops

  1      6 ms    1 ms    2 ms  10.1.1.2
...output omitted
Trace complete.
```

Next you need to confirm that this is in fact true by reviewing the output of HSRP **show** commands. Example 8-14 displays the output of **show standby brief** on SW2. Notice that under the Active column it states local and that under the Standby column it displays 10.1.1.1, which is the IP address of the standby router, SW1.

Example 8-14 *show standby brief Command Output on SW2*

```
SW2#show standby brief
                P indicates configured to preempt.
                |
Interface  Grp  Pri  P State  Active  Standby  Virtual IP
Vl10      10   100  P Active local   10.1.1.1 10.1.1.62
```

Reviewing Figure 8-3 indicates that SW1 should be the active forwarder for group 10. Now is an excellent time to review the output of **show standby brief** on SW1 to see whether anything stands out that might be the issue. Example 8-15 indicates that SW1 is indeed the standby router for group 10.

Example 8-15 *show standby brief Command Output on SW1*

```
SW1#show standby brief
                P indicates configured to preempt.
                |
Interface  Grp  Pri  P State  Active  Standby  Virtual IP
Vl10      10   10  P Standby 10.1.1.2 local   10.1.1.62
```

However, if you look very closely at Examples 8-14 and 8-15, you should notice that SW1 has a priority of 10, and SW2 has a priority of 100. The HSRP router that has the higher priority is the active forwarder. You should check the output of **show standby** on SW1 to determine whether that is the configured priority or if some tracked object is down and causing the priority to be lowered. Example 8-16 displays the output of **show standby** on SW1. Notice that the priority is listed as 10 and that it states it is configured as 10.

It must have been mistyped. Checking your documentation indicates that the priority should be configured to 110.

Example 8-16 *show standby Command Output on SW1*

```
SW1#show standby
Vlan10 - Group 10
  State is Standby
    4 state changes, last state change 00:06:51
  Virtual IP address is 10.1.1.62
  Active virtual MAC address is 0000.0c07.ac0a
    Local virtual MAC address is 0000.0c07.ac0a (v1 default)
  Hello time 3 sec, hold time 10 sec
    Next hello sent in 2.016 secs
  Preemption enabled
  Active router is 10.1.1.2, priority 100 (expires in 9.488 sec)
  Standby router is local
  Priority 10 (configured 10)
    Track interface GigabitEthernet1/0/10 state Up decrement 11
  Group name is "hsrp-Vl10-10" (default)
```

Example 8-17 displays the interface VLAN 10 configuration, which shows that the priority was configured to 10 instead of 110.

Example 8-17 *show run interface vlan 10 Command Output on SW1*

```
SW1#show run interface vlan 10
Building configuration...

Current configuration : 163 bytes
!
interface Vlan10
  ip address 10.1.1.1 255.255.255.192
  standby 10 ip 10.1.1.62
  standby 10 priority 10
  standby 10 preempt
  standby 10 track 1 decrement 11
end
```

After fixing the issue by executing the command **standby 10 priority 110** in VLAN 10 interface configuration mode on SW1, you see the following syslog message confirming that SW1 is now the active forwarder:

```
%HSRP-5-STATECHANGE: Vlan10 Grp 10 state Standby -> Active
```

You then reissue the **tracert** command on PC1, as shown in Example 8-18, and confirm that SW1 is in fact the active forwarder now.

Example 8-18 *A Trace from PC1 Confirming That SW1 Is the First Hop (Active Forwarder)*

```
C:\PC1>tracert 192.0.2.1
Tracing route to 192.0.2.1 over a maximum of 30 hops

  1    7 ms    <1 ms    2 ms  10.1.1.1
...output omitted...
Trace complete.
```

Trouble Ticket 8-2

Problem: According to traffic statistics, all traffic for VLAN 10 is flowing through SW2 to reach the core instead of SW1.

You start by verifying the problem from PC1 on VLAN 10. In this case, the best tool is traceroute because it will identify the router hops (real IPs) along the path. All you care about is the first hop; is it 10.1.1.1 or 10.1.1.2? This will identify whether traffic is flowing through SW1 or SW2 to reach the core. Example 8-19 indicates that SW2 is in fact the HSRP active forwarder for the 10.1.1.0/26 network because it was the first hop returned for the **tracert** command output.

Example 8-19 *A Trace from PC1 Confirming That SW2 Is the First Hop (Active Forwarder)*

```
C:\PC1>tracert 192.0.2.1
Tracing route to 192.0.2.1 over a maximum of 30 hops

  1    6 ms    1 ms    2 ms  10.1.1.2
...output omitted
Trace complete.
```

Next you need to confirm that this is in fact true by reviewing the output of HSRP **show** commands. Example 8-20 displays the output of **show standby brief** on SW2. Notice that under the Active column it states local and that under the Standby column it displays 10.1.1.1, which is the IP address of the standby router, SW1.

Example 8-20 *show standby brief Command Output on SW2*

```
SW2#show standby brief
                P indicates configured to preempt.
                |
Interface  Grp  Pri  P State  Active          Standby          Virtual IP
Vl10      10   100  P Active local          10.1.1.1         10.1.1.62
```

Reviewing Figure 8-3 indicates that SW1 should be the active forwarder for group 10. Now is an excellent time to review the output of **show standby brief** on SW1 to see whether anything stands out that might be the issue. Example 8-21 indicates that SW1 is indeed the standby router for group 10.

Example 8-21 *show standby brief Command Output on SW1*

```
SW1#show standby brief
                P indicates configured to preempt.
                |
Interface   Grp  Pri P State      Active      Standby      Virtual IP
Vl10        10   110  State 10.1.1.2    local        10.1.1.62
```

However, if you look very closely at Examples 8-20 and 8-21, you should notice that SW1 has a priority of 110 and that SW2 has a priority of 100. The HSRP router that has the higher priority should be the active forwarder. However, in this case, it is not. Taking an even closer look at Examples 8-20 and 8-21, you notice that SW1 does not have preemption enabled, as indicated by the missing *P* in the output.

You check the output of **show standby** on SW1, as shown in Example 8-22, and it indicates that preemption is disabled.

Example 8-22 *show standby Command Output on SW1*

```
SW1#show standby
Vlan10 - Group 10
  State is Standby
    7 state changes, last state change 02:39:07
  Virtual IP address is 10.1.1.62
  Active virtual MAC address is 0000.0c07.ac0a
    Local virtual MAC address is 0000.0c07.ac0a (v1 default)
  Hello time 3 sec, hold time 10 sec
    Next hello sent in 1.520 secs
  Preemption disabled
  Active router is 10.1.1.2, priority 100 (expires in 10.112 sec)
  Standby router is local
  Priority 110 (configured 110)
    Track interface GigabitEthernet1/0/10 state Up decrement 11
  Group name is "hsrp-Vl10-10" (default)
```

If SW1 is expected to take over as the active forwarder when it has a higher priority, preemption needs to be on.

After fixing the issue by executing the command, **standby 10 preempt** in VLAN 10 interface configuration mode on SW1, you see the following syslog message confirming that SW1 is now the active forwarder:

```
%HSRP-5-STATECHANGE: Vlan10 Grp 10 state Standby -> Active
```

You then reissue the **tracert** command on PC1, as shown in Example 8-23, and confirm that SW1 is in fact the active forwarder now.

Example 8-23 *A Trace from PC1 Confirming That SW1 Is the First Hop (Active Forwarder)*

```
C:\PC1>tracert 192.0.2.1
Tracing route to 192.0.2.1 over a maximum of 30 hops

  1    7 ms    <1 ms    2 ms  10.1.1.1
...output omitted...
Trace complete.
```

Trouble Ticket 8-3

Problem: Users in VLAN 10 are reporting that they are not able to reach any resources outside their LAN.

You start by verifying the problem from PC1 on VLAN 10. You ping 192.0.2.1, as shown in Example 8-24, and it fails.

Example 8-24 *Failed Ping from PC1 to Destination Outside LAN*

```
C:\PC1>ping 192.0.2.1
Pinging 192.0.2.1 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.0.2.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

You ping the default gateway of PC1, which is the virtual router IP address of 10.1.1.62, and it is successful, as shown in Example 8-25.

Example 8-25 *Successful Ping from PC1 to Default Gateway*

```
C:\PC1>ping 10.1.1.62

Reply from 10.1.1.62: bytes=32 time=1ms TTL=128
Reply from 10.1.1.62: bytes=32 time=1ms TTL=128
Reply from 10.1.1.62: bytes=32 time=1ms TTL=128
Reply from 10.1.1.62: bytes=32 time=1ms TTL=128

Ping statistics for 10.1.1.62:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

So far, you have confirmed that connectivity beyond the default gateway is not possible but that connectivity to the default gateway is. You decide to use traceroute to determine which router is currently the active forwarder. Example 8-26 confirms that it is SW1 at 10.1.1.1. However, notice how no other hop is displayed and you receive a destination host unreachable message from 10.1.1.1. Keep this in mind; we will come back to it.

Example 8-26 *A Trace from PC1 Confirming That SW1 Is the First Hop (Active Forwarder)*

```
C:\PC1>tracert 192.0.2.1
Tracing route to 192.0.2.1 over a maximum of 30 hops

  1         4 ms         2 ms         2 ms  10.1.1.1
  2  10.1.1.1  reports: Destination host unreachable.

Trace complete.
```

Next you need to confirm that SW1 is in fact the active forwarder by reviewing the output of HSRP **show** commands. Example 8-27 displays the output of **show standby brief** on SW1. Notice that under the Active column it states local and that under the Standby column it displays 10.1.1.2, which is the IP address of the standby router, SW2.

Example 8-27 *show standby brief Command Output on SW1*

```
SW1#show standby brief
                P indicates configured to preempt.
                |
Interface      Grp  Pri  P State      Active      Standby      Virtual IP
Vl10           10   109  P Active     local       10.1.1.2     10.1.1.62
```

Review Example 8-26 again. Remember how the **tracert** command output is failing at SW1? This is a good indication that SW1 cannot route the packet to 192.0.2.1. You issue the **show ip route** command on SW1, as shown in Example 8-28. All you see are connected and local routes. However, there is no connected route for Gig1/0/10, nor are there any routes learned from a neighboring router in the core on Gig1/0/10.

Example 8-28 *show ip route Command Output on SW1*

```
SW1#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override
```

```

Gateway of last resort is not set

    10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C       10.1.1.0/26 is directly connected, Vlan10
L       10.1.1.1/32 is directly connected, Vlan10
C       10.1.1.64/26 is directly connected, Vlan20
L       10.1.1.65/32 is directly connected, Vlan20

```

You issue the command **show ip interface brief | exclude unassigned**, as shown in Example 8-29, on SW1 and notice that Gig1/0/10 is down/down. There is an issue between SW1 and the core. You escalate the problem because it is beyond your control. However, you need to determine in the meantime why HSRP did not successfully fail over to SW2 as the active forwarder for group 10 in case this happens again.

Example 8-29 show ip interface brief | exclude unassigned *Command Output on SW1*

```

SW1#show ip int brief | ex unassigned

```

Interface	IP-Address	OK?	Method	Status	Protocol
Vlan10	10.1.1.1	YES	NVRAM	up	up
Vlan20	10.1.1.65	YES	NVRAM	up	up
GigabitEthernet1/0/10	10.1.10.2	YES	NVRAM	down	down

Interface tracking is a feature that allows an HSRP-enabled router to decrement its priority by a specified value if the status of an interface goes down. This ensures that the active forwarder does not maintain the active status if it is not fit to do so. If it did, it might black hole traffic as it did in this scenario. Using the command **show standby** on SW1 indicates that you are tracking interface Gigabit Ethernet 1/0/10, as shown in Example 8-30. It also shows that it is down and that the current priority is 109 instead of the configured 110.

Example 8-30 show standby *Command Output on SW1*

```

SW1#show standby
Vlan10 - Group 10
  State is Active
    8 state changes, last state change 00:14:11
  Virtual IP address is 10.1.1.62
  Active virtual MAC address is 0000.0c07.ac0a
    Local virtual MAC address is 0000.0c07.ac0a (v1 default)
  Hello time 3 sec, hold time 10 sec
    Next hello sent in 0.736 secs
  Preemption enabled
  Active router is local
  Standby router is 10.1.1.2, priority 100 (expires in 7.760 sec)
  Priority 109 (configured 110)
    Track interface GigabitEthernet1/0/10 state Down decrement 1
  Group name is "hsrp-Vl10-10" (default)

```

The problem in this case is clear. Interface tracking was configured incorrectly, as verified in Example 8-31, which displays the output of **show run interface vlan 10**. In this case, the decrement value was set to 1. It appears that whoever configured it thought that the decrement value identified what the new priority should be if the interface goes down. But in reality, it states how much to lower the configured priority by. Therefore, the configured priority is 110 and you minus 1, which gives you 109.

Example 8-31 *show run interface vlan 10 Command Output on SW1*

```
SW1#show run interface vlan 10
Building configuration...

Current configuration : 163 bytes
!
interface Vlan10
 ip address 10.1.1.1 255.255.255.192
 standby 10 ip 10.1.1.62
 standby 10 priority 110
 standby 10 preempt
 standby 10 track 1 decrement 1
end
```

After you solve this problem by changing the decrement value to a value of 11 or higher (so that the priority of SW1 will be 99 or lower), you will notice a syslog message on SW1 indicating that SW1 is no longer in the active state, and on SW2 you will see a syslog message indicating that it is now in the active state. These are examples of the syslog messages:

```
SW1#
%HSRP-5-STATECHANGE: Vlan10 Grp 10 state Active -> Speak
SW1#
%HSRP-5-STATECHANGE: Vlan10 Grp 10 state Speak -> Standby
SW1#
SW2#
%HSRP-5-STATECHANGE: Vlan10 Grp 10 state Standby -> Active
SW2#
```

You then reissue the **tracert** command on PC1, as shown in Example 8-32, and confirm that SW2 is the active forwarder.

Example 8-32 *A Trace from PC1 Confirming That SW2 Is the First Hop (Active Forwarder)*

```
C:\PC1>tracert 192.0.2.1
Tracing route to 192.0.2.1 over a maximum of 30 hops

 1          3 ms          2 ms          4 ms  10.1.1.2
```

```
...output omitted...
 7      48 ms      40 ms      30 ms  192.0.2.1

Trace complete.
```

In addition, you need to ping from a client to make sure that the problem is officially solved. It is, as shown by the successful ping in Example 8-33.

Example 8-33 *Successful Ping from PC1*

```
C:\PC1>ping 192.0.2.1

Reply from 192.0.2.1: bytes=32 time=1ms TTL=128
Reply from 192.0.2.1: bytes=32 time=1ms TTL=128
Reply from 192.0.2.1: bytes=32 time=1ms TTL=128
Reply from 192.0.2.1: bytes=32 time=1ms TTL=128

Ping statistics for 192.0.2.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Troubleshooting VRRP

Virtual Router Redundancy Protocol (VRRP), is an IETF standard FHRP based on Cisco's HSRP protocol. Therefore, your knowledge of HSRP can transfer over to VRRP. However, although they are similar, VRRP and HSRP are not compatible. In addition, as a troubleshooter, you need to understand the differences of VRRP so that you can successfully troubleshoot issues related to it.

This section focuses on the behavior of VRRP and how to verify and troubleshoot VRRP issues.

Reviewing VRRP



Like HSRP, VRRP allows a collection of routers to service traffic destined for a single IP address. Unlike HSRP, the IP address serviced by a VRRP group does not have to be a unique/unused IP address. The IP address can be the address of a router's physical interface on the LAN. A VRRP virtual router identifier (VRID) is made up of a *virtual master router* and multiple routers acting as *virtual router backups*, as shown in Figure 8-4. (Note that the VRID is the same concept as an HSRP group.) The virtual master router is responsible for handing out the virtual MAC address associated with the LAN's default gateway IP address and forwarding traffic sent to the default gateway. The *virtual router backups* are waiting for the master to fail so that one of them can take over the *virtual master router* role.

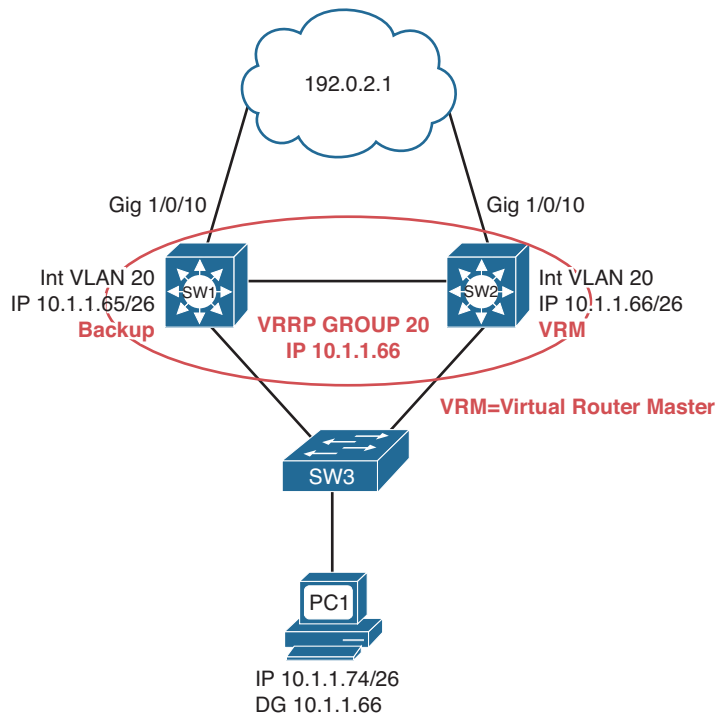


Figure 8-4 Basic VRRP Operation

Examples 8-34 and 8-35 show the VRRP configuration for SW1 and SW2.

Example 8-34 VRRP Configuration on Router R1

```
SW1#show run
...OUTPUT OMITTED...
interface vlan 20
 ip address 10.1.1.65 255.255.255.192
 vrrp 20 ip 10.1.1.66
...OUTPUT OMITTED...
```

Example 8-35 VRRP Configuration on Router R2

```
SW2#show run
...OUTPUT OMITTED...
interface vlan 20
 ip address 10.1.1.66 255.255.255.192
 vrrp 20 ip 10.1.1.66
...OUTPUT OMITTED...
```

Notice in Examples 8-34 and 8-35 that the VRRP group IP address is the same as the SVI on SW2. As a result of this, SW2 will automatically be the *virtual router master* because it owns that IP address, regardless of what the priority is because it will give itself a pri-

ority of 255 automatically. By default, VRRP uses a priority of 100 like HSRP. Also make note that preemption is on by default. Therefore, you do not have to manually enable it.

VRRP Verification and Troubleshooting

When verifying a VRRP configuration or troubleshooting a VRRP issue, you should begin by determining the following information about the VRRP group under inspection:



- Which router is the virtual router master?
- How was the virtual router master chosen?
- Which routers, if any, are configured with the preempt option? (Enabled by default)
- What is the IP address of the virtual router?
- What is the virtual MAC address?
- Is object tracking on?

You can use the **show vrrp brief** command to show which interface is participating in a VRRP group. It identifies the VRRP group number, the priority of the interface, whether it owns the IP being used as the virtual router IP, and whether preemption is enabled. In addition, this command will identify the current state of the router along with the master address and the group address.

Examples 8-36 and 8-37 show the output from the **show vrrp brief** command issued on SW1 and SW2. Notice how SW2 is currently the master router for group 20. You can also see that preemption is enabled and that SW2 owns the IP address that is being used as the virtual router IP address. SW1 is in the backup state.



Example 8-36 show vrrp brief Command Output on Router SW1

```
SW1#show vrrp brief
Interface      Grp Pri Time   Own Pre State   Master addr   Group addr
Vl20           20  100 3609      Y Backup  10.1.1.66     10.1.1.66
```

Example 8-37 show vrrp brief Command Output on SW2

```
SW2#show vrrp brief
Interface      Grp Pri Time   Own Pre State   Master addr   Group addr
Vl20           20  255 3003      Y  Y Master   10.1.1.66     10.1.1.66
```

In Examples 8-36 and 8-37, notice how SW2 has a priority of 255. In the previous configuration examples, we did not configure the priority. We kept it at the default of 100. In this case, it is 255 because SW2 owns the IP that is being used as the virtual IP address. Therefore, it automatically changes its priority to 255 so that it becomes the virtual router master for the group.

In addition to an interface's VRRP group number, the state, the priority, and the VRRP group's virtual IP address, the **show vrrp interface interface_type interface_number** command also displays the VRRP group's virtual MAC address and the VRRP timers.

By default, VRRP timers are 1 second for the Advertisement interval and 3 seconds for the Master Down interval. Issuing this command on SW2, as shown in Example 8-38, shows that the virtual MAC address for VRRP group 20 is 0000.5e00.0114, the timers are default at 1 and 3, the priority is 255, and SW2 is the master router.

Example 8-38 *show vrrp interface vlan 20 Command Output on SW2*

```
SW2#show vrrp interface vlan 20
Vlan20 - Group 20
  State is Master
  Virtual IP address is 10.1.1.66
  Virtual MAC address is 0000.5e00.0114
  Advertisement interval is 1.000 sec
  Preemption enabled
  Priority is 255
  Master Router is 10.1.1.66 (local), priority is 255
  Master Advertisement interval is 1.000 sec
  Master Down interval is 3.003 sec
```

Virtual Router MAC Address

The default virtual MAC address for a VRRP group, as shown in Figure 8-5, is based on the VRRP VRID, which is just a fancy way to identify the group number. Specifically, the virtual MAC address for a VRRP group begins with a vendor code of 0000.5e (IANA's organizationally unique identifier [OUI]), followed with a well-known VRRP address block of 00.01. The last two hexadecimal digits are the hexadecimal representation of the VRID (group) number. For example, a VRRP group of 20 yields a default virtual MAC address of 0000.5e00.0114, because 20 in decimal equates to 14 in hexadecimal.

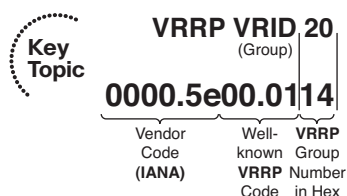


Figure 8-5 *VRRP Virtual MAC Address*

Object Tracking



Object tracking is a feature that most organizations will deploy when using VRRP. By default, VRRP will only detect a failure of the device itself or the path that is used by the hello packets. What about the uplinks from the routers running VRRP? If they fail, hello packets are still exchanged successfully, and the virtual master router is still available. Therefore, if the uplink is down, packets are dropped at the virtual master router because it cannot forward them. This is where object tracking comes into play. Object tracking enables you to control the priority of a router in a VRRP group based on the status of an object. The object can be IP-related information such as a route, a group of objects, the

status of an SLA probe, and the status of an interface. If the object is anything but up, the priority of the router can be decremented to a value that is lower than the standby router, and because preemption is enabled by default, the standby router will take over as the virtual master router because it now has the higher priority. You can use the **show vrrp** command to verify whether object tracking is configured, as shown in Example 8-39, and the state of the tracked object.

Example 8-39 **show vrrp** Command Output on Router SW2

```
SW2#show vrrp
VLAN 20 - Group 20
  State is Backup
  Virtual IP address is 10.1.1.126
  Virtual MAC address is 0000.5e00.0114
  Advertisement interval is 1.000 sec
  Preemption enabled
  Priority is 99 (cfgd 110)
    Track object 1 state Down decrement 11
  Master Router is 10.1.1.65, priority is 100
  Master Advertisement interval is 1.000 sec
  Master Down interval is 3.570 sec (expires in 3.026 sec)
```

In the case of Example 8-39, you can see that the tracked object 1 is in a state of down, and when it is down, it will decrement the priority by 11. You can see the current priority is 99 and the configured priority is 110 ($110 - 11 = 99$).

However, you need to find out what the tracked object is specifically so that you can troubleshoot further. Using the command **show track** you can verify what tracked object number 1 is tracking. In Example 8-40, you can verify that it is the status of the line protocol on interface Gigabit Ethernet 1/0/10. It is *admin-down* and being tracked by VRRP group 20.

Example 8-40 **show track** Command Output on Router SW2

```
SW2#show track
Track 1
  Interface GigabitEthernet1/0/10 line-protocol
  Line protocol is Down (hw admin-down)
    2 changes, last change 00:05:13
  Tracked by:
    VRRP VLAN20 20
```

Now you would have to troubleshoot why the interface is down, which is beyond the scope of our VRRP discussion.

Verifying First Hop



Once you know the current VRRP configuration, you might then check to see whether a host on the VRRP virtual IP address's subnet can ping the virtual IP address. Based on the topology previously shown in Figure 8-4, Example 8-41 shows a successful ping from PC1.

Example 8-41 *Ping Test from PC1 to the VRRP Virtual IP Address*

```
C:\PC1>ping 10.1.1.66

Pinging 10.1.1.66 with 32 bytes of data:

Reply from 10.1.1.66: bytes=32 time=2ms TTL=255
Reply from 10.1.1.66: bytes=32 time=1ms TTL=255
Reply from 10.1.1.66: bytes=32 time=1ms TTL=255
Reply from 10.1.1.66: bytes=32 time=1ms TTL=255

Ping statistics for 10.1.1.66:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 2ms, Average = 1ms
```

However, that does not prove that we are using the virtual MAC address and VRRP. Therefore, from the client, you should also verify the virtual MAC address learned by the client corresponds to the virtual MAC address reported by the VRRP virtual router master. Example 8-42 shows Workstation A's ARP cache entry for the VRRP virtual IP address of 10.1.1.66. Notice in the output that the MAC address learned via ARP does match the VRRP virtual MAC address of the master router.

Example 8-42 *PC1 ARP Cache*

```
C:\PC1>arp -a

Interface: 10.1.1.74 --- 0x4

   Internet Address      Physical Address      Type
   -----
   10.1.1.66             00-00-5e-00-01-14    dynamic
```

However, as discussed with HSRP, one of the best tools to use with FHRPs to verify the path is traceroute. With traceroute, you can identify the physical first-hop router that the packets are traversing. Example 8-43 displays the **tracert** command executed on PC1. Notice that it states that the first hop is 10.1.1.66. This is the IP address of SW2's VLAN 20 SVI. Therefore, you can conclude the SW2 is the virtual router master at the moment. Suppose, however, that a failure happened and SW1 became the virtual router master. The ARP cache would still be the same on PC1; however, the output of **tracert** on the PC would now display that the first hop is 10.1.1.65, as shown in Example 8-44.

Example 8-43 *A Trace from PC1 Confirming That SW2 Is the First Hop (Virtual Router Master)*

```
C:\PC1>tracert 192.0.2.1

Tracing route to 192.0.2.1 over a maximum of 30 hops

  1    7 ms    <1 ms    2 ms  10.1.1.66
...output omitted...
Trace complete.
```

Example 8-44 *A Trace from PC1 Confirming That SW1 Is the First Hop (Virtual Router Master)*

```

C:\PC1>tracert 192.0.2.1
Tracing route to 192.0.2.1 over a maximum of 30 hops

  1    7 ms    <1 ms    2 ms  10.1.1.65
...output omitted...
Trace complete.

```

VRRP Trouble Tickets

This section presents various trouble tickets relating to the topics discussed earlier in the chapter. The purpose of these trouble tickets is to give a process that you can follow when troubleshooting in the real world or in an exam environment. All trouble tickets in this section are based on the topology depicted in Figure 8-6.

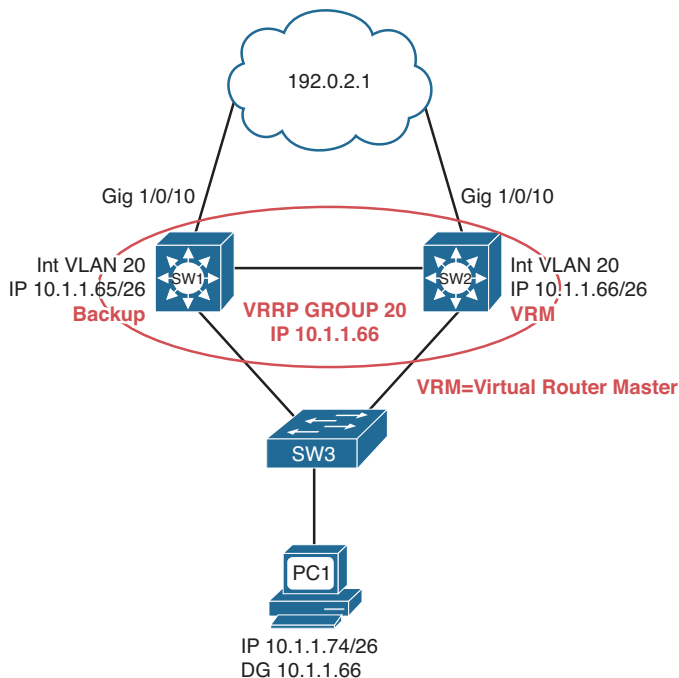


Figure 8-6 VRRP Trouble Ticket Topology

Trouble Ticket 8-4

Problem: According to traffic statistics, all traffic for VLAN 20 is flowing through SW1 to reach the core instead of SW2.

You start by verifying the problem from PC1 on VLAN 20. In this case, the best tool is traceroute because it will identify the router hops (real IPs) along the path. All you care about is the first hop; is it 10.1.1.65 or 10.1.1.66? This will identify whether traffic is flowing through SW1 or SW2 to reach the core. Example 8-45 indicates that SW1 should be the VRRP virtual router master for the 10.1.1.64/26 network, because it was the first hop returned for the **tracert** command.

Example 8-45 *A Trace from PC1 Confirming That SW1 Is the First Hop (Master)*

```
C:\PC1>tracert 192.0.2.1
Tracing route to 192.0.2.1 over a maximum of 30 hops

  1    2 ms    2 ms    1 ms   10.1.1.65
...output omitted...
Trace complete.
```

Next you need to confirm that this is in fact true by reviewing the output of VRRP **show** commands. Example 8-46 displays the output of **show vrrp brief** on SW1. Notice that under the State column it states Backup and the Master addr is 10.1.1.66, which is also the virtual IP address for the group. Therefore, SW1 is not the VRRP master, even though it is being used as the first hop.

Example 8-46 *show vrrp brief Command Output on SW1*

```
SW1#show vrrp brief
```

Interface	Grp	Pri	Time	Own	Pre	State	Master addr	Group addr
Vl20	20	100	3609		Y	Backup	10.1.1.66	10.1.1.66

Reviewing Figure 8-6 indicates that SW2 should be the virtual router master of the group, and it appears that it is. Now is an excellent time to review the output of **show vrrp brief** on SW2 to verify this. Example 8-47 indicates that SW2 is in the master state.

Example 8-47 *show standby brief Command Output on SW2*

```
SW2#show vrrp brief
```

Interface	Grp	Pri	Time	Own	Pre	State	Master addr	Group addr
Vl20	20	255	3003	Y	Y	Master	10.1.1.66	10.1.1.66

What would be causing SW1 and SW2 to be in their correct states, yet the wrong device being used as the first hop? Recall that when a client makes an ARP request for the VRRP group MAC address, the virtual router master will respond with the group MAC address. In this case, it should be 0000.5e00.0114 for group 20. On PC1, you issue the **arp -a** command, as shown in Example 8-48, to verify the MAC address being used by the client for the 10.1.1.66 address. It does not appear that the client is learning a VRRP MAC address, because none of the MAC addresses listed start with 0000.5e00.01. Also notice how the Internet address listed is 10.1.1.65, with a MAC of 28-93-fe-3a-e3-43. That is the IP and MAC address of interface VLAN 20 on SW1, as shown in Example 8-49, which displays the output of the **show interface vlan 20** command.

Example 8-48 *Verifying PC1's ARP Cache*

```
C:\PC1>arp -a

Interface: 10.1.1.74 --- 0x2

    Internet Address      Physical Address      Type
    10.1.1.65             28-93-fe-3a-e3-43     dynamic
```

Example 8-49 *Verifying SW1's SVI IP Address and MAC Address*

```
SW1#show interface vlan 20
Vlan20 is up, line protocol is up
  Hardware is EtherSVI, address is 2893.fe3a.e343 (bia 2893.fe3a.e343)
  Internet address is 10.1.1.65/26
  ...output omitted...
```

It appears that the PCs might be configured with the wrong default gateway IP address. From the **show** commands you just reviewed, it seems as if they are using 10.1.1.65 as the default gateway address instead of the VRRP virtual IP of 10.1.1.66. Using the command **ipconfig** on PC1, you confirm that the default gateway is 10.1.1.65 and not 10.1.1.66, as shown in Example 8-50.

Example 8-50 *Verifying the Default Gateway on PCs*

```
C:\PC1>ipconfig

Windows IP Configuration

Ethernet adapter PC1:

    Connection-specific DNS Suffix  . : 
    IP Address. . . . . : 10.1.1.74
    Subnet Mask . . . . . : 255.255.255.192
    IP Address. . . . . : 2001:20::20
    IP Address. . . . . : fe80::a00:27ff:fea2:ce47%4
    Default Gateway . . . . . : 10.1.1.65
```

You contact the administrator of the DHCP server and inform him of the issue. After the adjustments are made and the clients have the correct default gateway, as shown in Example 8-51, you reissue the **tracert** command and confirm that SW2 (10.1.1.66) is being used as the first hop, as shown in Example 8-51 as well.

Example 8-51 *Verifying the Default Gateway on PCs After Adjustments*

```
C:\PC1>ipconfig

Windows IP Configuration

Ethernet adapter PC1:

    Connection-specific DNS Suffix  . : 
    IP Address. . . . . : 10.1.1.74
```

```

Subnet Mask . . . . . : 255.255.255.192
IP Address. . . . . : 2001:20::20
IP Address. . . . . : fe80::a00:27ff:fea2:ce47%4
Default Gateway . . . . . : 10.1.1.66

C:\PC1>tracert 192.0.2.1
Tracing route to 192.0.2.1 over a maximum of 30 hops

 1      3 ms      1 ms      2 ms  10.1.1.66
...output omitted...
Trace complete.

```

However, it is important that you confirm the correct VRRP MAC address is being used by checking the ARP cache on the PCs. In Example 8-52, you confirm with the **arp -a** command that the MAC address of 0000.5e00.0114 for group 20 is being used.

Example 8-52 Verifying PC1's ARP Cache After Adjustments

```

C:\PC1>arp -a

Interface: 10.1.1.74 --- 0x2

 Internet Address      Physical Address      Type
-----
 10.1.1.66             00-00-5e-00-01-14     dynamic

```

Trouble Ticket 8-5

Problem: According to traffic statistics, when the uplink between SW3 and SW2 goes down, all traffic for VLAN 20 is flowing through SW3, SW1, and then SW2 and routed out to the core, as shown in Figure 8-7. (Note that the default gateway IP address differs from the previous figures.)

If the uplink between SW3 and SW2 is not available, SW1 should become the VRRP virtual router master so that traffic flow is optimized in the LAN.

You start verifying the problem by shutting down the link between SW3 and SW2. You then trace the path from PC1 to an IP address outside the LAN. All you care about is the first hop; is it 10.1.1.65 or 10.1.1.66? This will identify whether traffic is flowing through SW1 or SW2 to reach the core. Example 8-53 indicates that SW2 is in fact the VRRP virtual router master for the 10.1.1.64/26 network, because it was the first hop returned for the **tracert** command.

Example 8-53 A Trace from PC1 Confirming That SW2 Is the First Hop (Master)

```

C:\PC1>tracert 192.0.2.1
Tracing route to 192.0.2.1 over a maximum of 30 hops

 1      2 ms      2 ms      2 ms  10.1.1.66
...output omitted...
Trace complete.

```

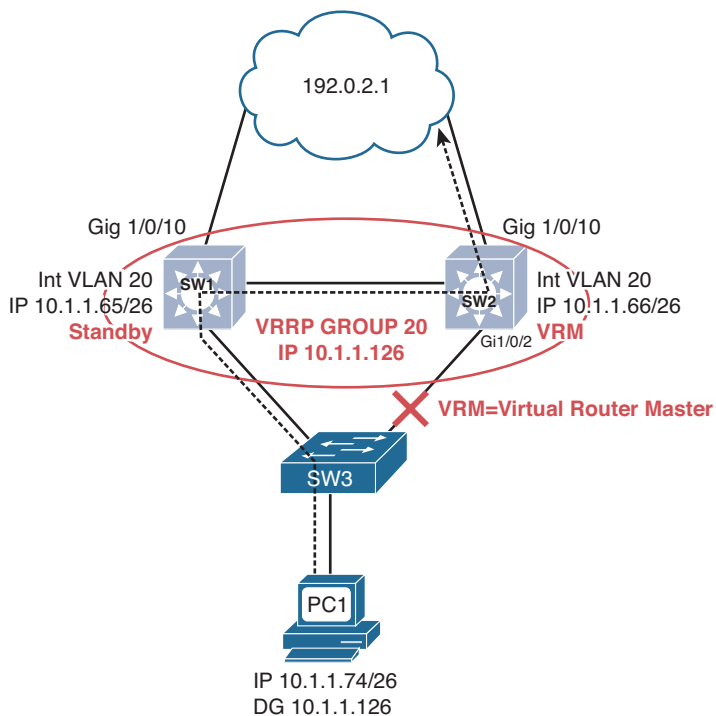



Figure 8-7 VRRP Suboptimal Traffic Flow Topology

Next you need to confirm that this is in fact true by reviewing the output of VRRP **show** commands. Example 8-54 displays the output of **show vrrp brief** on SW2. Notice that under the State column it states Master and the Master addr is 10.1.1.66 (SW2) for the group address 10.1.1.126. All looks fine so far.

Example 8-54 **show vrrp brief** Command Output on SW2

```
SW2#show vrrp brief
```

Interface	Grp	Pri	Time	Own	Pre	State	Master addr	Group addr
Vl20	20	100	3570		Y	Master	10.1.1.66	10.1.1.126

Next you review the output of **show vrrp**, as shown in Example 8-55. In this output, you notice that SW2 is the master but that there is a problem with the priority. The configured priority is 110, but the current is 100. As a result, it has been decremented dynamically. This can be verified with the tracked object that is currently down. It indicates that the tracking object 1 is down, and when it is down, the priority will be decremented by 10 ($110 - 10 = 100$).

Example 8-55 **show vrrp** Command Output on SW2

```
SW2#show vrrp
```

```
Vlan20 - Group 20
State is Master
```

```

Virtual IP address is 10.1.1.126
Virtual MAC address is 0000.5e00.0114
Advertisement interval is 1.000 sec
Preemption enabled
Priority is 100 (cfgd 110)
  Track object 1 state Down decrement 10
Master Router is 10.1.1.66 (local), priority is 100
Master Advertisement interval is 1.000 sec
Master Down interval is 3.570 sec

```

What is tracking object 1? To verify, you execute the **show track** command on SW2. As Example 8-56 displays, the output of **show track** indicates that you are tracking the line protocol of Gigabit Ethernet 1/0/2 for VRRP on interface VLAN 20 for group 20. At this point in time, Gigabit Ethernet 1/0/2 is down, and as a result, VRRP decremented the priority by 10, as you saw in Example 8-55. However, SW2 is still the virtual router master for group 20 even though the priority is being decremented.

Example 8-56 **show track** Command Output on SW2

```

SW2#show track
Track 1
  Interface GigabitEthernet1/0/2 line-protocol
  Line protocol is Down (hw down)
    6 changes, last change 01:39:45
  Tracked by:
    VRRP Vlan20 20

```

Next you verify the priority on SW1 with the **show vrrp** command, as shown in Example 8-57. The output clearly shows that the priority of SW1 is 100, which is the same as SW2. Reviewing the output of **show vrrp** for SW1 and SW2 identifies that preemption is enabled. If that is the case, and the priority is tied, why is SW2 the virtual router master? When priority is tied, the IP address of the LAN interface participating in VRRP is used as the tiebreaker, just like HSRP. Because SW2 has the higher LAN IP address, it is the virtual router master.

Example 8-57 **show vrrp** Command Output on SW1

```

SW1#show vrrp
Vlan20 - Group 20
  State is Backup
  Virtual IP address is 10.1.1.126
  Virtual MAC address is 0000.5e00.0114
  Advertisement interval is 1.000 sec
  Preemption enabled
  Priority is 100
  Master Router is 10.1.1.66, priority is 100
  Master Advertisement interval is 1.000 sec
  Master Down interval is 3.609 sec (expires in 3.575 sec)

```

How can you make sure that SW1 takes over as the virtual router master if the uplink between SW3 and SW2 fails? In this case, make sure that the priority of SW2 is dropped below that of SW1. On SW2, you issue the **vrrp track 1 decrement 11** command in interface VLAN 20 configuration mode. As soon as you do this, a syslog message is displayed on SW2, as follows:

```
%VRRP-6-STATECHANGE: V120 Grp 20 state Master -> Backup
```

On SW1, the following syslog message is displayed:

```
%VRRP-6-STATECHANGE: V120 Grp 20 state Backup -> Master
```

You now reissue the **tracert** command on PC1 to verify the first hop. It is now SW1, as shown in Example 8-58.

Example 8-58 *A Trace from PC1 Confirming That SW1 Is the First Hop (Master)*

```
C:\PC1>tracert 192.0.2.1
Tracing route to 192.0.2.1 over a maximum of 30 hops

 1      2 ms      2 ms      2 ms  10.1.1.65
...output omitted...
Trace complete.
```

Next you enable the interface between SW3 and SW2 with the **no shutdown** command and receive the following syslog message on SW2:

```
%TRACKING-5-STATE: 1 interface Gi1/0/2 line-protocol Down->Up
```

```
%VRRP-6-STATECHANGE: V120 Grp 20 state Backup -> Master
```

Because the interface is up, the tracking object is up, which means that SW2's priority goes back to 110, and SW2 becomes the virtual router master. You then reissue the **tracert** command on PC1 to verify the first hop. It is now SW2, as shown in Example 8-59.

Example 8-59 *A Trace from PC1 Confirming That SW2 Is the First Hop (Master)*

```
C:\PC1>tracert 192.0.2.1
Tracing route to 192.0.2.1 over a maximum of 30 hops

 1      2 ms      2 ms      2 ms  10.1.1.66
...output omitted...
Trace complete.
```

Troubleshooting GLBP

Whereas HSRP can only have one active forwarder for each group, Gateway Load Balancing Protocol (GLBP) can have multiple forwarders for each group. Therefore, GLBP can load balance traffic destined for a next-hop gateway across a collection of routers within the GLBP group.

This section explains the GLBP active virtual gateway (AVG) and active virtual forwarder (AVF) concepts and how to verify and troubleshoot issues related to GLBP.

Reviewing GLBP

Key Topic

With GLBP, there is one AVG and up to four AVFs in a group. The AVG is responsible for handing out the AVF MAC addresses to the hosts in the LAN. Therefore, it is responsible for replying to ARP requests for the MAC address of the default gateway. Note that the AVG is usually an AVF as well. The AVFs are responsible for processing the frames that are sent to their MAC address.

Figure 8-8 shows a GLBP topology example. R1 is the AVG, and R1 and R2 are AVFs. The virtual router IP address that will be used as the default gateway on all the hosts is 10.1.1.62. When Workstation A sends an ARP request for the MAC address of 10.1.1.62, R1 (AVG) responds with the MAC of 0007.b400.0a01. When Workstation B sends an ARP request for the MAC address of 10.1.1.62, R1 (AVG) responds with the MAC of 0007.b400.0a02. The next workstation that sends an ARP request will get 0007.b400.0a01 and then 0007.b400.0a02, and so on. This is the default behavior known as *round-robin*, which can be changed with the `glbp group_id load-balancing` interface configuration command. The other options are *host-dependent* and *weighted*.

As you can see from Figure 8-8, Workstation A sends default gateway destined traffic to R1, and Workstation B sends default gateway destined traffic to R2.

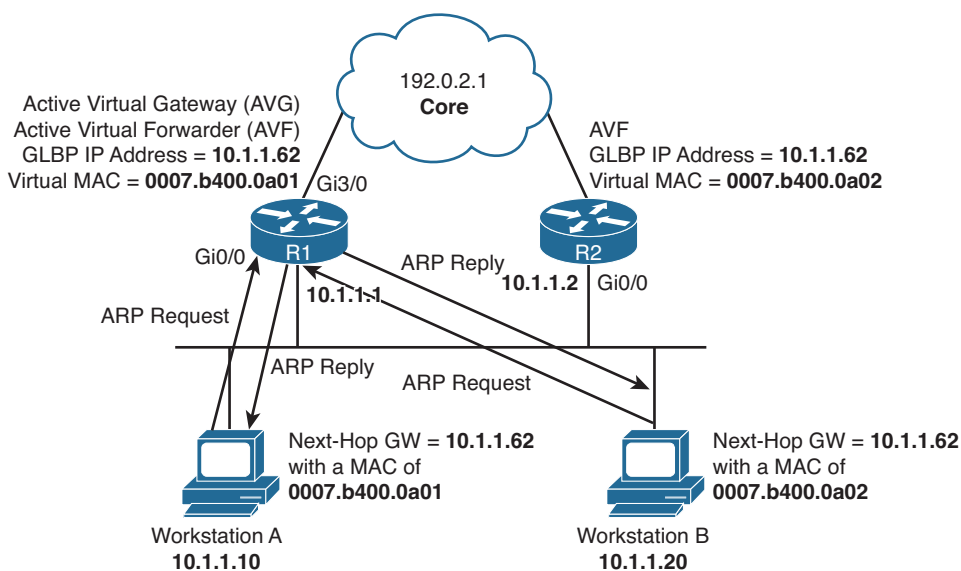


Figure 8-8 Basic GLBP Operation

Examples 8-60 and 8-61 show the possible GLBP configurations for routers R1 and R2.

Example 8-60 Possible GLBP Configuration on Router R1

```
R1#show run interface gigabitethernet 0/0
Building configuration...

Current configuration : 269 bytes
!
```

```

interface GigabitEthernet0/0
  ip address 10.1.1.1 255.255.255.192
  glbp 10 ip 10.1.1.62
  glbp 10 priority 150
  glbp 10 preempt
  glbp 10 weighting 110 lower 90 upper 100
  glbp 10 load-balancing weighted
end

```

Example 8-61 Possible GLBP Configuration on Router R2

```

R2#show run interface gigabitethernet 0/0
Building configuration...

Current configuration : 237 bytes
!
interface GigabitEthernet0/0
  ip address 10.1.1.2 255.255.255.192
  glbp 10 ip 10.1.1.62
  glbp 10 preempt
  glbp 10 weighting 100 lower 80
  glbp 10 load-balancing weighted
end

```

Notice that both routers R1 and R2 have been configured with the same virtual IP address of 10.1.1.62 for GLBP group 10. Router R1 is configured to be the AVG with a higher priority using the **glbp 10 priority 150** command. Router R2 has a default GLBP priority of 100, and with GLBP, higher-priority values are more preferable. Also, notice that both routers are configured with the **glbp 10 preempt** command. This ensures that the router with the higher priority will be the AVG. Remember that preemption is not enabled by default for the AVG election process.

The last two commands in Examples 8-60 and 8-61 relate to the AVFs and how their MAC addresses will be handed out to hosts on the LAN by the AVG, and whether they will be allowed to forward traffic. By default, the MACs will be handed out in a round-robin fashion. However, in these examples, load balancing has been configured to weighted. This means that the initial weighting value defined in the **glbp 10 weighting** command will determine the ratio that will be used to hand out MAC addresses. In this case, the AVG will hand out the MAC addresses in a 110:100 ratio, or 11:10 ratio. This means that R1's virtual MAC address will be given to clients 11 times for every 10 times that R2's virtual MAC address will be given out. Therefore, R1 will handle more hosts on average than R2. The lower and upper values are related to when the AVF will lose its ability to forward traffic for its virtual MAC address and when it will regain its ability to forward traffic for its virtual MAC address. Referring to Example 8-60 again, notice that R1's lower limit is 90. This means that R1 will lose its ability to forward traffic for its virtual MAC address if its weighting drops below 90. It will regain its ability to forward traffic for its virtual MAC address if its weighting goes back above 100. The initial weighting

value is 110. Notice that R2 in Example 8-61 has no upper weighting, which means that it is the same as the initial weighting.

GLBP Verification and Troubleshooting

When verifying a GLBP configuration or troubleshooting a GLBP issue, begin by determining the following information about the GLBP group under inspection:



- Which router is the AVG?
- Which routers are the AVFs?
- How was the AVG chosen?
- Which routers, if any, are configured with the preempt option?
- What is the IP address of the virtual router?
- What are the AVFs virtual MAC addresses?
- Is object tracking on?

The **show glbp brief** command displays a great deal of GLBP information. Examples 8-62 and 8-63 provide samples of the **show glbp brief** command. The output identifies the interfaces that are participating in a GLBP group. It identifies who the AVFs are under the Fwd column. The – refers to the AVG information, and the numbers 1 and 2 refer to the AVFs in the group. The Priority column is used to display the priority used during the AVG election process. The State column identifies the state of the device for the group. If it is the AVG row, in this case the top row, active means that it is the AVG, and standby means that it is waiting to become the AVG if the AVG fails. For the second and third rows, it is referring to the state of the AVF. In these examples, active means that the router is forwarding for the virtual MAC address in the Address column. Listen means that the router is waiting to take over the forwarding process for the virtual MAC address in the Address column if the router listed in the Active Router column is no longer able to forward traffic for the virtual MAC address.



Example 8-62 show glbp brief Command Output on Router R1

R1#show glbp brief							
Interface	Grp	Fwd	Pri	State	Address	Active router	Standby router
Gi0/0	10	-	150	Active	10.1.1.62	local	10.1.1.2
Gi0/0	10	1	-	Active	0007.b400.0a01	local	-
Gi0/0	10	2	-	Listen	0007.b400.0a02	10.1.1.2	-

Example 8-63 show glbp brief Command Output on Router R2

R2#show glbp brief							
Interface	Grp	Fwd	Pri	State	Address	Active router	Standby router
Gi0/0	10	-	100	Standby	10.1.1.62	10.1.1.1	local
Gi0/0	10	1	-	Listen	0007.b400.0a01	10.1.1.1	-
Gi0/0	10	2	-	Active	0007.b400.0a02	local	-

The **show glbp** command output provides significant details about the GLBP groups, as shown in Example 8-64. In the output, you can verify the group number and the interface associated with it. You can determine whether it is the AVG based on whether it is active or standby. The virtual IP address, the hello and hold timers, and the status of preemption is also listed. Depending on the state of the device, you will be able to verify the active or standby routers IP address and its priority. You will also be able to see your current local priority and the configured priority. This is a great command to verify the weighting values, the type of load balancing being used, and the members of the group, which are identified by their physical MAC address and IP address associated with the interface participating in the GLBP group.

Example 8-64 *show glbp Command Output on Router R1*

```
R1#show glbp gigabitethernet0/0
GigabitEthernet0/0 - Group 10
  State is Active
    1 state change, last state change 00:31:34
  Virtual IP address is 10.1.1.62
  Hello time 3 sec, hold time 10 sec
    Next hello sent in 1.568 secs
  Redirect time 600 sec, forwarder time-out 14400 sec
  Preemption enabled, min delay 0 sec
  Active is local
  Standby is 10.1.1.2, priority 100 (expires in 9.984 sec)
  Priority 150 (configured)
  Weighting 110 (configured 110), thresholds: lower 90, upper 100
    Track object 1 state Up decrement 25
  Load balancing: weighted
  Group members:
    ca12.0854.0008 (10.1.1.2)
    ca13.0854.0008 (10.1.1.1) local
  There are 2 forwarders (1 active)
  Forwarder 1
    State is Active
      3 state changes, last state change 00:03:35
    MAC address is 0007.b400.0a01 (default)
    Owner ID is ca13.0854.0008
    Redirection enabled
    Preemption enabled, min delay 30 sec
    Active is local, weighting 110
  Forwarder 2
    State is Listen
    MAC address is 0007.b400.0a02 (learnt)
    Owner ID is ca12.0854.0008
    Redirection enabled, 600.000 sec remaining (maximum 600 sec)
    Time to live: 14400.000 sec (maximum 14400 sec)
    Preemption enabled, min delay 30 sec
    Active is 10.1.1.2 (primary), weighting 100 (expires in 11.232 sec)
```

Still referring to Example 8-64, focus on the area related to the forwarders. This information is related to the AVFs in the group. In this case, you can verify that there are two AVFs. This router is currently active for Forwarder 1, meaning that it is forwarding for the MAC address 0007.b400.0a01 that is listed. It also states who the current owner is of the virtual MAC address, based on the physical MAC address of the device. The owner is the device currently responsible for forwarding traffic for the virtual MAC address. R1 is in the listen state for Forwarder 2, meaning that it is waiting for the current owner of the virtual MAC 0007.b400.0a02 to no longer be able to forward for the MAC so that it can take over.

Virtual Router MAC Addresses

The default virtual MAC address for the AVFs in a GLBP group, as shown in Figure 8-9, is based on the group number and the AVF forwarder ID within the group. Specifically, the virtual MAC address for a GLBP group begins with a well-known GLBP code of 0007.b400. The next two hexadecimal digits represent the group number. The last two hexadecimal digits represent the forwarder ID within the group. For example, a GLBP group of 43 yields a default virtual MAC address for AVF 1 of 0007.b400.2b01, because 43 in decimal equates to 2b in hexadecimal. For AVF 2, it would be 0007.b400.2b02, and for AVF 3, it would be 0007.b400.2b03.

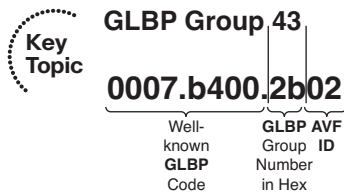


Figure 8-9 GLBP Virtual MAC Address

GLBP Object Tracking



As with VRRP, you can implement object tracking. By default, GLBP will only detect a failure of the device itself or the path that is used by the hello packets. That is perfectly fine for the AVG because a failure of an uplink outside the LAN will not affect the AVG because hello packets are still exchanged successfully, and the AVG is still reachable. However, what about the AVFs? If the uplinks fail, the AVF cannot forward packets for the virtual IP and MAC it owns. This is where object tracking comes into play for the AVFs. Object tracking allows you to control the weighting of an AVF in a GLBP group based on the status of an object. The object can be IP-related information such as a route, a group of objects, the status of an SLA probe, and the status of an interface. If the object is anything but up, the weight of the router can be decremented to a value that is lower than a configured threshold so that another AVF can forward on behalf of the router that cannot. You can use the **show glbp** command to verify whether object tracking is configured, as shown in Example 8-65, and the state of the tracked object.

Example 8-65 *show glbp Command Output on Router R1*

```

R1#show glbp
GigabitEthernet0/0 - Group 10
  State is Active
    1 state change, last state change 00:31:34
  Virtual IP address is 10.1.1.62
  Hello time 3 sec, hold time 10 sec
    Next hello sent in 1.568 secs
  Redirect time 600 sec, forwarder time-out 14400 sec
  Preemption enabled, min delay 0 sec
  Active is local
  Standby is 10.1.1.2, priority 100 (expires in 9.984 sec)
  Priority 150 (configured)
  Weighting 110 (configured 110), thresholds: lower 90, upper 100
    Track object 1 state Up decrement 25
  Load balancing: weighted
  Group members:
    ca12.0854.0008 (10.1.1.2)
    ca13.0854.0008 (10.1.1.1) local
  There are 2 forwarders (1 active)
  Forwarder 1
    State is Active
      3 state changes, last state change 00:03:35
    MAC address is 0007.b400.0a01 (default)
    Owner ID is ca13.0854.0008
    Redirection enabled
    Preemption enabled, min delay 30 sec
    Active is local, weighting 110
  Forwarder 2
    State is Listen
    MAC address is 0007.b400.0a02 (learnt)
    Owner ID is ca12.0854.0008
    Redirection enabled, 600.000 sec remaining (maximum 600 sec)
    Time to live: 14400.000 sec (maximum 14400 sec)
    Preemption enabled, min delay 30 sec
    Active is 10.1.1.2 (primary), weighting 100 (expires in 11.232 sec)

```

In the case of Example 8-65, you can see that the tracked object 1 is in a state of up. However, if the tracked object goes down, the weighting will be decremented by 25, and as a result, the weighting will be lower than the lower threshold of 90 and R1 will no longer be able to be the AVF for MAC 0007b400.0a01. AVF2, which is R2, will have to forward for both MAC addresses at this point.

However, if you need to find out what the tracked object is specifically so that you can troubleshoot further, use the command **show track**, as shown in Example 8-66. In this output, the line protocol of interface Gigabit Ethernet 3/0 is being tracked by GLBP.

Example 8-66 *show track Command Output on R1*

```
R1#show track
Track 1
  Interface GigabitEthernet3/0 line-protocol
  Line protocol is Up
    3 changes, last change 00:05:56
  Tracked by:
    GLBP GigabitEthernet0/0 10
```

Verifying GLBP First Hop

Once you know the current GLBP configuration, you might then check to see whether a host on the GLBP virtual IP address's subnet can ping the virtual IP address. Based on the topology previously shown in Figure 8-8, Example 8-67 shows a successful ping from Workstation A.

Example 8-67 *Ping Test from Workstation A to the GLBP Virtual IP Address*

```
C:\>ping 10.1.1.62

Pinging 10.1.1.62 with 32 bytes of data:

Reply from 10.1.1.62: bytes=32 time=2ms TTL=255
Reply from 10.1.1.62: bytes=32 time=1ms TTL=255
Reply from 10.1.1.62: bytes=32 time=1ms TTL=255
Reply from 10.1.1.62: bytes=32 time=1ms TTL=255

Ping statistics for 10.1.1.62:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 2ms, Average = 1ms
```

However, that does not prove that we are using the virtual MAC address and GLBP successfully. Therefore, from the client, you should also verify that the virtual MAC address learned by the client corresponds to the virtual MAC address reported by the GLBP AVG. Example 8-68 shows Workstation A's ARP cache entry for the GLBP virtual IP address of 10.1.1.62. Notice in the output that the MAC address learned via ARP does match the GLBP virtual MAC address of the first AVE.

Example 8-68 *Workstation A's ARP Cache*

```
C:\>arp -a

Interface: 10.1.1.10 --- 0x4

    Internet Address      Physical Address      Type
    10.1.1.62            00-07-b4-00-0a-01    dynamic
```

However, as discussed with HSRP and VRRP, one of the best tools to use with FHRPs to verify the path is traceroute. With traceroute, you can identify the physical first-hop router that the packets are traversing. Example 8-69 displays the **tracert** command executed on Workstation A. Notice that it states that the first hop is 10.1.1.1. This is the IP address of R1's Gig0/0 interface. Example 8-70 displays the **tracert** command executed on Workstation B. Notice that it states that the first hop is 10.1.1.2. This is the IP address on R2's Gig0/0 interface. But remember in both cases they are configured to use the virtual IP 10.1.1.62 and are dynamically provided a virtual MAC address based on the AVG load-balancing method.

Example 8-69 *A Trace from Workstation A Confirming That R1 Is the First Hop (AVF)*

```
C:\>tracert 192.0.2.1
Tracing route to 192.0.2.1 over a maximum of 30 hops

  1    2 ms    2 ms    2 ms  10.1.1.1
...output omitted...
Trace complete.
```

Example 8-70 *A Trace from Workstation B Confirming That R2 Is the First Hop (AVF)*

```
C:\>tracert 192.0.2.1
Tracing route to 192.0.2.1 over a maximum of 30 hops

  1    2 ms    2 ms    2 ms  10.1.1.2
...output omitted...
Trace complete.
```

GLBP Trouble Tickets

This section presents various trouble tickets relating to the topics discussed earlier in the chapter. The purpose of these trouble tickets is to give a process that you can follow when troubleshooting in the real world or in an exam environment. All trouble tickets in this section are based on the topology depicted in Figure 8-10.

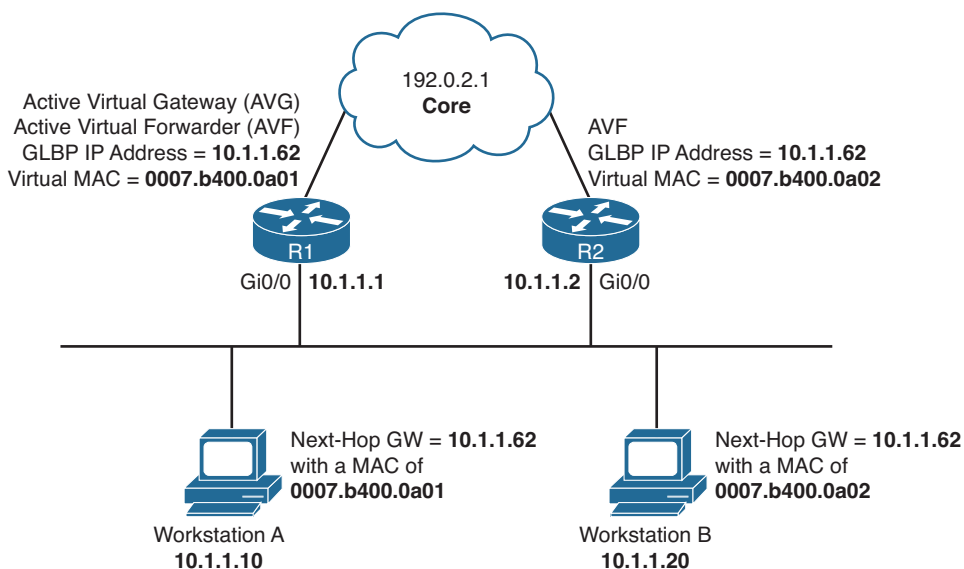


Figure 8-10 GLBP Trouble Ticker Topology

Trouble Ticket 8-6

Problem: A junior administrator has stated that GLBP is behaving strangely.

With a puzzled look on your face, you ask the junior admin to show you what she means. The junior admin provides the output shown in Example 8-71 and Example 8-72. You review them, and then ask the junior administrator to explain.

Example 8-71 Output of show glbp brief on R1

```
R1#show glbp brief
```

Interface	Grp	Fwd	Pri	State	Address	Active router	Standby router
Gi0/0	10	-	150	Active	10.1.1.62	local	unknown
Gi0/0	10	1	-	Active	0007.b400.0a01	local	-
Gi0/0	10	2	-	Active	0007.b400.0a02	local	-

Example 8-72 Output of show glbp brief on R2

```
R2#show glbp brief
```

Interface	Grp	Fwd	Pri	State	Address	Active router	Standby router
Gi0/0	10	-	100	Active	10.1.1.62	local	unknown
Gi0/0	10	1	-	Active	0007.b400.0a01	local	-
Gi0/0	10	2	-	Active	0007.b400.0a02	local	-

The junior administrator indicates that R1 and R2 are both in group 10. R1 has a priority of 150, and R2 has a priority of 100. However, they are both indicating that they are the AVG for the virtual address 10.1.1.62. In addition, R1 and R2 are both stating that they are the AVFs for the MAC addresses listed.

You then ask the junior admin, “Why would they both consider themselves as the AVG and AVFs?” The junior admin replies, “They don’t know each other is on the LAN and participating in GLBP group 10.” You grin and state, “That is correct. Now find out why!”

The junior admin issues the command **show glbp** on R1 and R2, as displayed in Examples 8-73 and 8-74, and reviews them hoping to *spot the difference*. The output confirms that both R1 and R2 are the AVG for group 10 because it states “State is Active” near the top. The virtual IP is the same at 10.1.1.62. The timers are the same, although they do not have to be as long as they do not cause flapping neighbor relationships. At this point, the junior admin spots the difference. Do you see it?

Example 8-73 Output of show glbp on R1

```
R1#show glbp brief
GigabitEthernet0/0 - Group 10
  State is Active
    3 state changes, last state change 00:23:29
  Virtual IP address is 10.1.1.62
  Hello time 3 sec, hold time 10 sec
    Next hello sent in 0.288 secs
  Redirect time 600 sec, forwarder time-out 14400 sec
  Authentication text, string "TSHOOT"
  Preemption enabled, min delay 0 sec
  Active is local
  Standby is unknown
  Priority 150 (configured)
  ...output omitted...
```

Example 8-74 Output of show glbp on R2

```
R2#show glbp brief
GigabitEthernet0/0 - Group 10
  State is Active
    8 state changes, last state change 00:21:32
  Virtual IP address is 10.1.1.62
  Hello time 3 sec, hold time 10 sec
    Next hello sent in 2.592 secs
  Redirect time 600 sec, forwarder time-out 14400 sec
  Authentication MD5, key-string
  Preemption enabled, min delay 0 sec
  Active is local
  Standby is unknown
  Priority 100 (default)
  ... output omitted...
```

R1 is using plain-text GLBP authentication, and R2 is using message digest 5 (MD5) GLBP authentication. They are both using authentication, but the type of authentication does not match. Therefore, they know each other is there, but because they cannot

authenticate each other, they consider each other to be rogue GLBP devices and will not accept the GLBP information from each other.

Your security policy states to use MD5 authentication, so you change R1 with the command **glbp 10 authentication md5 key-string TSHOOT** in interface configuration mode. You then check the output of **show glbp brief** on R1 and R2, as shown in Examples 8-75 and 8-76, to verify whether the output has changed. It has. R1 is the AVG and AVF for the first MAC, and R2 is standby and the AVF for the second MAC.

Example 8-75 Output of show glbp brief on R1

R1#show glbp brief							
Interface	Grp	Fwd	Pri	State	Address	Active router	Standby router
Gi0/0	10	-	150	Active	10.1.1.62	local	10.1.1.2
Gi0/0	10	1	-	Active	0007.b400.0a01	local	-
Gi0/0	10	2	-	Listen	0007.b400.0a02	10.1.1.2	-

Example 8-76 Output of show glbp brief on R2

R2#show glbp brief							
Interface	Grp	Fwd	Pri	State	Address	Active router	Standby router
Gi0/0	10	-	100	Standby	10.1.1.62	10.1.1.1	local
Gi0/0	10	1	-	Listen	0007.b400.0a01	10.1.1.1	-
Gi0/0	10	2	-	Active	0007.b400.0a02	local	-

Trouble Ticket 8-7

Problem: The uplink has failed between R2 and the core; however, R2 is still the AVF for MAC 0007.b400.0a02 when it should be R1.

Let's *shoot from the hip* this time!

Brainstorm: Uplink failed + R2 still AVF when it should not be = object tracking and weight issue?

Let's use the **show glbp** command to see what the weight of R2 is and whether object tracking is enabled. Example 8-77 displays the output of **show glbp** on R2, and it clearly indicates that we are tracking object 1, which is down, and when it is down, the weighting will be decremented by 20, which it has been because the configured weight is 100 and the current weight is 80. However, R2's weighting still has not passed the lower threshold. Therefore, it will still be the AVF for the MAC address assigned to it by the AVG.

Example 8-77 Output of show glbp on R2

R2#show glbp	
GigabitEthernet0/0 - Group 10	
State is Standby	
10 state changes, last state change 00:20:58	
Virtual IP address is 10.1.1.62	

```

Hello time 3 sec, hold time 10 sec
  Next hello sent in 0.736 secs
Redirect time 600 sec, forwarder time-out 14400 sec
Authentication MD5, key-string
Preemption enabled, min delay 0 sec
Active is 10.1.1.1, priority 150 (expires in 8.480 sec)
Standby is local
Priority 100 (default)
Weighting 80 (configured 100), thresholds: lower 80, upper 100
  Track object 1 state Down decrement 20
Load balancing: weighted

```

To solve this problem, you need to modify the **glbp 10 weighting track 1** command so that the decrement is greater than 20 (for example, **glbp 10 weighting track 1 decrement 21**). When you do so, R1 will be the AVF for both MACs. On R1, you can confirm this with the **show glbp brief** command, as shown in Example 8-78.

Example 8-78 *Output of show glbp brief on R1*

```

R1#show glbp brief
Interface  Grp  Fwd Pri State      Address          Active router    Standby router
Gi0/0      10   -   150 Active  10.1.1.62       local            10.1.1.2
Gi0/0      10   1   -   Active  0007.b400.0a01  local            -
Gi0/0      10   2   -   Active  0007.b400.0a02  local            -

```

Comparing HSRP, VRRP, and GLBP

As you have witnessed in this chapter, HSRP, VRRP, and GLBP are very similar. The output provided by the **show** commands is similar as well. It is important to note that the issues will be similar with these FHRPs, making them easy to troubleshoot for most. However, although HSRP, VRRP, and GLBP have commonalities, it is important for you as a troubleshooter to understand the differences to make sure that you are troubleshooting as efficiently as possible. Table 8-2 compares several characteristics of these FHRPs.

Table 8-2 *Comparing HSRP, VRRP, and GLBP*

Characteristic	HSRP	VRRP	GLBP
Cisco proprietary.	Yes	No	Yes
Interface IP address can act as virtual IP address.	No	Yes	No
More than one router in a group can simultaneously forward traffic for that group.	No	No	Yes
Hello timer default value.	3 seconds	1 second	3 seconds



Characteristic	HSRP	VRRP	GLBP
Hold timer default value.	10 seconds	3 seconds	10 seconds
Preemption enabled by default.	No	Yes	No for AVG, Yes for AVFs
Default priority.	100	100	100
Default weight.	—	—	100
Authentication supported.	Yes	Yes	Yes
Multicast address.	224.0.0.2	224.0.0.18	224.0.0.102
Virtual MAC address. (xx = group number)(yy = AVF)	V1: 0000.0c07.acxx V2: 0000.0c9f.fxxx	0000.5e00.01xx	0007.b400.xxyy

Exam Preparation Tasks

As mentioned in the section “How to Use This Book” in the Introduction, you have a couple of choices for exam preparation: the exercises here; Chapter 22, “Final Preparation;” and the exam simulation questions on the CD-ROM.

Review All Key Topics

Review the most important topics in this chapter, noted with the Key Topic icon in the outer margin of the page. Table 8-3 lists a reference of these key topics and the page numbers on which each is found.



Table 8-3 *Key Topics for Chapter 8*

Key Topic Element	Description	Page Number
Paragraph	Describes how to configure an HSRP group and explains priority and preempt	291
List	Identifies HSRP parameters that should be verified while troubleshooting HSRP issues	292
Example 8-3	show standby brief command output on Router R1	292
Figure 8-2	HSRPv1 virtual MAC address	293
Section	Interface tracking	293
Section	Verifying first hop	294
Section	Reviewing VRRP	306
List	Identifies VRRP parameters that should be verified while troubleshooting issues	308
Example 8-36	show vrrp brief command output on router SW1	308
Figure 8-5	VRRP virtual MAC address	309
Section	Object tracking	309
Section	Verifying first hop	310
Section	Reviewing GLBP	319
List	Identifies GLBP parameters that should be verified while troubleshooting HSRP issues	321
Example 8-62	show glbp brief command output on router SW1	321
Figure 8-9	GLBP virtual MAC address	323
Section	GLBP object tracking	323
Section	Verifying GLBP first hop	325
Table 8-2	Comparing HSRP, VRRP, and GLBP	330

Define Key Terms

Define the following key terms from this chapter and check your answers in the glossary:

HSRP, VRRP, GLBP, priority, preempt, interface tracking, object tracking, virtual router, virtual MAC address, active forwarder, standby router, virtual master router, virtual router backup, AVG, AVF, weighting

Complete Tables and Lists from Memory

Print a copy of Appendix C, “Memory Tables,” (found on the disc), or at least the section for this chapter, and complete the tables and lists from memory. Appendix D, “Memory Tables Answer Key,” also on the disc, includes completed tables and lists to check your work.

Command Reference to Check Your Memory

This section includes the most important **show** commands covered in this chapter. It might not be necessary to memorize the complete syntax of every command, but you should be able to remember the basic keywords that are needed.

To test your memory of the commands, cover the right side of Table 8-4 with a piece of paper, read the description on the left side, and then see how much of the command you can remember.

The 300-135 TSHOOT exam focuses on practical, hands-on skills that are used by a networking professional. Therefore, you should be able to identify the commands needed to troubleshoot issues related to the topics covered in this chapter.

Table 8-4 *show commands*

Task	Command Syntax
Displays a summary of the HSRP standby group configuration on a switch or router	show standby brief
Displays details of the HSRP standby group configuration on a switch or router interface, including timers and tracked interfaces or objects	show standby interface_type interface_number
Displays the commands configured on a router or switch interface	show run interface_type interface_number
Displays a summary of the VRRP group configuration on a switch or router	show vrrp brief
Displays details of the VRRP group configuration on a switch or router interface, including timers and tracked objects	show vrrp interface_type interface_number
Displays the tracking objects configured on the router or switch	show track
Displays a summary of the GLBP group configuration on a switch or router	show glbp brief
Displays details of the GLBP group configuration on a router interface, including timers, who the AVG is, who the AVFs are, in addition to tracked objects	show glbp interface_type interface_number



This chapter covers the following topics:

- **Troubleshooting IPv4 Addressing:** This section focuses on how you can verify that devices are addressed correctly in the network during your troubleshooting process.
- **Troubleshooting DHCP for IPv4:** This section reviews the DHCP for IPv4 operations and identifies how you can successfully troubleshoot DHCP related issues.
- **Troubleshooting NAT:** This section explains the reasons why NAT may not be translating addresses and how to recognize them.
- **IPv4 Addressing and Addressing Technologies Trouble Tickets:** This section provides trouble tickets that demonstrate how you can use a structured troubleshooting process to solve a reported problem.

Troubleshooting IPv4 Addressing and Addressing Technologies

Although IPv6 is currently being deployed, it is being done at a slow pace. Therefore, most networks are still relying on IPv4, and many new networks and network additions are being deployed with IPv4. Therefore, as a troubleshooter, you need the skills necessary to successfully identify issues related to improper IPv4 addressing on devices. It might be a bad address, subnet mask, or even the address of the default gateway.

Typically, when deploying IPv4 addresses, Dynamic Host Configuration Protocol (DHCP) will be used so that they can be dynamically assigned. However, with this dynamic process, issues may arise that prevent a device from successfully obtaining an IPv4 address from the DHCP server. Therefore, you need a solid understanding of how DHCP operates and how to identify the issues that would prevent a client from obtaining an IP address from a DHCP server.

Because RFC 1918 addresses are not routable on the Internet, Network Address Translation (NAT) is needed to translate IPv4 private addresses to public addresses that are routable on the Internet. This adds another bit of complexity to the environment that you need to know how to troubleshoot so that devices can access resources external to the organization.

This chapter covers the different methods that you can use to troubleshoot IPv4 addressing issues, DHCP for IPv4-related issues, and NAT issues.

“Do I Know This Already?” Quiz

The “Do I Know This Already?” quiz allows you to assess whether you should read this entire chapter thoroughly or jump to the “Exam Preparation Tasks” section. If you are in doubt about your answers to these questions or your own assessment of your knowledge of the topics, read the entire chapter. Table 9-1 lists the major headings in this chapter and their corresponding “Do I Know This Already?” quiz questions. You can find the answers in Appendix A, “Answers to the ‘Do I Know This Already?’ Quizzes.”

Table 9-1 “Do I Know This Already?” Section-to-Question Mapping

Foundation Topics Section	Questions
Troubleshooting IPv4 Addressing	1–4
Troubleshooting DHCP for IPv4	5–7
Troubleshooting NAT	8–10

Caution The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you do not know the answer to a question or are only partially sure of the answer, you should mark that question as wrong for purposes of the self-assessment. Giving yourself credit for an answer that you correctly guess skews your self-assessment results and might provide you with a false sense of security.

1. What will occur when a PC with the IP address 10.1.1.27/28 needs to communicate with a PC that has an IP address of 10.1.1.18? (Choose two answers.)
 - a. It will send the frame to its default gateway.
 - b. It will send the frame directly to the destination PC.
 - c. It will ARP for the MAC address of the default gateway.
 - d. It will ARP for the MAC address of the destination PC.
2. What will occur when a PC with the IP address 10.1.1.27/29 needs to communicate with a PC that has an IP address of 10.1.1.18? (Choose two answers.)
 - a. It will send the frame to its default gateway.
 - b. It will send the frame directly to the destination PC.
 - c. It will ARP for the MAC address of the default gateway.
 - d. It will ARP for the MAC address of the destination PC.
3. Which command enables you to verify the IP address configured on a Windows PC interface?
 - a. `ipconfig`
 - b. `show ip interface`
 - c. `arp -a`
 - d. `show ip arp`
4. Which command enables you to verify the IP address configured on a router's interface?
 - a. `ipconfig`
 - b. `show ip interface`
 - c. `arp -a`
 - d. `show ip arp`

5. What is the correct order of operations for the DHCP for IPv4 process?
 - a. Offer, Request, Ack, Discover
 - b. Discover, Request, Ack, Offer
 - c. Request, Offer, Discover, Ack
 - d. Discover, Offer, Request, Ack
6. Which command is needed on a router interface to forward DHCP Discover messages to a DHCP server on a different subnet?
 - a. `ip address dhcp`
 - b. `ip helper-address`
 - c. `ip dhcp-forwarder`
 - d. `ip dhcp server`
7. Which command will enable a router interface to obtain an IP address from a DHCP server?
 - a. `ip dhcp client`
 - b. `ip dhcp server`
 - c. `ip address dhcp`
 - d. `ip helper-address`
8. Which parameter is necessary in the `ip nat inside source` command to enable PAT?
 - a. `pat`
 - b. `list`
 - c. `overload`
 - d. `private`
9. Which command enables you to verify the interfaces that are configured for NAT?
 - a. `show ip nat translations`
 - b. `show ip nat statistics`
 - c. `show ip nat interfaces`
 - d. `show ip nat`
10. Which column in the output of `show ip nat translations` displays the address that source IPs have been translated to?
 - a. Inside Local
 - b. Inside Global
 - c. Outside Local
 - d. Outside Global

Foundation Topics

Troubleshooting IPv4 Addressing

Just like your personal street address uniquely defines where you live, an IPv4 address uniquely defines where a device resides in a network. Your street address is made of two parts, the street name and the number of your residence; and the combination of these will be unique within your city/town. As a result, the pizza delivery person is able to drop off your pizza at your house in 30 minutes or it is free. If your house is addressed incorrectly, you may or may not get your pizza, and we do not want that to happen.

The same is true with IPv4 addressing. If devices are addressed incorrectly, they may or may not receive the packets that are intended for them. Therefore, it is imperative that you have a solid understanding of IPv4 addressing and how to verify that devices are addressed correctly on the network.

This section focuses on how we can troubleshoot IPv4 addressing issues.

IPv4 Addressing Issues

An IPv4 address is made up of two parts: a network/subnet portion and a host portion. It is imperative that all devices in the same network/subnet share the exact same network/subnet portion. If they are not exactly the same, the PC could end up addressing the Layer 2 frame incorrectly and sending the packet in the wrong direction. Refer to Figure 9-1, which shows a sample subnet (10.1.1.0/26) with two PCs and their default gateway, R1.

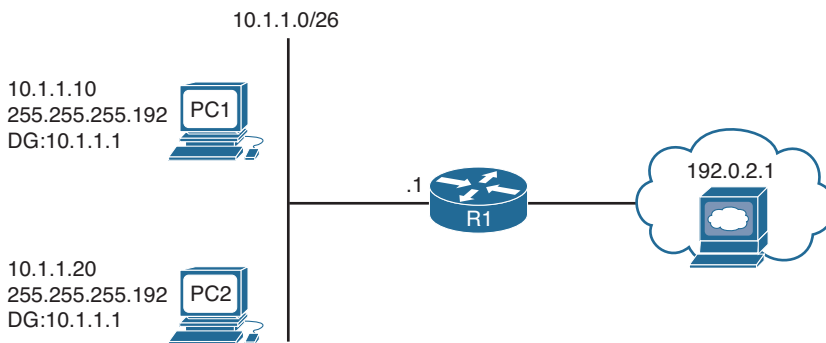


Figure 9-1 Correct IPv4 Addressing Example



When PC1 needs to communicate with PC2, it does a DNS lookup for the IP address of PC2. The IP address 10.1.1.20 is returned. Now PC1 needs to determine whether PC2 is

located in the same subnet because this will determine whether the frame will have the MAC of PC2 or the MAC of the default gateway (DG). PC1 determines its network/subnet portion by comparing its IP address to its subnet mask in binary as follows:

```
00001010.00000001.00000001.00001010 - PC1 IP address in binary
11111111.11111111.11111111.11000000 - PC1 subnet mask in binary
-----
00001010.00000001.00000001.00 - PC1 network/subnet ID
```

(The 1s in the subnet mask identify the network portion.)

Now PC1 compares the exact same binary bits to those binary bits in PC2's address as follows:

```
00001010.00000001.00000001.00 - PC1 network/subnet ID
00001010.00000001.00000001.00010100 - PC2 IP address in binary
```

Because the binary bits are the same, PC1 concludes that PC2 is in the same network/subnet; therefore, it can communicate directly with it and does not need to send the data to its default gateway. PC1 will create a frame with its own source MAC address and the MAC of PC2 as the destination.

Consider what occurs when PC1 needs to communicate with the web server at 192.0.2.1. It does a DNS lookup for the IP address of the web server. The IP address 192.0.2.1 is returned. Now PC1 needs to determine whether the web server is located in the same network/subnet. This will determine whether the frame will have the MAC of the web server or the MAC of the DG. PC1 determines its network/subnet portion by comparing its IP address to its subnet mask in binary as follows:

```
00001010.00000001.00000001.00001010 - PC1 IP address in binary
11111111.11111111.11111111.11000000 - PC1 subnet mask in binary
-----
00001010.00000001.00000001.00 - PC1 network/subnet ID
```

(The 1s in the subnet mask identify the network portion.)

Now PC1 compares the exact same binary bits to those binary bits in the web server address as follows:

```
00001010.00000001.00000001.00 - PC1 network/subnet ID
11000000.00000000.00000010.00000001 - web server IP address in binary
```

PC1 concludes that the web server is in a different network/subnet, because the bits are not the same; therefore, to communicate with the web server, it needs to send the data to its default gateway. PC1 will create a frame with its own source MAC address and the MAC of R1 as the destination.

As you can see, accurate IP addressing is paramount for successful communication. Let's see what happens if PC1 is configured with the wrong subnet mask (255.255.255.240), as shown in Figure 9-2.

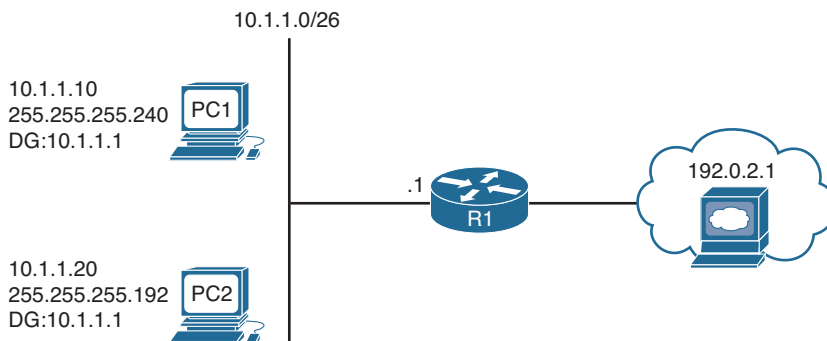


Figure 9-2 *Incorrect IPv4 Addressing Example*

Key Topic

PC1 determines its network/subnet portion by comparing its IP address to its subnet mask in binary as follows:

```
00001010.00000001.00000001.00001010 - PC1 IP address in binary
11111111.11111111.11111111.11110000 - PC1 subnet mask in binary
-----
00001010.00000001.00000001.0000 - PC1 network/subnet ID
```

Now PC1 compares the exact same binary bits to those binary bits in PC2's address as follows:

```
00001010.00000001.00000001.0000 - PC1 network/subnet ID
00001010.00000001.00000001.00010100 - PC2 IP address in binary
```

PC1 concludes that PC2 is not in the same network/subnet, because the binary bits are not exactly the same. Therefore, it cannot communicate directly with it and will need to send the frame to the router so that the router can route the packet to the subnet PC2 is in. However, the PCs are actually connected to the same subnet, and as a result we have an IPv4 addressing and connectivity issue.

Not only will an *improper subnet mask* cause issues, but an *inappropriate IP address combined with the correct subnet mask* will also cause issues. In addition, if the *default gateway is not configured correctly* on the PCs, packets will not be forwarded to the correct device when packets need to be sent to a different subnet.

As a troubleshooter, you need to be able to recognize these issues, or eliminate them as a possible issue quickly. You can verify the IP addressing information on a Windows PC using the **ipconfig** command and on a router or switch using the **show ip interface interface_type interface_number** command, as shown in Example 9-1.

Key Topic

Example 9-1 *Verifying IP Addressing on a PC and on a Router*

```
C:\>ipconfig

Windows IP Configuration

Ethernet adapter PC1:

    Connection-specific DNS Suffix  . :
```

```

IP Address. . . . . : 10.1.1.10
Subnet Mask . . . . . : 255.255.255.192
IP Address. . . . . : 2001:10::10
IP Address. . . . . : fe80::a00:27ff:fe5d:6d6%4
Default Gateway . . . . . : 10.1.1.1

```

```

R1#show ip interface gigabitEthernet 1/0
GigabitEthernet1/0 is up, line protocol is up
Internet address is 10.1.1.1/26
...output omitted..

```

Determining IP Addresses Within a Subnet

You want to be quick! Here is a quick way to determine all the IP addresses that will be in a particular subnet. Refer to Figure 9-3 as you are exploring this method.

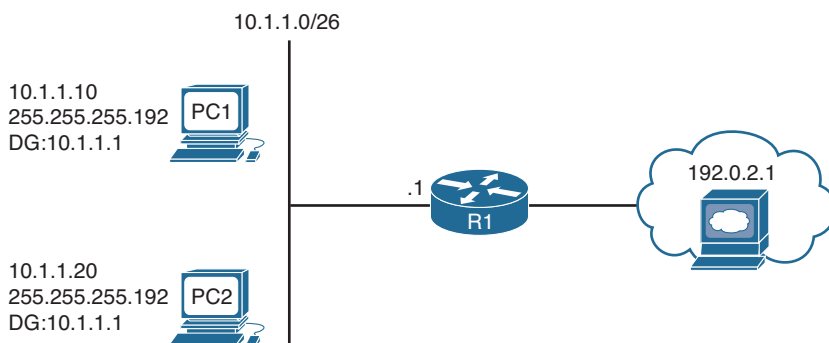


Figure 9-3 *Determining IP Address Within a Subnet*

Key Topic

Take the subnet mask and find the most interesting octet. This is where the last binary 1 would be. In this case, 255.255.255.192 would have the last binary 1 in the fourth octet, which is 192.

Now, take 256 and subtract 192 from it. The result is 64. The number 64 represents the block size or the total number of addresses in that subnet. Our subnet in this case is 10.1.1.0/26, and because the block size is 64, this subnet would begin at 10.1.1.0/26 and end at 10.1.1.63/26, which is a total of 64 addresses. The next subnet would be 10.1.1.64/26 to 10.1.1.127/26. The third subnet would be 10.1.1.128/26 to 10.1.1.191/26 and so on.

Now you can compare the addresses of devices with the subnet ranges you just identified. In this case, PC1, PC2, and an interface on R1 are supposed to be in the same subnet. As a result, they better all be addressed correctly or communication will not occur correctly. For example, if you are reviewing the output of **ipconfig** on PC1, as shown in Example 9-2, now that you have the ranges, you can easily see that PC1 is not in the same subnet as R1 and PC2. Although they have the same subnet mask, in this case PC1 falls in the range 10.1.1.64/26 to 10.1.1.127/26, whereas PC2 and the default gateway fall in the

range 10.1.1.0/26 to 10.1.1.63/26. PC1 is in a different network/subnet, when it should be in the same according to Figure 9-3. You will have to fix the address on PC1 so that it is within the correct network/subnet.

Example 9-2 Verifying IP Addressing on PC with the ipconfig Command

```
C:\>ipconfig

Windows IP Configuration

Ethernet adapter PC1:

    Connection-specific DNS Suffix  . : 
    IP Address. . . . . : 10.1.1.74
    Subnet Mask . . . . . : 255.255.255.192
    IP Address. . . . . : 2001:10::10
    IP Address. . . . . : fe80::a00:27ff:fe5d:6d6%4
    Default Gateway . . . . . : 10.1.1.1
```

Troubleshooting DHCP for IPv4

Dynamic Host Configuration Protocol (DHCP) serves as one of the most common methods of assigning IPv4 address information to a network host. Specifically, DHCP allows a DHCP client to obtain an IP address, subnet mask, default gateway IP address, DNS server IP address, and other types of IP address information from a DHCP server. The DHCP server can be local within the subnet, in a remote subnet, or the same device that is also the default gateway.

Because it is the most common way to deploy IPv4 addresses, you need to be well versed in the DHCP process and able to recognize issues related to DHCP. This section explains how DHCP operates and focuses on how to identify DHCP-related issues.

Reviewing DHCP Operations

If you have a cable modem, digital subscriber line (DSL), or fiber connection in your home, your router more than likely obtains its IP address from your service provider via DHCP. The router is also acting as a DHCP server for the devices in your home. In corporate networks, when a PC boots, that PC receives its IP address configuration information from a corporate DHCP server. Figure 9-4 illustrates the exchange of messages (Discover, Offer, Request, Acknowledgment [DORA] process) that occur as a DHCP client obtains IP address information from a DHCP server.

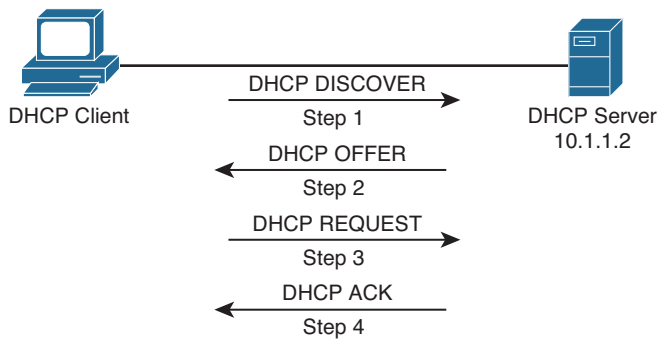


Figure 9-4 *DHCP DORA Process*



- Step 1.** When a DHCP client initially boots, it has no IP address, default gateway, or other such configuration information. Therefore, the way a DHCP client initially communicates is by sending a broadcast message (that is, a DHCPDISCOVER message) to a destination IP address of 255.255.255.255 and a destination MAC address of FFFF:FFFF:FFFF in an attempt to discover a DHCP server. The source IP address will be 0.0.0.0, and the source MAC address will be the MAC address of the sending device.
- Step 2.** When a DHCP server receives a DHCPDISCOVER message, it can respond with a DHCPOFFER message with an unleased IP address, subnet mask, and default gateway information. Because the DHCPDISCOVER message is sent as a broadcast, more than one DHCP server might respond to this Discover message with a DHCPOFFER. However, the client typically selects the server that sent the first DHCPOFFER response it received.
- Step 3.** The DHCP client communicates with the selected server by sending a broadcasted DHCPREQUEST message indicating that it will be using the address provided in the DHCPOFFER and as a result wants the associated address leased to itself.
- Step 4.** Finally, the DHCP server responds to the client with a DHCPACK message indicating that the IP address is leased to the client and includes any additional DHCP options that might be needed at this point.

Notice that in Step 1 the DHCPDISCOVER message was sent as a broadcast. The broadcast cannot cross a router boundary. Therefore, if a client resides on a different network than the DHCP server, the default gateway of the client should be configured as a DHCP relay agent to forward the broadcast packets as unicast packets to the server. You can use the `ip helper-address ip_address` interface configuration mode command to configure a router to relay DHCP messages to a DHCP server in the organization.

To illustrate, consider Figure 9-5 and Example 9-3. In the figure, the DHCP client belongs to the 172.16.1.0/24 network, whereas the DHCP server belongs to the 10.1.1.0/24 network. Router R1 is configured as a DHCP relay agent by using the syntax shown in Example 9-3.

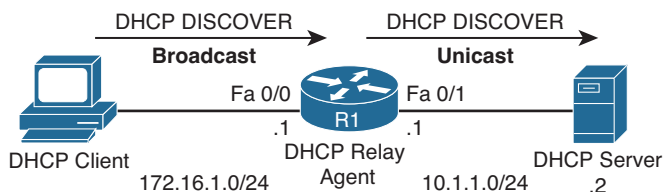


Figure 9-5 *DHCP Relay Agent*

**Key
Topic**

Example 9-3 *DHCP Relay Agent Configuration*

```
R1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#service dhcp
R1(config)#interface fa 0/0
R1(config-if)#ip helper-address 10.1.1.2
```

In the configuration, notice the **service dhcp** command. This command enables the DHCP service on the router, which must be enabled for the DHCP services to function. This command is usually not required because the DHCP service is enabled by default; however, when troubleshooting a DHCP relay agent issue, you might want to confirm that the service is enabled. Also, the **ip helper-address 10.1.1.2** command specifies the IP address of the DHCP server. If the wrong IP address is specified, the DHCP messages will be relayed to the wrong device. In addition, the **ip helper-address** command must be configured on the interface that is receiving the DHCPDISCOVER messages from the clients. If not, the router cannot relay the DHCP messages.

When you configure a router to act as a DHCP relay agent, realize that it relays a few other broadcast types in addition to a DHCP message. Other protocols that are forwarded by a DHCP relay agent include the following:

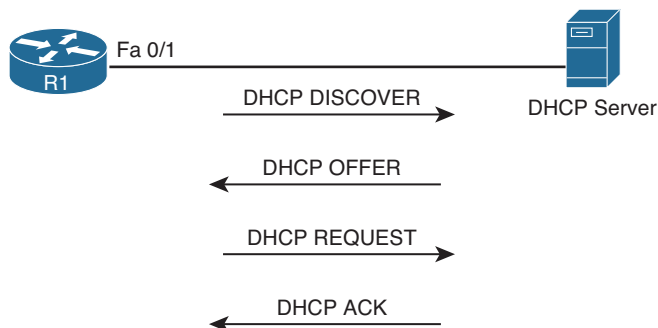
- TFTP
- Domain Name System (DNS)
- Internet Time Service (ITS)
- NetBIOS name server
- NetBIOS datagram server
- BootP
- TACACS

As a reference, Table 9-2 provides a comprehensive listing of DHCP message types you might encounter while troubleshooting a DHCP issue.

Table 9-2 *DHCP Message Types*

DHCP Message	Description
DHCPDISCOVER	A client sends this message in an attempt to locate a DHCP server. This message is sent to a broadcast IP address of 255.255.255.255 using UDP port 67.
DHCPOFFER	A DHCP server sends this message in response to a DHCPDISCOVER message using UDP port 68.
DHCPREQUEST	This broadcast message is a request from the client to the DHCP server for the IP addressing information and options that were received in the DHCP Offer message.
DHCPDECLINE	This message is sent from a client to a DHCP server to inform the server that an IP address is already in use on the network.
DHCPACK	A DHCP server sends this message to a client and includes IP configuration parameters.
DHCPNAK	A DHCP server sends this message to a client and informs the client that the DHCP server declines to provide the client with the requested IP configuration information.
DHCPRELEASE	A client sends this message to a DHCP server and informs the DHCP server that the client has released its DHCP lease, thus allowing the DHCP server to reassign the client IP address to another client.
DHCPINFORM	This message is sent from a client to a DHCP server and requests IP configuration parameters. Such a message might be sent from an access server requesting IP configuration information for a remote client attaching to the access server.

In addition to acting as a DHCP relay agent, a router might act as a DHCP client. Specifically, the interface of a router might obtain its IP address from a DHCP server. Figure 9-6 shows a router acting as a DHCP client, where the router's Fast Ethernet 0/1 interface obtains its IP address from a DHCP server. Example 9-4 provides the configuration for the router in the topology (that is, router R1). Notice the **dhcp** option used in the **ip address** command, instead of the usual IP address and subnet mask information.

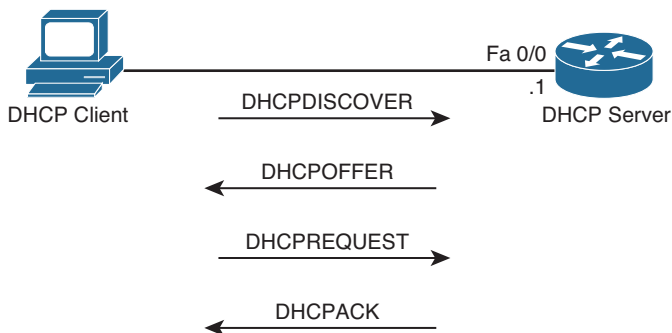
**Figure 9-6** *Router Acting as a DHCP Client*

**Key
Topic****Example 9-4** *DHCP Client Configuration*

```
R1#configure terminal
R1(config)#int fa 0/1
R1(config-if)#ip address dhcp
```

**Key
Topic**

A router and multilayer switch can also act as a DHCP server. Figure 9-7 shows a router acting as a DHCP server, and Example 9-5 shows the router configuration. The **ip dhcp excluded-address 10.8.8.1 10.8.8.10** command prevents DHCP from assigning those IP addresses to a client. Note that you do not have to include the IP address of the router interface in this exclusion because the router will never hand out its own interface IP address. The **ip dhcp pool POOL-A** command creates a DHCP pool named POOL-A. This pool can hand out IP addresses from the 10.8.8.0/24 network, with a default gateway of 10.8.8.1, a DNS server of 192.168.1.1, and a WINS server of 192.168.1.2.

**Figure 9-7** *Router Acting as a DHCP Server***Example 9-5** *DHCP Server Configuration*

```
R1#show run
...OUTPUT OMITTED...
ip dhcp excluded-address 10.8.8.1 10.8.8.10
!
ip dhcp pool POOL-A
    network 10.8.8.0 255.255.255.0
    default-router 10.8.8.1
    dns-server 192.168.1.1
    netbios-name-server 192.168.1.2
...OUTPUT OMITTED...
```

If your device is configured to receive an IP address from a DHCP server but the IP address of the client is an APIPA (Automatic Private IP Addressing) address (169.254.x.x) because of autoconfiguration, as shown in Example 9-6, you can conclude that the client was not able to obtain an IP address from the DHCP server. Do not immediately assume that DHCP is the problem. It is quite possible that you have a Layer 2 problem, such as VLANs, trunks, Spanning Tree Protocol (STP), or security, for example, that is preventing the client's DHCPDISCOVER message from reaching the DHCP server.

Example 9-6 *Verifying DHCP-Assigned IP Address on PC*

```

C:\>ipconfig /all
Windows IP Configuration

...output omitted...

Ethernet adapter PCI Lab:

    Connection-specific DNS Suffix  . : 
    Description . . . . . : AMD PCNET Family PCI Ethernet Adapter
    Physical Address. . . . . : 08-00-27-5D-06-D6
    Dhcp Enabled. . . . . : Yes
    Autoconfiguration Enabled . . . . : Yes
    Autoconfiguration IP Address. . . : 169.254.180.166
    Subnet Mask . . . . . : 255.255.0.0
    IP Address. . . . . : 2001:10::10
    IP Address. . . . . : fe80::a00:27ff:fe5d:6d6%4
    Default Gateway . . . . . : 

```

Potential DHCP Troubleshooting Issues

When troubleshooting what you suspect might be a DHCP issue, consider the following potential issues:



- **A router not forwarding broadcasts:** By default, a router does not forward broadcasts, including DHCPDISCOVER broadcast messages. Therefore, a router needs to be explicitly configured to act as a DHCP relay agent if the DHCP client and DHCP server are on different subnets.
- **DHCP pool out of IP addresses:** A DHCP pool contains a finite number of addresses. Once a DHCP pool becomes depleted, new DHCP requests are rejected.
- **Misconfiguration:** The configuration of a DHCP server might be incorrect. For example, the range of network addresses to be given out by a particular pool might be incorrect, or the exclusion of addresses statically assigned to routers or DNS servers might be incorrect.
- **Duplicate IP addresses:** A DHCP server might hand out an IP address to a client that is already statically assigned to another host on the network. These duplicate IP addresses can cause connectivity issues for both the DHCP client and the host that had been statically configured for the IP address.
- **Redundant services not communicating:** Some DHCP servers can coexist with other DHCP servers for redundancy. For this redundancy to function, these DHCP servers need to communicate with one another. If this interserver communication fails, the DHCP servers can hand out overlapping IP addresses to their clients.

- **The “pull” nature of DHCP:** When a DHCP client wants an IP address, it can request an IP address from a DHCP server. However, the DHCP server has no ability to initiate a change in the client IP address after the client obtains an IP address. In other words, the DHCP client pulls information from the DHCP server, but the DHCP server cannot push information to the DHCP client.
- **Interface not configured with IP address in DHCP pool:** A router or a multilayer switch that is acting as a DHCP server must have an interface with an IP address that is part of the pool/subnet that it is handing out IP addresses for. The router will only hand the addresses in the pool to clients reachable out that interface. This ensures that the router interface and the clients are in the same subnet. However, note that this is not the case if a relay agent is forwarding DHCP messages between the client and the router that is the DHCP server. In that case, the DHCP server does not have to have an IP address on an interface that is part of the pool it is handing out addresses for.

At this point in this section, you have reviewed basic DHCP operations and potential DHCP troubleshooting targets. When you begin your troubleshooting efforts, you might want to collect the following information to help you better isolate the underlying cause of the DHCP issue you are investigating:

- **The configuration of the DHCP server:** For example, confirm that the pools are correctly defined with appropriate network addresses, default gateways, and other relevant IP address information.
- **The configuration of the DHCP relay agent:** For example, ensure that the specified helper address is the correct unicast IP address and ensure that it is configured on the interface on which DHCPDISCOVER broadcasts will be received.
- **Determine the size of a DHCP pool:** Because a pool in a DHCP server accommodates only a limited number of IP addresses, determine how many IP addresses (if any) are still available from a given DHCP pool.
- **Verify IP address of router interface:** If the router is a DHCP server (and there is no relay agent configured to forward DHCP messages to it), verify that the router has the correct IP address assigned to the correct interface based on the pools it will be handing out addresses for. If the interface does not have an IP address that is part of a pool it is handing out addresses for, it will not hand out addresses to the clients out the interface it is receiving DHCPDISCOVER messages on.

DHCP Troubleshooting Commands



Example 9-7 provides sample output from the **show ip dhcp conflict** command. The output indicates a duplicate 172.16.1.3 IP address on the network, which the router discovered via a ping. You can clear the information displayed by issuing the **clear ip dhcp conflict *** command after you have resolved the duplicate address issue on the network.

Example 9-7 show ip dhcp conflict *Command Output*

```
R1#show ip dhcp conflict
IP address      Detection method  Detection time
172.16.1.3      Ping              Oct 15 2014 8:56 PM
```

Example 9-8 shows sample output from the **show ip dhcp binding** command. The output indicates that an IP address of 10.1.1.10 was assigned to a DHCP client. You can release this DHCP lease with the **clear ip dhcp binding *** command.

Example 9-8 show ip dhcp binding *Command Output*

```
R1#show ip dhcp binding
Bindings from all pools not associated with VRF:
IP address      Client-ID/          Lease expiration    Type
                Hardware address/
                User name
10.1.1.3        0100.50b6.0765.7a   Oct 17 2014 07:53 PM Automatic
10.1.1.10       0108.0027.5d06.d6   Oct 17 2014 07:53 PM Automatic
```

Example 9-9 shows sample output from the **debug ip dhcp server events** command. The output shows updates to the DHCP database.

Example 9-9 debug ip dhcp server events *Command Output*

```
R1#debug ip dhcp server events
DHCPD: Seeing if there is an internally specified pool class:
  DHCPD: htype 1 chaddr c001.0f1c.0000
  DHCPD: remote id 020a00000a01010101000000
  DHCPD: circuit id 00000000
DHCPD: Seeing if there is an internally specified pool class:
  DHCPD: htype 1 chaddr c001.0f1c.0000
  DHCPD: remote id 020a00000a01010101000000
  DHCPD: circuit id 00000000
DHCPD: no subnet configured for 192.168.1.238.
```

Example 9-10 shows sample output from the **debug ip dhcp server packet** command. The output shows a DHCPRELEASE message being received when a DHCP client with an IP address of 10.1.1.3 is shut down. You can also see the four-step process of a DHCP client obtaining an IP address of 10.1.1.4 with the following messages: DHCPDISCOVER, DHCPOFFER, DHCPREQUEST, and DHCPACK.

Example 9-10 debug ip dhcp server packet *Command Output*

```
R1#debug ip dhcp server packet
DHCPD: DHCPRELEASE message received from client
0063.6973.636f.2d63.3030.312e.3066.3163.2e30.3030.302d.4661.302f.30 (10.1.1.3).
DHCPD: DHCPRELEASE message received from client
0063.6973.636f.2d63.3030.312e.3066.3163.2e30.3030.302d.4661.302f.30 (10.1.1.3).
DHCPD: Finding a relay for client
```

```

0063.6973.636f.2d63.3030.312e.3066.3163.2e30.3030.302d.4661.302f.30 on interface
FastEthernet0/1.
DHCPD: DHCPDISCOVER received from client
0063.6973.636f.2d63.3030.312e.3066.3163.2e30.3030.302d.4661.302f.30 on interface
FastEthernet0/1.
DHCPD: Allocate an address without class information
(10.1.1.0)
DHCPD: Sending DHCPOFFER to client
0063.6973.636f.2d63.3030.312e.3066.3163.2e30.3030.302d.4661.302f.30 (10.1.1.4) .
DHCPD: broadcasting BOOTREPLY to client c001.0f1c.0000.
DHCPD: DHCPREQUEST received from client
0063.6973.636f.2d63.3030.312e.3066.3163.2e30.3030.302d.4661.302f.30.
DHCPD: No default domain to append - abort update
DHCPD: Sending DHCPACK to client
0063.6973.636f.2d63.3030.312e.3066.3163.2e30.3030.302d.4661.302f.30 (10.1.1.4) .
DHCPD: broadcasting BOOTREPLY to client c001.0f1c.0000.

```

Troubleshooting NAT

In IPv4 networks, Network Address Translation (NAT) is needed for multiple reasons. However, the most common reason is to translate a private IPv4 address to a public IPv4 address. This section explains how NAT operates and identifies NAT related issues and how to troubleshoot them.

Reviewing NAT

Public IP addresses are routable through the public Internet, whereas private IP addresses (as defined in RFC 1918) are not, and are intended for use within an organization. Because devices within an organization using private IP addresses need to communicate outside of their local networks (Internet/public network), NAT is needed to translate the private IP addresses into Internet-routable IP addresses (that is, public IP addresses). Table 9-3 identifies three types of NAT.



Table 9-3 *Types of NAT*

Type of NAT	Description (Based on Private to Public IPv4 Address Translations)
Static NAT	A one-to-one mapping of private internal IP addresses to public external IP addresses
Dynamic NAT	A dynamic mapping of private internal IP addresses to a pool of public external IP addresses
NAT overloading or PAT	Allows multiple private internal IP addresses to use a single public external IP address by keeping track of Layer 4 port numbers, which makes each session unique.

Consider Figure 9-8, which shows a basic NAT topology. In the topology, a client with a private IP address of 10.1.1.10 wants to communicate with a server on the public Internet at 192.0.2.1. Router R1 is configured for NAT. Router R1 takes packets coming from 10.1.1.10 that are destined for remote devices (such as the server) and changes the source IP address in the packet headers to 203.0.113.1, which is a publicly routable IP address. When the server at IP address 192.0.2.1 receives traffic from the client, the return traffic from the server is sent to a destination address of 203.0.113.1. When router R1 receives traffic from the outside network destined for 203.0.113.1, the router translates the destination IP address to 10.1.1.10 and forwards the traffic to the inside network, where the client receives the traffic.

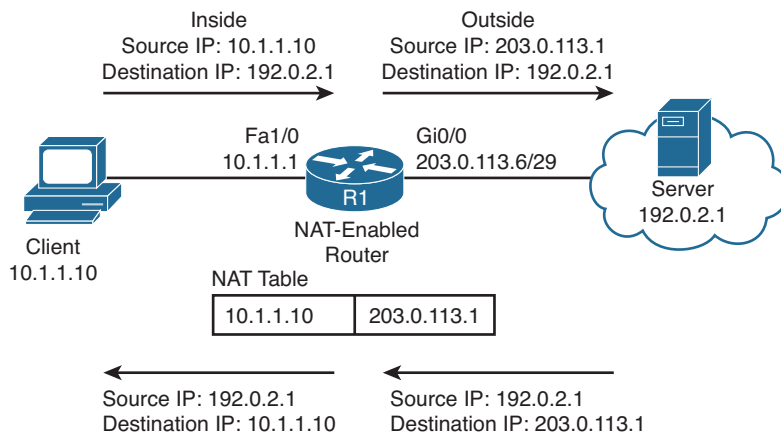


Figure 9-8 Basic NAT Topology Example

Note that NAT by itself does not scale well for an organization that has many privately addressed devices that need access to the public Internet all at the same time. As you can see from the example just discussed, you would need one public IP for every single private IP you want to translate. To overcome this issue, you use Port Address Translation (PAT), which can translate multiple private IP addresses to the same public IP address by keeping track of port numbers. PAT is simply a feature that NAT provides. Figure 9-9 displays how PAT can use port numbers and have multiple private IPs use the same public IP address. In this case, the public IP address is the same one that is used on the outside interface; however, you could also use a pool of public addresses. When the packet sourced from IP 10.1.1.10 port 65500 arrives at the PAT-enabled router, it translates it to the appropriate public IP and makes note of the port number as well as the IP address in the NAT table. Therefore, when traffic returns from the server destined to 203.0.113.1 port 65500, the router knows it is destined for 10.1.1.10 at port 65500. The client at 10.1.1.20 will also be translated to the same outside IP address, but a different port number will be used. This ensures that the router can differentiate between the different packets and who they are truly sourced from and destined to.

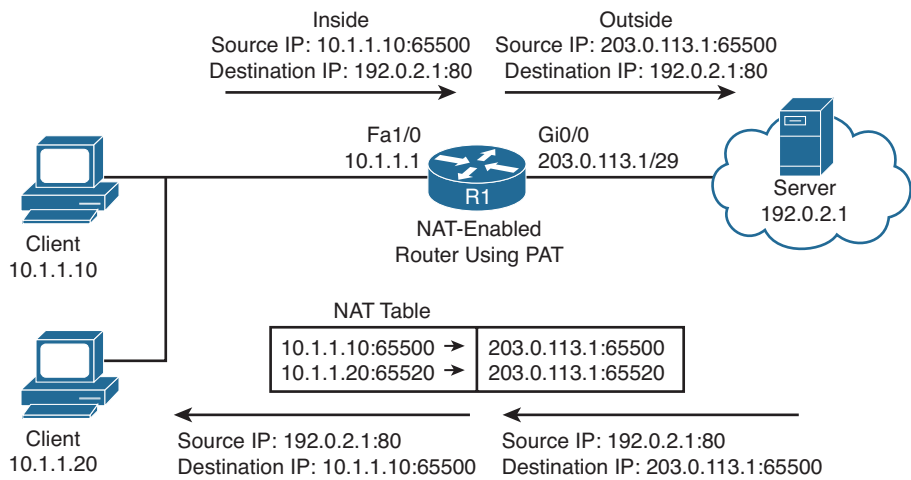


Figure 9-9 Basic PAT Topology Example

To effectively troubleshoot a NAT configuration, you should be familiar with the terminology describing the various IP addresses involved in a translation, as outlined in Table 9-4.



Table 9-4 Names of NAT IP Addresses

NAT IPs	Definition
Inside local	The IP address of a device inside the network; this address will be translated to the inside global address. (Example: PC inside the network)
Inside global	The IP address that the Inside local address is translated to. (Example: public IP address used on Internet)
Outside local	The IP address of a remote device as it appears to the devices inside the network. This may or may not be the actual address of the remote device if NAT translated it. Note that usually the outside global and outside local addresses are the same.
Outside global	The IP address of the device that the inside local address is trying to communicate with. This may be translated to the outside local address (but usually is not translated). (Example: web server)

Based on the definitions in Table 9-4, Table 9-5 categorizes the IP addresses previously shown in Figure 9-9.

Table 9-5 Classifying the NAT IP Addresses in Figure 9-9

NAT IPs	Address
Inside local	10.1.1.10
Inside global	203.0.113.1

NAT IPs	Address
Outside local	192.0.2.1
Outside global	192.0.2.1

NAT Troubleshooting Issues

Refer to Example 9-11, which displays a sample NAT/PAT configuration that uses a pool of public addresses. Notice how many different issues could arise based on a mistake in the configuration.



- **Interfaces not configured correctly:** You need to specify which interfaces will be the outside and inside interfaces. If not configured correctly, NAT will not translate addresses properly.
- **The pool may be misconfigured:** The pool must specify the correct public addresses that will be translated. This pool represents the inside global addresses.
- **The public addresses in the pool are not reachable:** For traffic to return from the destination, the public addresses need to be reachable (advertised) to the Internet.
- **The access list may not reference the correct inside devices:** The ACL identifies the inside local addresses that will be translated to inside global addresses. If the ACL is incorrect, addresses will not be translated correctly.
- **ACL and pool not mapped correctly:** The `ip nat inside source` command marries the ACL and the pool together. If this mapping is incorrect, NAT will not translate addresses correctly.
- **Overload keyword missing:** To enable PAT, the `overload` keyword must be included in the `ip nat inside source` command.

Example 9-11 *Dynamic NAT with PAT Sample Configuration*

```
R1#show run
...OUTPUT OMITTED...
interface FastEthernet1/0
 ip address 10.1.1.1 255.255.255.0
 ip nat inside
!
interface GigabitEthernet 0/0
 ip address 203.0.113.1 255.255.255.248
 ip nat outside
!
ip nat pool OUTSIDE_POOL 203.0.113.3 203.0.113.6 netmask 255.255.255.248
ip nat inside source list 1 pool OUTSIDE_POOL overload
!
access-list 1 permit 10.1.1.0 0.0.0.255
...OUTPUT OMITTED...
```

From a troubleshooting perspective, adding NAT into a network introduces additional troubleshooting issues. Consider the following situations in which NAT might cause an issue for end users:

- **Using NAT over a VPN:** Some VPN protocols check the checksum of a packet to verify its integrity. The checksum calculated for a packet before NAT differs from a checksum calculated for that same packet after NAT (because performing NAT on a packet changes IP address information). Therefore, a virtual private network (VPN) protocol (for example, IPsec) might reject such a packet because it appears to have been altered.
- **NAT hiding true IP address information:** Because NAT translates an inside IP address to an outside IP address, tracing a data flow from end to end for troubleshooting purposes can be challenging. You can start troubleshooting by using the **show ip nat translation** command to verify whether the translation does exist in the translation table.
- **Applications that are not NAT compatible:** When some applications initialize, they randomly determine what ports are going to be used for communication, which might be incompatible with how NAT handles incoming traffic. Some Voice over IP (VoIP) protocols face such an issue, as they select the User Datagram Protocol (UDP) port numbers to be used for their Real-time Transport Protocol (RTP) media streams. Also, when setting up communication with a remote device, an application might include IP address information in the payload of a packet. If the remote device attempted to return traffic to the IP address embedded in that payload, that IP address might be unreachable because of the NAT translation.
- **Delays experienced due to NAT's processing:** Because NAT manipulates Layer 3 information of packets, the packets are subject to a bit more delay than they would otherwise experience. This delay might become more evident on routers performing numerous NAT translations.

NAT Troubleshooting Commands



Example 9-12 provides sample output from the **show ip nat translations** command and how to clear all dynamic entries in this output with the **clear ip nat translation *** command. Initially, the **show ip nat translations** command shows three statically configured NAT translations and one dynamically learned translation (which is highlighted in the output). Then, after issuing the **clear ip nat translation *** command, the dynamically learned NAT entry is deleted from the IP NAT table, leaving the three statically configured NAT entries.

Example 9-12 show ip nat translations and clear ip nat translation * Command Output

```
Router#show ip nat translations
Pro Inside global      Inside local      Outside local      Outside global
--- 192.168.1.12        192.168.0.1      ---                ---
```

```

--- 192.168.1.13      192.168.0.2      ---      ---
tcp 192.168.1.27:23    192.168.0.27:23    192.168.1.50:1158  192.168.1.50:1158
--- 192.168.1.27      192.168.0.27      ---      ---
Router#clear ip nat translation *
Router#show ip nat translations
Pro Inside global      Inside local      Outside local      Outside global
--- 192.168.1.12        192.168.0.1        ---                ---
--- 192.168.1.13        192.168.0.2        ---                ---
--- 192.168.1.27        192.168.0.27       ---                ---

```

Example 9-13 provides sample output from the **show ip nat statistics** command. The output shows which interfaces are acting as the inside and outside interfaces, and it shows the current number of static and dynamic translations.

Example 9-13 show ip nat statistics Command Output

```

R1#show ip nat statistics
Total active translations: 4 (3 static, 1 dynamic; 1 extended)
Outside interfaces:
  FastEthernet0/0
Inside interfaces:
  FastEthernet0/1
Hits: 10 Misses: 0
CEF Translated packets: 5, CEF Punted packets: 0
Expired translations: 0
Dynamic mappings:
  Appl doors: 0
  Normal doors: 0
  Queued Packets: 0

```

Example 9-14 provides sample output from the **debug ip nat** command, which you should use with caution because it can crash a router. The output shows that when a source IP address of 192.168.1.50 is attempting to communicate with a destination IP address of 192.168.1.27, the router translates the destination IP address into 192.168.0.27. Also, when a source IP address of 192.168.1.11 is attempting to communicate with a destination IP address of 192.168.1.50, the router translates the source IP address of 192.168.1.11 into an IP address of 192.168.1.27.

Example 9-14 debug ip nat Command Output

```

R1#debug ip nat
IP NAT debugging is on
NAT*: s=192.168.1.50, d=192.168.1.27->192.168.0.27 [10202]
NAT: s=192.168.1.11->192.168.1.27, d=192.168.1.50 [210]
NAT*: s=192.168.1.50, d=192.168.1.27->192.168.0.27 [10370]
NAT: s=192.168.1.11->192.168.1.27, d=192.168.1.50 [211]
NAT*: s=192.168.1.50, d=192.168.1.27->192.168.0.27 [10540]
NAT: s=192.168.1.11->192.168.1.27, d=192.168.1.50 [214]

```


IPv4 Addressing and Addressing Technologies Trouble Tickets

This section presents various trouble tickets relating to the topics discussed earlier in the chapter. The purpose of these trouble tickets is to give a process that you can follow when troubleshooting in the real world or in an exam environment. All trouble tickets in this section are based on the topology depicted in Figure 9-10.

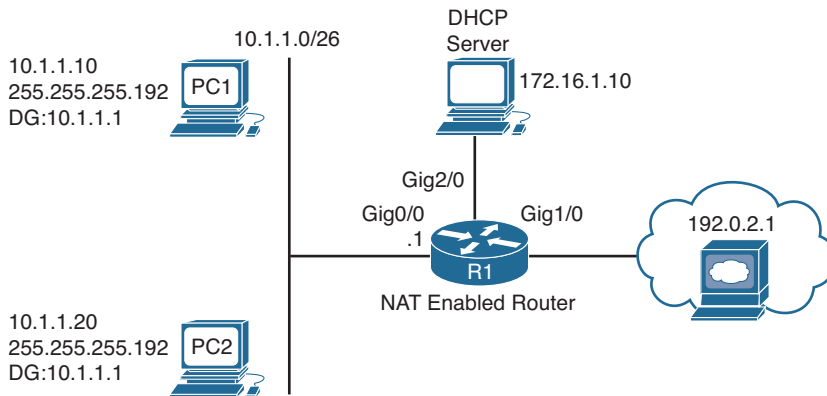


Figure 9-10 *IPv4 Addressing Trouble Tickets Topology*

Trouble Ticket 9-1

Problem: PC1 is not able to access resources on the web server 192.0.2.1.

You begin troubleshooting by verifying the issue with a ping from PC1 to 192.0.2.1. As shown in Example 9-15, the ping fails.

Example 9-15 *Failed Ping from PC1 to 192.0.2.1*

```
C:\PC1>ping 192.0.2.1
Pinging 192.0.2.1 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.0.2.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Next you ping the default gateway for PC1, which is R1, at 10.1.1.1. As shown in Example 9-16, the ping is successful.

Example 9-16 *Successful Ping from PC1 to Default Gateway*

```
C:\PC1>ping 10.1.1.1

Reply from 10.1.1.1: bytes=32 time 1ms TTL=128
Reply from 10.1.1.1: bytes=32 time 1ms TTL=128
Reply from 10.1.1.1: bytes=32 time 1ms TTL=128
Reply from 10.1.1.1: bytes=32 time 1ms TTL=128

Ping statistics for 10.1.1.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

You decide to see whether this is an isolated incident. You access PC2 and ping 192.0.2.1, which is successful, as shown in Example 9-17.

Example 9-17 *Successful Ping from PC2 to 192.0.2.1*

```
C:\PC2>ping 192.0.2.1

Reply from 192.0.2.1: bytes=32 time 1ms TTL=128
Reply from 192.0.2.1: bytes=32 time 1ms TTL=128
Reply from 192.0.2.1: bytes=32 time 1ms TTL=128
Reply from 192.0.2.1: bytes=32 time 1ms TTL=128

Ping statistics for 192.0.2.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

At this point, you have determined that Layer 2 and Layer 3 connectivity from PC1 and PC2 to the router is fine. You also confirmed that PC2 can reach Internet resources even though PC1 cannot. There are many reasons why this situation might exist. One of the big ones is an ACL on Gig0/0 or Gig1/0 that is denying PC1 from accessing resources on the Internet. It could also be a NAT issue that is preventing 10.1.1.10 from being translated. However, before we go down that path, review the basics. For example, what about the default gateway configured on PC1? If it is configured incorrectly, PC1 is sending packets that are destined to a remote subnet to the wrong default gateway. Reviewing the output of **ipconfig** on PC1, as shown in Example 9-18, indicates that the default gateway is configured as 10.1.1.100, which is not the IP address of R1's interface.

Example 9-18 *ipconfig Output on PC1*

```
C:\PC1>ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:
```

```

Connection-specific DNS Suffix  . :
IP Address. . . . . : 10.1.1.10
Subnet Mask . . . . . : 255.255.255.192
Default Gateway . . . . . : 10.1.1.100

```

After you change the default gateway on R1 to 10.1.1.1, the ping to 192.0.2.1 is successful, as shown in Example 9-19.

Example 9-19 Successful Ping from PC1 to 192.0.2.1

```

C:\PC1>ping 192.0.2.1

Reply from 192.0.2.1: bytes=32 time=1ms TTL=128
Reply from 192.0.2.1: bytes=32 time=1ms TTL=128
Reply from 192.0.2.1: bytes=32 time=1ms TTL=128
Reply from 192.0.2.1: bytes=32 time=1ms TTL=128

Ping statistics for 192.0.2.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

```

Trouble Ticket 9-2

Problem: PC1 is not able to access resources on the web server 192.0.2.1.

You begin troubleshooting by verifying the issue with a ping from PC1 to 192.0.2.1. As shown in Example 9-20, the ping fails.

Example 9-20 Failed Ping from PC1 to 192.0.2.1

```

C:\PC1>ping 192.0.2.1

Pinging 192.0.2.1 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.0.2.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

```

Next you ping the default gateway for PC1, which is R1, at 10.1.1.1. As shown in Example 9-21, it fails as well.

Example 9-21 *Failed Ping from PC1 to Default Gateway*

```

C:\PC1>ping 10.1.1.1
Pinging 10.1.1.1 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.1.1.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

```

Next you decide to see whether this is an isolated incident by pinging from PC2 to the IP address 192.0.2.1 and to the default gateway at 10.1.1.1. As shown in Example 9-22, both pings fail as well, indicating that it is not isolated.

Example 9-22 *Failed Ping from PC2 to 192.0.2.1 and Default Gateway*

```

C:\PC2>ping 192.0.2.1
Pinging 192.0.2.1 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.0.2.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\PC2>ping 10.1.1.1
Pinging 10.1.1.1 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.1.1.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

```

At this point, you have confirmed that there is no Layer 2 or Layer 3 connectivity from PC1 or PC2 to their default gateway. As you have seen in previous chapters, this can be caused by many different reasons. For example, VLANs, VACLs, trunks, VTP, STP, are all possible reasons why this issue is occurring. However, always remember to check the basics first: IP addressing on the client. On PC1, you issue the **ipconfig** command, and as shown in Example 9-23, PC1 has an APIPA (Automatic Private IP Addressing) address of 169.254.180.166/16 and no default gateway. This means that PC1 cannot contact a DHCP

server and is autoconfiguring an IP address. This still does not rule out a VLAN, trunk, VTP, STP, and so on. However, it is helping us narrow the focus.

Example 9-23 *ipconfig Output on PC1*

```
C:\PC1>ipconfig
Windows IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . : 
    IP Address. . . . . : 169.254.180.166
    Subnet Mask . . . . . : 255.255.0.0
    Default Gateway . . . . . :
```

We can clearly see in the trouble ticket topology in Figure 9-10 that the DHCP server is located out interface Gig2/0 on R1. It is in a different subnet than the PCs. Therefore, R1 is required to forward the DHCPDISCOVER messages from the PCs to the DHCP server at 172.16.1.10. To do this, it needs the **ip helper-address** command configured on Gig0/0. Let's start there so that we can eliminate this as the issue and focus elsewhere if need be. On R1, you issue the command **show run interface gig 0/0**, as shown in Example 9-24. The output indicates that the IP helper address is 172.16.1.100, which is not correct according to the network diagram.

Example 9-24 *Verifying the IP Helper Address on Gig0/0 of R1*

```
R1#show run interface gigabitEthernet 0/0
Building configuration...

Current configuration : 193 bytes
!
interface GigabitEthernet0/0
 ip address 10.1.1.1 255.255.255.192
 ip helper-address 172.16.1.100
 ip nat inside
end
```

After you fix the IP helper address with the **no ip helper-address 172.16.1.100** command and issue the **ip helper-address 172.16.1.10** command in interface configuration mode, PC1 successfully receives IP addressing information from the DHCP server, as shown in Example 9-25.

Example 9-25 *R1 with Correct IP Addressing After Fixing the IP Helper Address Command*

```
C:\PC1>ipconfig
Windows IP Configuration

Ethernet adapter Local Area Connection:
```

```

Connection-specific DNS Suffix  . :
IP Address. . . . . : 10.1.1.10
Subnet Mask . . . . . : 255.255.255.192
Default Gateway . . . . . : 10.1.1.1

```

After you verify the addressing information on R1, the ping to 192.0.2.1 is successful, as shown in Example 9-26.

Example 9-26 *Successful Ping from PC1 to 192.0.2.1*

```

C:\PC1>ping 192.0.2.1

Reply from 192.0.2.1: bytes=32 time=1ms TTL=128
Reply from 192.0.2.1: bytes=32 time=1ms TTL=128
Reply from 192.0.2.1: bytes=32 time=1ms TTL=128
Reply from 192.0.2.1: bytes=32 time=1ms TTL=128

Ping statistics for 192.0.2.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

```

Trouble Ticket 9-3

Problem: No PC in the 10.1.1.0 network can access resources on the Internet. You suspect NAT is the issue. As shown in Example 9-27, there are no translations occurring when you look at the output of **show ip nat translations**.

Example 9-27 *Viewing the NAT Translations on R1*

```

R1#show ip nat translations
R1#

```

In Example 9-28, you issue the **show ip nat statistics** command on R1 to verify which interfaces are participating in the NAT process. In this example, Gig0/0 and Gig1/0 are both configured to participate in NAT. However, take a moment to compare the output to Figure 9-10. Figure 9-10 shows that Gig0/0 is connected to the private network and Gig1/0 is connected to the public network. As such, Gig0/0 should be the inside interface, and Gig1/0 should be the outside interface. In the output of **show ip nat statistics**, this is not the case. As a result, you will need to modify the configuration so that Gig0/0 is the inside interface and Gig1/0 is the outside interface.

Example 9-28 *Viewing the NAT Statistics on R1*

```

R1#show ip nat statistics
Total active translations: 0 (0 static, 0 dynamic; 0 extended)
Outside interfaces:
    GigabitEthernet0/0

```

```

Inside interfaces:
  GigabitEthernet1/0
Hits: 0   Misses: 0
CEF Translated packets: 0, CEF Punted packets: 0
Expired translations: 0
Dynamic mappings:
-- Inside Source
[Id: 1] access-list 1 interface GigabitEthernet0/0 refcount 0
nat-limit statistics:
  max entry: max allowed 0, used 0, missed 0

```

In addition, if you look at the bottom of Example 9-28, you will notice that it states that IP addresses associated with access list 1 will be translated to the IP address associated with Gig0/0, which is supposed to be the inside interface according to Figure 9-10. Therefore, the **ip nat inside source** command needs to be modified as well. Example 9-29 shows the commands that are needed to modify the NAT configuration so that translations are successful.

Example 9-29 *Modifying the NAT Configuration on R1*

```

R1#config t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#interface gigabitEthernet 0/0
R1(config-if)#no ip nat outside
R1(config-if)#ip nat inside
R1(config-if)#interface gigabitEthernet 1/0
R1(config-if)#no ip nat inside
R1(config-if)#ip nat outside
R1(config-if)#exit
R1(config)#ip nat inside source list 1 interface gigabitEthernet 1/0 overload
R1(config)#end

```

After making the modifications, you ping from 10.1.1.10 to 192.0.2.1 and it fails. The output of **show ip nat translations** in Example 9-30 still shows no translations occurring.

Example 9-30 *Viewing the NAT Translations on R1 After the Configuration Changes*

```

R1#show ip nat translations
R1#

```

You review the output of **show ip nat statistics** again, as shown in Example 9-31, and confirm that everything looks fine. However, translations are still not occurring.

Example 9-31 *Viewing the NAT Statistics on R1 After the Configuration Changes*

```

R1#show ip nat statistics
Total active translations: 0 (0 static, 0 dynamic; 0 extended)
Outside interfaces:
  GigabitEthernet1/0

```

```

Inside interfaces:
  GigabitEthernet0/0
Hits: 0   Misses: 0
CEF Translated packets: 0, CEF Punted packets: 0
Expired translations: 0
Dynamic mappings:
-- Inside Source
[Id: 1] access-list 1 interface GigabitEthernet1/0 refcount 0
nat-limit statistics:
  max entry: max allowed 0, used 0, missed 0

```

One item that has not been checked yet is the access list. In this case, ACL 1 is being used to identify the private addresses that will be translated. Example 9-32 displays the output of **show access-list 1**. This ACL is permitting the IP addresses from 10.1.1.64 to 10.1.1.127. If you compare this range of addresses to Figure 9-10, it is incorrect. It should be 10.1.1.0 to 10.1.1.63 that is permitted.

Example 9-32 Viewing the NAT Statistics on R1

```

R1#show access-lists 1
Standard IP access list 1
  10 permit 10.1.1.64, wildcard bits 0.0.0.63

```

After removing the ACL with the **no access-list 1** command and creating a new one with the **access-list 1 permit 10.1.1.0 0.0.0.63** command, a ping from 10.1.1.10 to 192.0.2.1 is successful. In addition, as shown in Example 9-33, the output of **show ip nat translations** indicates that a translation has occurred.

Example 9-33 Successful Translations on R1

```

R1#show ip nat translations
Pro  Inside global      Inside local      Outside local      Outside global
icmp 203.0.113.1:1024    10.1.1.10:512     192.0.2.1:512     192.0.2.1:1024

```


Exam Preparation Tasks

As mentioned in the section “How to Use This Book” in the Introduction, you have a couple of choices for exam preparation: the exercises here; Chapter 22, “Final Preparation;” and the exam simulation questions on the CD-ROM.

Review All Key Topics

Review the most important topics in this chapter, noted with the Key Topic icon in the outer margin of the page. Table 9-6 lists a reference of these key topics and the page numbers on which each is found.



Table 9-6 *Key Topics for Chapter 9*

Key Topic Element	Description	Page Number
Paragraph	Examines the process that is used by a device to determine whether the packet will be sent to a local or remote device	338
Paragraph	Examines what occurs when IPv4 addressing is not correct	340
Example 9-1	Verifying IP addressing on PC with the <code>ipconfig</code> command	340
Paragraph	Explores how to determine the valid usable IPv4 addresses within a subnet	341
Step list	Examines the DHCPv4 DORA process	343
Example 9-3	DHCP relay agent configuration	344
Example 9-4	DHCP client configuration	346
Paragraph	Describes how a router can be configured as a DHCP server	346
List	Items to look out for while troubleshooting DHCP related issues	347
Section	DHCP troubleshooting commands	348
Table 9-3	Types of NAT	350
Table 9-4	Names of NAT IP addresses	352
List	Outlines the issues that you may have to troubleshoot with a NAT configuration	353
Section	NAT troubleshooting commands	354

Define Key Terms

Define the following key terms from this chapter and check your answers in the glossary:

DHCP, DORA, DHCPDISCOVER, DHCP OFFER, DHCP REQUEST, DHCP ACK, DHCP relay agent, APIPA, NAT, PAT/NAT overloading, static NAT, dynamic NAT, inside local, inside global, outside local, outside global

Command Reference to Check Your Memory

This section includes the most important verification and **show** commands covered in this chapter. It might not be necessary to memorize the complete syntax of every command, but you should be able to remember the basic keywords that are needed.

To test your memory of the commands, cover the right side of Table 9-7 with a piece of paper, read the description on the left side, and then see how much of the command you can remember.

The 300-135 TSHOOT exam focuses on practical, hands-on skills that are used by a networking professional. Therefore, you should be able to identify the commands needed to verify and troubleshoot the topics covered in this chapter.

Table 9-7 *Verification and show Commands*

Task	Command Syntax
Displays the IP address, subnet mask, and default gateway of a PC	ipconfig
Displays the IP address, subnet mask, and default gateway of a PC, in addition to DNS servers, domain name, MAC address, and whether autoconfiguration is enabled or not	ipconfig /all
Displays various IP related parameters for a router interface, including the IP address and subnet mask that have been assigned	show ip interface <i>interface_type interface_number</i>
Identifies any IP address conflicts a router configured as a DHCP server identifies, along with the method the router used to identify the conflicts (this is, via ping or gratuitous ARP)	show ip dhcp conflict
Displays IP addresses that an IOS DHCP server assigns, their corresponding MAC addresses, and lease expirations	show ip dhcp binding
Used to see all entries in a router's NAT translation table	show ip nat translations
Used to display NAT configuration and statistical information on a router, such as inside and outside interfaces, total translations, number of expired translations, address ACL, and address pool information	show ip nat statistics



This chapter covers the following topics:

- **Troubleshooting IPv6 Addressing:** This section explains how IPv6 devices determine whether traffic is destined locally or remotely. In addition, the section covers how MAC addresses are learned with Neighbor Solicitation and Neighbor Advertisement messages when using IPv6.
- **Troubleshooting IPv6 Address Assignment:** This section identifies the different methods that you can use to assign IPv6 addresses to clients. These methods include SLAAC, stateless DHCPv6, and stateful DHCPv6. You will also learn how to verify and troubleshoot IPv6 address assignment methods.
- **IPv6 Addressing Trouble Tickets:** This section provides trouble tickets that demonstrate how a structured troubleshooting process can be used to solve a reported problem.

Troubleshooting IPv6 Addressing and Addressing Technologies

Most organizations are still using IPv4; however, sooner or later they will have to switch to IPv6. When comparing IPv6 to IPv4, there is a whole lot more to IPv6 than it just being a larger address space. For example, because broadcasts have been removed from IPv6, multicast addresses are used in its place for addressing functions. Therefore, you need to be aware of these multicast addresses to successfully troubleshoot IPv6 addressing issues.

This chapter covers how an IPv6-enabled device determines whether the destination is local or remote. You will also learn how MAC addresses are determined for known IPv6 addresses, and you will explore the various options for address assignment and what to look for while troubleshooting related issues.

“Do I Know This Already?” Quiz

The “Do I Know This Already?” quiz allows you to assess whether you should read this entire chapter thoroughly or jump to the “Exam Preparation Tasks” section. If you are in doubt about your answers to these questions or your own assessment of your knowledge of the topics, read the entire chapter. Table 10-1 lists the major headings in this chapter and their corresponding “Do I Know This Already?” quiz questions. You can find the answers in Appendix A, “Answers to the ‘Do I Know This Already?’ Quizzes.”

Table 10-1 “Do I Know This Already?” Section-to-Question Mapping

Foundation Topics Section	Questions
Troubleshooting IPv6 Addressing	1–4
Troubleshooting IPv6 Addressing Assignment	5–10

Caution The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you do not know the answer to a question or are only partially sure of the answer, you should mark that question as wrong for purposes of the self-assessment. Giving yourself credit for an answer that you correctly guess skews your self-assessment results and might provide you with a false sense of security.

1. What protocol is used with IPv6 to determine the MAC address of a device in the same local-area network?
 - a. Address Resolution Protocol
 - b. Inverse Address Resolution Protocol
 - c. Neighbor Discovery Protocol
 - d. Neighbor Solicitation
2. What type of message is used to determine the MAC address of a known IPv6 address?
 - a. Router Solicitation
 - b. Router Advertisement
 - c. Neighbor Solicitation
 - d. Neighbor Advertisement
3. Which of the following are true when using EUI-64? (Choose two answers.)
 - a. The interface MAC address is used unmodified.
 - b. The interface MAC address is used with FFFE added to the middle.
 - c. The seventh bit from the left in the MAC address is flipped.
 - d. The seventh bit from the right in the MAC address is flipped.
4. What command is used on a Cisco IOS router to enable SLAAC on an interface?
 - a. `ipv6 address autoconfig`
 - b. `ipv6 address dhcp`
 - c. `ipv6 address prefix eui-64`
 - d. `ipv6 nd ra suppress`
5. What are requirements for stateless autoconfiguration to function? (Choose three answers.)
 - a. The prefix must be a /64.
 - b. The router must be sending and not suppressing RA messages.
 - c. The router must be enabled for IPv6 unicast routing.
 - d. The router must be sending RS messages.

6. Which command is used on a Cisco IOS router to verify the IPv6 addresses that have been deployed to clients?
 - a. `show ipv6 dhcp mappings`
 - b. `show ipv6 dhcp interface`
 - c. `show ipv6 dhcp binding`
 - d. `show ipv6 dhcp pool`
7. Which command is used to enable a router to inform clients that they need to get additional configuration information from a DHCPv6 server?
 - a. `ipv6 nd ra suppress`
 - b. `ipv6 dhcp relay destination`
 - c. `ipv6 address autoconfig`
 - d. `ipv6 nd other-config-flag`
8. Which DHCPv6 message type is sent from the client as it is searching for a DHCPv6 server?
 - a. ADVERTISE
 - b. REPLY
 - c. SOLICIT
 - d. REQUEST
9. What is needed when a DHCPv6 server resides in a different network than the clients it is providing IPv6 addresses to?
 - a. Address Resolution Protocol
 - b. Neighbor Discovery Protocol
 - c. Relay agent
 - d. Network Address Translation
10. What command enables you to configure a router interface as a DHCPv6 relay agent?
 - a. `ipv6 forwarder`
 - b. `ipv6 helper-address`
 - c. `ipv6 dhcp relay destination`
 - d. `ipv6 dhcp client`

Foundation Topics

Troubleshooting IPv6 Addressing

Just like your personal street address uniquely defines where you live, an IPv6 address uniquely defines where a device resides. Your street address is made of two parts, the street name and the number of your residence; and the combination of these will be unique. The same is true with IPv6 addresses. They are made up of two parts. The first 64 bits usually represent the subnet prefix (what network you belong to), and the last 64 bits usually represent the interface ID/host ID (who you are in the network).

This section covers IPv6 addressing and assignment so that you are armed with the knowledge needed for troubleshooting IPv6 addressing issues.

IPv6 Addressing Review

As with IPv4, it is important that devices are configured with the appropriate IPv6 address based on where they reside so that packets can be successfully routed to and from them. Refer to Figure 10-1, which depicts an IPv6 network. 2001:db8:a:a::/64 represents the first 64 bits of the IPv6 address, which is the subnet prefix. This is the IPv6 network the nodes reside in. Router R1 has an interface IPv6 address of 2001:db8:a:a::1 where the last 64 bits, which are ::1 in this case, represent the interface/host ID or who it is in the IPv6 network. PC1 is ::10 and PC2 is ::20. All the devices in 2001:db8:a:a::/64 are configured with a default gateway address of R1's Gig0/0 interface, which is 2001:db8:a:a::1.

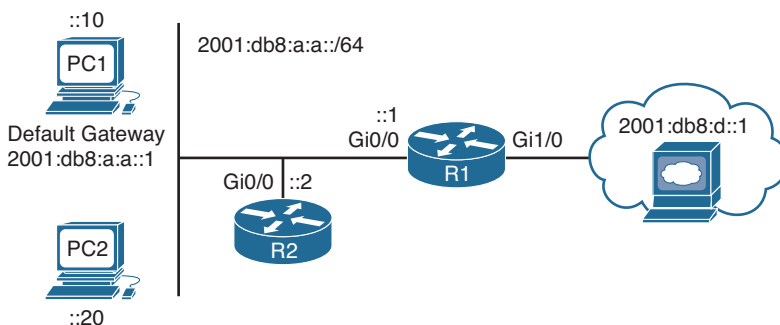


Figure 10-1 *IPv6 Addressing Example*

Neighbor Solicitation and Neighbor Advertisement

Just like IPv4, when a host wants to communicate with another host, it compares its subnet bits to the exact same bits in the destination IP address. If they match, both devices are in the same subnet; if they do not match, both devices are in different subnets. If both

devices are in the same subnet, they can communicate directly with each other, and if they are in different subnets, they will need to communicate through the default gateway.

For example, referring to Figure 10-1 again, when PC1 needs to communicate with the server at 2001:db8:d::1, it realizes that the web server is in a different network. Therefore, PC1 has to send the frame to the default gateway using the default gateway's MAC address. In IPv4, Address Resolution Protocol (ARP) was used to determine the MAC associated with an IPv4 address. ARP does not exist in IPv6, and neither do broadcasts. Instead, Neighbor Discovery Protocol (NDP) is used, which is based on multicasts.

Refer to Figure 10-2. In this case, PC1 sends a Neighbor Solicitation (NS) message sourced from its own IPv6 address 2001:db8:a:a::10 and MAC address 0800:275d:06d6. However, the destination IPv6 address and MAC address are solicited node multicast addresses because broadcasts do not exist. The IPv6 address solicited node multicast looks like this FF02:0:0:0:1:FFXX:XXXX. The X's are replaced with the last 24 bits (6 hex values) of the destination's IPv6 address. In this case, the IPv6 address of R1 (the destination) is 2001:db8:a:a::1. Therefore, the last 24 bits in hexadecimal would be 00:0001. So, the IPv6 destination solicited node multicast address would be FF02::1:FF00:1. The destination MAC solicited node multicast address looks like this 33:33:FF:XX:XX:XX. The last 24 bits (6 hex values) are the last 6 hex values of the IPv6 address (not MAC address). Therefore, the destination MAC address is 33:33:FF:00:00:01.

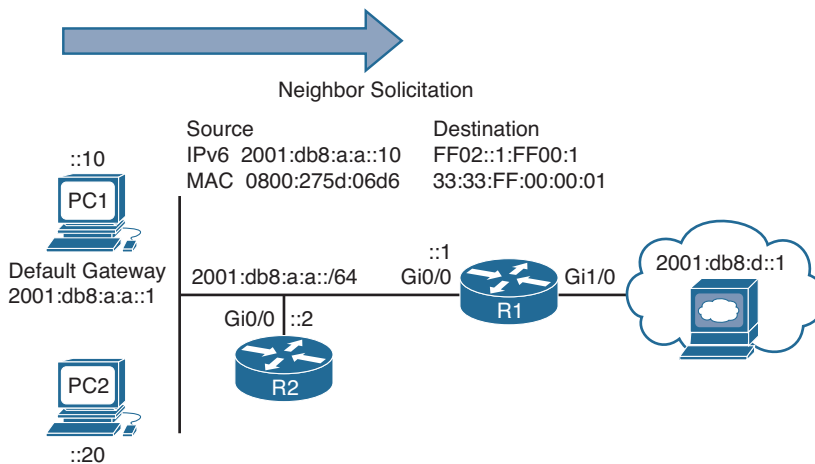


Figure 10-2 Neighbor Solicitation Example

Why does NDP go to this length just to send an NS message? Remember, there are no broadcasts with IPv6 at Layer 2 and Layer 3. Therefore, unicast communication or multicast communication is needed. Because we do not know the destination MAC, unicast is out of the question until we know it. So, multicast is used. However, you do not want to multicast to everyone, you only want to multicast to those devices that need to receive the multicast packet; therefore, those devices listening to the multicast group address. So, what is the group in this case? It is R1, the default gateway!

By default, all devices will create their own solicited node multicast group by appending the last 6 hex values of their IPv6 address to the IPv6 solicited node multicast address FF02:0:0:0:1:FF00::/104. As a result, when PC1 in our example sends the NS message, the destination is the solicited node multicast address that R1 is listening to. To verify the multicast groups that a router interface is listening to, you can use the **show ipv6 interface interface_type interface_number** command, as shown in Example 10-1. Notice that R1 is listening for packets destined to the multicast group FF02::1:FF00:1, as we had discussed with Figure 10-2. In addition, you can view the global unicast addresses assigned to the interface as well as the link-local address.

**Key
Topic**

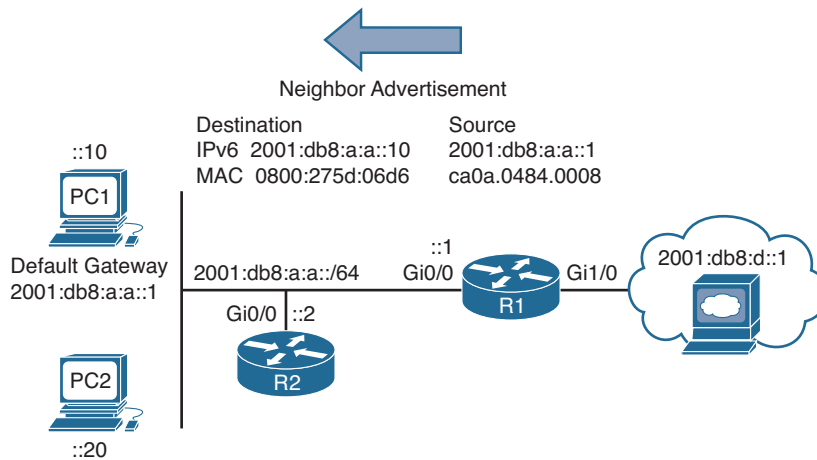
Example 10-1 *Verifying IPv6 Multicast Groups a Router Interface Is Listening To*

```
R1#show ipv6 interface gigabitEthernet 0/0
GigabitEthernet0/0 is up, line protocol is up
IPv6 is enabled, link-local address is FE80::C80A:4FF:FE84:8
No Virtual link-local address(es):
Global unicast address(es):
  2001:DB8:A:A::1, subnet is 2001:DB8:A:A::/64
Joined group address(es):
  FF02::1
  FF02::1:FF00:1
  FF02::1:FF84:8
MTU is 1500 bytes
ICMP error messages limited to one every 100 milliseconds
ICMP redirects are enabled
ICMP unreachable are sent
ND DAD is enabled, number of DAD attempts: 1
ND reachable time is 30000 milliseconds (using 30000)
ND NS retransmit interval is 1000 milliseconds
```

After R1 receives the NS message, it responds with a Neighbor Advertisement (NA), which will be a unicast packet. Refer to Figure 10-3, which shows R1 sending the NA to PC1 with a source IPv6 address 2001:db8:a:a:1 and MAC address ca0a.0484.0008.

Now PC1 can communicate with the server at 2001:db8:d::1 because it can send the frame to R1 and then R1 can route it.

You can verify the IPv6 address of a PC using the **ipconfig** command, as shown in Example 10-2. In this example, PC1 has a link-local address of fe80::a00:27ff:fe5d:6d6 and a global unicast address of 2001:db8:a:a::10, which was statically configured. Notice the %11 at the end of the link-local address in this case. This is the interface identification number. This is needed so that the system knows which interface to send the packets out of. The reason is because you can have multiple interfaces on the same device with the same link-local address assigned to it.

**Figure 10-3** Neighbor Advertisement Example**Example 10-2** Using `ipconfig` to Verify IPv6 Addressing

```
C:\PC1>ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . : 
    IPv6 Address. . . . . : 2001:db8:a:a::10
    Link-local IPv6 Address . . . . . : fe80::a00:27ff:fe5d:6d6%11
    IPv4 Address. . . . . : 10.1.1.10
    Subnet Mask . . . . . : 255.255.255.192
    Default Gateway . . . . . : 2001:db8:a:a::1
                                10.1.1.1
```

EUI-64

Recall that the IPv6 address consists of two parts: the subnet ID and the interface/host ID. The host ID is usually 64 bits long, and as a result is not something you want to be configuring manually in your organization. Although you can statically define the interface ID, the best approach is to allow your end devices to automatically assign their own interface ID for global unicast and link-local addresses based on the IEEE EUI-64 standard.



EUI-64 takes the clients MAC address, which is 48 bits, splits it in half, and adds the hex values FFFE in the middle. In addition, it takes the seventh bit from the left and flips it. So, if it is a 1, it becomes a 0, and if it is a 0, it becomes a 1. Look back at Example 10-2.

Notice the link-local address is fe80::a00:27ff:fe5d:6d6. The subnet ID is FE80::, and the interface ID is a00:27ff:fe5d:6d6. Let's fill in the missing leading 0s so that the address is 0a00:27ff:fe5d:06d6. This is an EUI-64 interface ID because it has FFFE in it. Let's see how it is derived.

Example 10-3 displays the output of **ipconfig /all** on PC1. Notice that the MAC address is 08-00-27-5D-06-D6. Split it in half and add FFFE in the middle so that you get 08-00-27-FF-FE-5D-06-D6. Now group the hex values into groups of four and replace the dashes (-) with colons, like this: 0800:27FF:FE5D:06D6. This looks very close to what is listed in the link-local address, but it is not exact. The interface ID in the link-local address starts with 0a and ours starts with 08. This is because the seventh bit is flipped, as discussed earlier. Let's flip it. 08 hex in binary is 00001000. The seventh bit from the left is a 0, so make it a 1. Now you have 00001010. Convert to hex and you get 0a. So, our interface ID is 0A00:27FF:FE5D:06D6.

Example 10-3 Using ipconfig /all to Verify IPv6 Addressing

```
C:\PC1>ipconfig /all

Windows IP Configuration

Host Name . . . . . : PC1_Win7
Primary Dns Suffix . . . . . :
Node Type . . . . . : Broadcast
IP Routing Enabled. . . . . : No
WINS Proxy Enabled. . . . . : No

Ethernet adapter Local Area Connection:

Connection-specific DNS Suffix . . :
Description . . . . . : Intel(R) PRO/1000 MT Desktop Adapter
Physical Address. . . . . : 08-00-27-5D-06-D6
DHCP Enabled. . . . . : No
Autoconfiguration Enabled . . . . : Yes
IPv6 Address. . . . . : 2001:db8:a:a::10 (Preferred)
Link-local IPv6 Address . . . . . : fe80::a00:27ff:fe5d:6d6%11 (Preferred)
IPv4 Address. . . . . : 10.1.1.10 (Preferred)
Subnet Mask . . . . . : 255.255.255.192
Default Gateway . . . . . : 2001:db8:a:a::1
                             10.1.1.1
DNS Servers . . . . . : fec0:0:0:ffff::1%1
                       fec0:0:0:ffff::2%1
                       fec0:0:0:ffff::3%1
NetBIOS over Tcpip. . . . . : Enabled
```

By default, routers will use EUI-64 when generating the interface portion of the link-local address of an interface. Modern Windows PCs will randomly generate the interface

portion by default for both the link-local address and the global unicast address when autoconfiguring their IPv6 addresses. However, this can be changed so that EUI-64 is used instead. When statically configuring an IPv6 address on a PC, the interface portion is manually assigned. However, on a router, if you want to use EUI-64 for a statically configured global unicast address, you can use the **eui-64** keyword at the end of the **ipv6 address** command, as shown in Example 10-4.

Example 10-4 Using EUI-64 on a Router Interface

```
R2#config t
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#interface gigabitEthernet 0/0
R2(config-if)#ipv6 address 2001:db8:a:a::/64 eui-64
```

You can verify the global unicast address and the EUI-64 interface ID assigned to it using the **show ipv6 interface** command, as shown in Example 10-5. In this case, R2's Gig0/0 interface has a global unicast address that obtained the interface ID from the EUI-64 standard.



Example 10-5 Verifying EUI-64 on a Router Interface

```
R2#show ipv6 interface gigabitEthernet 0/0
GigabitEthernet0/0 is up, line protocol is up
IPv6 is enabled, link-local address is FE80::C80E:15FF:FEF4:8
No Virtual link-local address(es):
Global unicast address(es):
  2001:DB8:A:A:C80E:15FF:FEF4:8, subnet is 2001:DB8:A:A::/64 [EUI]
Joined group address(es):
  FF02::1
  FF02::1:FFF4:8
MTU is 1500 bytes
...output omitted...
```

Troubleshooting IPv6 Address Assignment

Assigning any IP address (IPv4 or IPv6) manually is not a scalable option. With IPv4, you had Dynamic Host Configuration Protocol (DHCP) as your dynamic option. With IPv6, you have three dynamic options to choose from: stateless address autoconfiguration (or SLAAC for short), stateful DHCPv6, or stateless DHCPv6. Let's look at the issues that might arise for each and how we can troubleshoot these issues.

Stateless Address Autoconfiguration/SLAAC

Stateless address autoconfiguration (SLAAC) is designed so that devices are able to configure their own IPv6 address, prefix, and default gateway without a DHCPv6 server. Windows PCs are automatically enabled for SLAAC and will generate their own IPv6 addresses, as shown in Example 10-6, which displays the output of **ipconfig /all** on PC1.

Example 10-6 *Using ipconfig /all to Verify IPv6 SLAAC Is Enabled*

```

C:\PC1>ipconfig /all

Windows IP Configuration

    Host Name . . . . . : PC1_Win7
    Primary Dns Suffix . . . . . :
    Node Type . . . . . : Broadcast
    IP Routing Enabled. . . . . : No
    WINS Proxy Enabled. . . . . : No

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix . : SWITCH.local
    Description . . . . . : Intel(R) PRO/1000 MT Desktop Adapter
    Physical Address. . . . . : 08-00-27-5D-06-D6
    DHCP Enabled. . . . . : Yes
    Autoconfiguration Enabled . . . . : Yes
    IPv6 Address. . . . . : 2001:db8::a00:27ff:fe5d:6d6 (Preferred)
    Link-local IPv6 Address . . . . . : fe80::a00:27ff:fe5d:6d6%11 (Preferred)
    IPv4 Address. . . . . : 10.1.1.10 (Preferred)
    Subnet Mask . . . . . : 255.255.255.192
    ...output omitted...

```

On Cisco routers, if you want to take advantage of SLAAC, you need to enable it manually on an interface with the **ipv6 address autoconfig** command, as shown in Example 10-7.

**Example 10-7** *Enabling SLAAC on a Router Interface*

```

R2#config t
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#interface gigabitEthernet 0/0
R2(config-if)#ipv6 address autoconfig

```

When a Windows PC and router interface are enabled for SLAAC, they will send a Router Solicitation (RS) message to determine whether there are any routers connected to the local link. In turn, they wait for a router to send a Router Advertisement (RA) that identifies the prefix being used by the router (default gateway) connected to the same network they are on. They will then use that prefix information to generate their own IPv6 address in the same network as the router interface that generated the RA. The router will use EUI-64 for the interface portion, and the PC will randomly generate the interface portion unless it is configured to use EUI-64. In addition, the PC will use the IPv6 link-local address of the device that sent the RA as the default gateway address.

Key Topic

Refer to Figure 10-4, which displays the RA process. R1 sends an RA out its Gig0/0 interface. The source IPv6 address is the Gig0/0 link-local address, and the source MAC address is the MAC address of interface Gig0/0. The destination IPv6 address is the all-nodes link-local multicast IPv6 address of FF02::1. The destination MAC address is the all-nodes destination MAC address of 33:33:00:00:00:01 that is associated with the all-nodes link-local multicast IPv6 address FF02::1. By default, all IPv6-enabled interfaces listen for packets and frames destined for these two addresses.

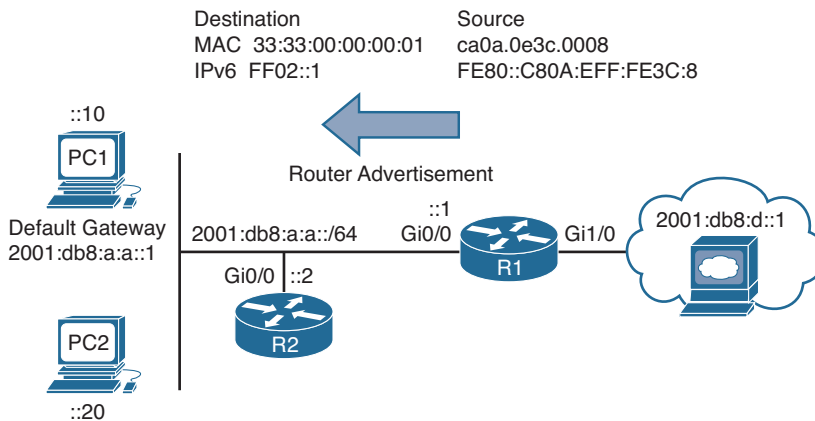


Figure 10-4 Router Advertisement Example

Once PC1 in Figure 10-4 receives the RA, it takes the prefix included in the RA, which is 2001:db8:a:a::/64, and in this case uses EUI-64 to create its IPv6 address. It also takes the link-local address from the source of the RA and uses it as the default gateway address, as shown in Example 10-8, which displays the output of **ipconfig** on PC1.

Example 10-8 Verifying IPv6 Addresses Generated by SLAAC on a PC

```
C:\PC1>ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . : 
    IPv6 Address. . . . . : 2001:db8:a:a:a00:27ff:fe5d:6d6
    Link-local IPv6 Address . . . . . : fe80::a00:27ff:fe5d:6d6%11
    IPv4 Address. . . . . : 10.1.1.10
    Subnet Mask . . . . . : 255.255.255.192
    Default Gateway . . . . . : fe80::c80a:eff:fe3c:8%11
                                10.1.1.1
```

Key Topic

To verify an IPv6 address generated by SLAAC on a router interface, use the **show ipv6 interface** command. As shown in Example 10-9, the global unicast address was generated using SLAAC. Also notice at the bottom of the example that the default router is listed

as the link-local address of R1. However, note that this will occur only if IPv6 unicast routing has not been enabled on the router and as a result the router is acting as an end device.

Example 10-9 Verifying IPv6 Addresses Generated by SLAAC on a Router Interface

```
R2#show ipv6 interface gig 0/0
GigabitEthernet0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::C80B:EFF:FE3C:8
  No Virtual link-local address(es):
  Stateless address autoconfig enabled
  Global unicast address(es):
    2001:DB8:A:A:C80B:EFF:FE3C:8, subnet is 2001:DB8:A:A::/64 [EUI/CAL/PRE]
    valid lifetime 2591816 preferred lifetime 604616
  Joined group address(es):
    FF02::1
    FF02::1:FF3C:8
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ICMP unreachables are sent
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds (using 30000)
  ND NS retransmit interval is 1000 milliseconds
  Default router is FE80::C80A:EFF:FE3C:8 on GigabitEthernet0/0
```

It is important to realize that RAs are generated by default on router interfaces only if the router interface is enabled for IPv6, IPv6 unicast routing is enabled, and RAs are not being suppressed on the interface. Therefore, if SLAAC is not working, check the following:



- Make sure that IPv6 unicast routing is enabled on the router that should be generating RAs by using the **show run | include ipv6 unicast-routing** command, as shown in Example 10-10.
- Make sure that the appropriate interface is enabled for IPv6 with the **show ipv6 interface** command, as shown in Example 10-11.
- Make sure that the router interface advertising RAs has a /64 prefix by using the **show ipv6 interface** command, as shown in Example 10-11. (SLAAC works only if the router is using a /64 prefix.)
- Make sure that RAs are not being suppressed on the interface by using the **show ipv6 interface** command, as shown in Example 10-12. In this example they are.

Example 10-10 Verifying IPv6 Unicast Routing Is Enabled on a Router

```
R1#show run | include ipv6 unicast-routing
ipv6 unicast-routing
```

**Key
Topic**

Example 10-11 *Verifying an Interface Is Enabled for IPv6*

```
R1#show ipv6 interface gigabitEthernet 0/0
GigabitEthernet0/0 is up, line protocol is up
IPv6 is enabled, link-local address is FE80::C80A:EFF:FE3C:8
No Virtual link-local address(es):
Global unicast address(es):
    2001:DB8:A:A::1, subnet is 2001:DB8:A:A::/64
Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF00:1
    FF02::1:FF3C:8
...output omitted...
```

**Key
Topic**

Example 10-12 *Verifying that RAs Are Not Suppressed*

```
R1#show ipv6 interface gigabitEthernet 0/0
GigabitEthernet0/0 is up, line protocol is up
IPv6 is enabled, link-local address is FE80::C80A:EFF:FE3C:8
No Virtual link-local address(es):
Global unicast address(es):
    2001:DB8:A:A::1, subnet is 2001:DB8:A:A::/64
Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF00:1
    FF02::1:FF3C:8
MTU is 1500 bytes
ICMP error messages limited to one every 100 milliseconds
ICMP redirects are enabled
ICMP unreachable are sent
ND DAD is enabled, number of DAD attempts: 1
ND reachable time is 30000 milliseconds (using 30000)
ND RAs are suppressed (all)
Hosts use stateless autoconfig for addresses.
```

In addition, if you have more than one router on the subnet generating RAs, which is normal when you have redundant default gateways, the clients will learn about multiple default gateways from the RAs, as shown in Example 10-13. The top default gateway is R2's link-local address, and the bottom default gateway is R1's link-local address. Now, this might seem like a benefit; however, it is a benefit only if both default gateways can reach the same networks. Refer to Figure 10-5. If PC1 uses R2 as the default gateway, the packets to the web server will be dropped because R2 does not have a way to route packets to the web server, as shown in the Example 10-14 ping, unless it redirects them back out the interface they arrived on, which is not a normal behavior. Therefore, if users are complaining that they cannot access resources, and they are connected to a network with

multiple routers generating RAs, check the default gateways learned by SLAAC and make sure that those default gateways can route to the intended resources.



Example 10-13 Verifying Default Gateways Configured on a PC

```
C:\PC1>#ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . : 
    IPv6 Address. . . . . : 2001:db8:a:a00:27ff:fe5d:6d6
    Link-local IPv6 Address . . . . . : fe80::a00:27ff:fe5d:6d6%11
    IPv4 Address. . . . . : 10.1.1.10
    Subnet Mask . . . . . : 255.255.255.192
    Default Gateway . . . . . : fe80::c80b:eff:fe3c:8%11
                              fe80::c80a:eff:fe3c:8%11
                              10.1.1.1
```

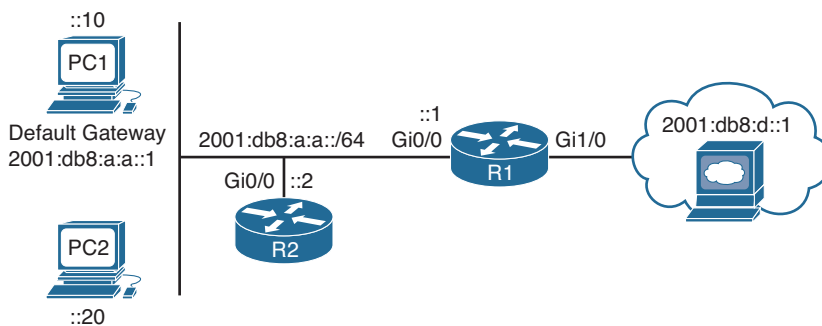


Figure 10-5 Redundant Default Gateways

Example 10-14 Failed Ping from PC1 to 2001:db8:a:a::1

```
C:\PC1>ping 2001:db8:d::1

Pinging 2001:db8:d::1 with 32 bytes of data:
Destination net unreachable.
Destination net unreachable.
Destination net unreachable.
Destination net unreachable.

Ping statistics for 2001:db8:d::1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Stateful DHCPv6

Although a device is able to determine its IPv6 address, prefix, and default gateway using SLAAC, there is not much else the devices can obtain. In a modern-day network, the devices may also need Network Time Protocol (NTP) server information, domain name information, DNS server information, and TFTP server information to name a few. To hand out the IPv6 addressing information along with all optional information, you need to use a DHCPv6 server. Both Cisco routers and multilayer switches can act as DHCP servers. Example 10-15 provides a sample DHCPv6 configuration on R1 and the **ipv6 dhcp server** interface command necessary to enable the interface to use the DHCP pool for handing out IPv6 addressing information. If you are troubleshooting an issue where clients are not receiving IPv6 addressing information or wrong IPv6 addressing information from a router or multilayer switch acting as a DHCPv6 server, check the interface and make sure that it has been associated with the correct pool.



Example 10-15 Sample DHCPv6 Configuration on R1

```
R1#show run | section dhcp
ipv6 dhcp pool DHCPV6POOL
  address prefix 2001:DB8:A:A::/64
  dns-server 2001:DB8:B:B::1
  domain-name TSHOOT.com

R1#show run interface gigabitEthernet 0/0
Building configuration...

Current configuration : 173 bytes
!
interface GigabitEthernet0/0
  no ip address
  ipv6 address 2001:DB8:A:A::1/64
  ipv6 dhcp server DHCPV6POOL
end
```

In Example 10-16, you can see samples of the **show ipv6 dhcp binding** command, which displays the IPv6 addresses that are being used by clients, the **show ipv6 dhcp interface** command, which displays the IPv6 addresses that are being used by clients, and the **show ipv6 dhcp pool** command, which displays the configured pools.



Example 10-16 Verifying DHCPv6 Information on R1

```
R1#show ipv6 dhcp binding
Client: FE80::A00:27FF:FE5D:6D6
  DUID: 000100011B101C740800275D06D6
  Username : unassigned
  VRF : default
  IA NA: IA ID 0x0E080027, T1 43200, T2 69120
  Address: 2001:DB8:A:A:D519:19AB:E903:F802
```

```

        preferred lifetime 86400, valid lifetime 172800
        expires at May 25 2014 08:37 PM (172584 seconds)

R1#show ipv6 dhcp interface
GigabitEthernet0/0 is in server mode
    Using pool: DHCPV6POOL
    Preference value: 0
    Hint from client: ignored
    Rapid-Commit: disabled

R1#show ipv6 dhcp pool
DHCPv6 pool: DHCPV6POOL
    Address allocation prefix: 2001:DB8:A:A::/64 valid 172800 preferred 86400 (1 in
    use, 0 conflicts)
    DNS server: 2001:DB8:B:B::1
    Domain name: TSHOOT.com
    Active clients: 0

```

Stateless DHCPv6

Stateless DHCPv6 is a combination of SLAAC and DHCPv6. In this case, a router's RA is used by the clients to automatically determine their IPv6 address, prefix, and default gateway. Included in the RA is a flag that tells the client to get other nonaddressing information from a DHCPv6 server, such as the address of a DNS server or a TFTP server. To accomplish this you need to ensure that the **ipv6 nd other-config-flag** interface configuration command is enabled. This ensures that the RA informs the client that it must contact a DHCPv6 server for other information. In Example 10-17, you can see this command configured under interface Gigabit Ethernet 0/0. Also, in Example 10-17, you can see the output of **show ipv6 interface gigabitEthernet 0/0**, which states that hosts will obtain IPv6 addressing from *stateless autoconfig* and other information from a *DHCP server*.



Example 10-17 Verifying Stateless DHCPv6

```

R1#show run int gig 0/0
Building configuration...

Current configuration : 171 bytes
!
interface GigabitEthernet0/0
    no ip address
    media-type gbic
    speed 1000
    duplex full
    negotiation auto
    ipv6 address 2001:DB8:A:A::1/64

```

```

ipv6 nd other-config-flag
end

R1#show ipv6 interface gigabitEthernet 0/0
GigabitEthernet0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::C80A:EFF:FE3C:8
  No Virtual link-local address(es):
  Global unicast address(es):
    2001:DB8:A:A::1, subnet is 2001:DB8:A:A::/64
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF00:1
    FF02::1:FF3C:8
  ...output omitted...
  ND advertised default router preference is Medium
  Hosts use stateless autoconfig for addresses.
  Hosts use DHCP to obtain other configuration.

```

Example 10-18 shows the `ipconfig /all` output on PC1 after it has used stateless autoconfig for IPv6 addressing and then contacted a DHCPv6 server for DNS and domain name information.

Example 10-18 Verifying IPv6 Configuration on PC1

```

C:\PC1>ipconfig /all

Windows IP Configuration

Host Name . . . . . : PC1_Win7
Primary Dns Suffix . . . . . :
Node Type . . . . . : Broadcast
IP Routing Enabled. . . . . : No
WINS Proxy Enabled. . . . . : No
DNS Suffix Search List. . . . . : TSHOOT.com

Ethernet adapter Local Area Connection:

Connection-specific DNS Suffix . : TSHOOT.com
Description . . . . . : Intel(R) PRO/1000 MT Desktop Adapter
Physical Address. . . . . : 08-00-27-5D-06-D6
DHCP Enabled. . . . . : No
Autoconfiguration Enabled . . . . : Yes
IPv6 Address. . . . . : 2001:db8:a:a:a00:27ff:fe5d:6d6 (Preferred)

Link-local IPv6 Address . . . . . : fe80::a00:27ff:fe5d:6d6%11 (Preferred)
IPv4 Address. . . . . : 10.1.1.10 (Preferred)

```

```

Subnet Mask . . . . . : 255.255.255.192
Default Gateway . . . . . : fe80::c80a:eff:fe3c:8%11
                               10.1.1.1
DHCPv6 IAID . . . . . : 235405351
DHCPv6 Client DUID. . . . . : 00-01-00-01-1B-10-1C-74-08-00-27-5D-06-D6

DNS Servers . . . . . : 2001:db8:b:b::10
NetBIOS over Tcpi. . . . . : Enabled
Connection-specific DNS Suffix Search List : TSHOOT.com

```

DHCPv6 Operation

DHCPv6 has a four-way negotiation process, like IPv4. However, DHCPv6 uses the following messages:



- Step 1. SOLICIT:** A client sends this message to locate DHCPv6 servers using the multicast address FF02::1:2, which is the all DHCPv6 servers multicast address.
- Step 2. ADVERTISE:** Servers respond to SOLICIT messages with a unicast ADVERTISE message offering addressing information to the client.
- Step 3. REQUEST:** The client sends this message to the server confirming the addresses provided and any other parameters.
- Step 4. REPLY:** The server finalizes the process with this message.

As a reference, Table 10-2 provides a comprehensive listing of DHCPv6 message types you might encounter while troubleshooting a DHCPv6 issue.

Table 10-2 *DHCP Message Types*

DHCP Message	Description
SOLICIT	A client sends this message in an attempt to locate a DHCPv6 server.
ADVERTISE	A DHCPv6 server sends this message in response to a SOLICIT, indicating it is available.
REQUEST	This message is a request for IP configuration parameters sent from a client to a specific DHCPv6 server.
CONFIRM	Sent from client to any server to determine whether the address it was assigned is still appropriate.
RENEW	Sent from client to server that assigned the address, to extend the lifetime of the addresses assigned.
REBIND	When there is no response to a RENEW, a REBIND is sent from client to any server to extend the lifetime on the address assigned.

DHCP Message	Description
REPLY	Sent from server to client containing assigned address and configuration parameters in response to a SOLICIT, REQUEST, RENEW, or REBIND message received from a client.
RELEASE	Sent from client to server to inform the server that the assigned address is no longer needed.
DECLINE	Sent from client to server to inform the server that the assigned address is already in use.
RECONFIGURE	Sent from server to client when the server has new or updated information.
INFORMATION-REQUEST	Sent from client to server when the client only needs additional configuration information without any IP address assignment.
RELAY-FORW	Used by relay agent to forward messages to DHCP server.
RELAY-REPL	Used by DHCP server to send messages back to the relay agent.

DHCPv6 Relay Agent

All the DHCPv6 examples so far have included the DHCP server within the same local network. However, in most networks, the DHCP server will be located in a different network, which creates an issue. If you review the multicast address of the SOLICIT message, you will notice it is a link-local scope multicast address. It starts with FF02. Therefore, the multicast will not leave the local network, and the client will not be able to reach the DHCPv6 server.

To relay the DHCPv6 messages to a DHCPv6 server in another network, the local router interface in the network the client belongs needs to be configured as a relay agent with the **ipv6 dhcp relay destination** interface configuration command. Example 10-19 shows interface Gigabit Ethernet 0/0 configured with the command **ipv6 dhcp relay destination 2001:db8:a:b::7**, which will be used to forward SOLICIT messages to a DHCPv6 server at the address listed.



Example 10-19 Configuring R1 as a DHCPv6 Relay Agent

```
R1#config t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#interface gigabitethernet0/0
R1(config-if)#ipv6 dhcp relay destination 2001:db8:a:b::7
```

IPv6 Addressing Trouble Tickets

This section presents various trouble tickets relating to the topics discussed earlier in the chapter. The purpose of these trouble tickets is to give a process that you can follow when troubleshooting in the real world or in an exam environment. All trouble tickets in this section are based on the topology depicted in Figure 10-6.

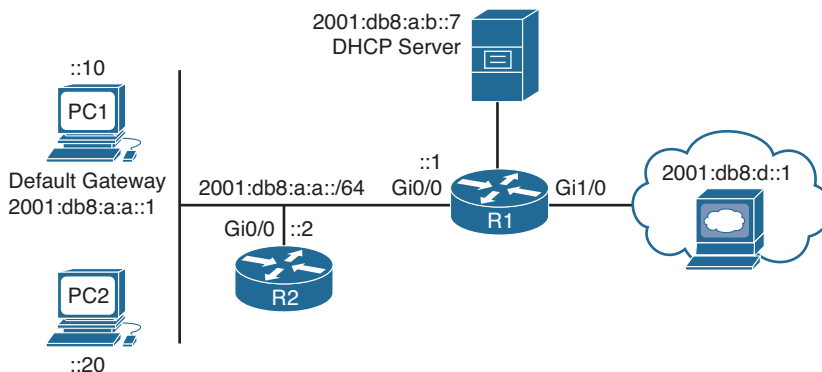


Figure 10-6 *IPv6 Addressing Trouble Tickets Topology*

Trouble Ticket 10-1

Problem: PC1 is not able to access resources on the web server at 2001:db8:d::1.

Your network uses stateless autoconfiguration for IPv6 addressing and DHCPv6 for additional options such as a domain name, TFTP server addresses, and DNS server addresses.

You begin troubleshooting by verifying the issue with a ping from PC1 to 2001:db8:d::1. As shown in Example 10-20, the ping fails.

Example 10-20 *Failed Ping from PC1 to Web Server at 2001:db8:d::1*

```
C:\PC1>ping 2001:db8:d::1

Pinging 2001:db8:d::1 with 32 bytes of data:
PING: transmit failed. General failure.
PING: transmit failed. General failure.
PING: transmit failed. General failure.
PING: transmit failed. General failure.

Ping statistics for 2001:db8:d::1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

You ping the default gateway at 2001:db8:a:a::1, but the ping fails, as shown in Example 10-21.

Example 10-21 *Failed Ping from PC1 to Default Gateway at 2001:db8:a:a::1*

```
C:\PC1>ping 2001:db8:a:a::1

Pinging 2001:db8:a:a::1 with 32 bytes of data:
PING: transmit failed. General failure.
PING: transmit failed. General failure.
PING: transmit failed. General failure.
PING: transmit failed. General failure.

Ping statistics for 2001:db8:a:a::1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Next you verify the IPv6 addresses on PC1 using the **ipconfig** command. Example 10-22 indicates that PC1 is not generating its own global unicast address using stateless auto-configuration or identifying a default gateway on the network.

Example 10-22 *Verifying IPv6 Addressing on PC1*

```
C:\PC1>ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . : TSHOOT.com
    Link-local IPv6 Address . . . . . : fe80::a00:27ff:fe5d:6d6%11
    IPv4 Address. . . . . : 10.1.1.10
    Subnet Mask . . . . . : 255.255.255.192
    Default Gateway . . . . . : 10.1.1.1
```

Your phone rings, and the user at PC2 is indicating that he cannot access any of the IPv6-enabled resources. You access PC2 and issue the **ipconfig** command, as shown in Example 10-23, and notice that it is not generating an IPv6 address either or identifying a default gateway.

Example 10-23 *Verifying IPv6 Addressing on PC2*

```
C:\PC2>ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . : TSHOOT.com
    Link-local IPv6 Address . . . . . : fe80::a00:27ff:fe5d:ce47%9
    IPv4 Address. . . . . : 10.1.1.20
    Subnet Mask . . . . . : 255.255.255.192
    Default Gateway . . . . . : 10.1.1.1
```


Recall that SLAAC relies on RAs. Therefore, R1's Gig0/0 interface needs to be sending RAs on the link for PC1 and PC2 to generate their own IPv6 addresses using SLAAC. You issue the command **show ipv6 interface gigabitethernet0/0** on R1, as shown in Example 10-24. The output indicates that hosts will use SLAAC for addresses, and DHCP will be used for other configuration values. However, it also indicates that RAs are suppressed. Therefore, PC1 and PC2 will not be receiving RAs that provide the prefix information necessary to perform autoconfiguration.

Example 10-24 *Verifying Whether RAs Are Suppressed on R1*

```
R1#show ipv6 interface gigabitEthernet 0/0
GigabitEthernet0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::C80A:EFF:FE3C:8
  No Virtual link-local address(es):
  Global unicast address(es):
    2001:DB8:A:A::1, subnet is 2001:DB8:A:A::/64
  Joined group address(es):
    FE02::1
    FE02::2
    FE02::1:2
    FE02::1:FF00:1
    FE02::1:FF3C:8
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ICMP unreachables are sent
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds (using 30000)
  ND RAs are suppressed (all)
  Hosts use stateless autoconfig for addresses.
  Hosts use DHCP to obtain other configuration.
```

You issue the command **show run interface gigabitethernet0/0** to verify the configuration commands on the interface. As shown in Example 10-25, the interface is configured with the command **ipv6 nd ra suppress all**, which stops R1 from sending RAs.

Example 10-25 *Verifying Interface Configuration on R1*

```
R1#show run interface gigabitEthernet 0/0
Building configuration...

Current configuration : 241 bytes
!
interface GigabitEthernet0/0
  no ip address
  ipv6 address 2001:DB8:A:A::1/64
  ipv6 nd other-config-flag
  ipv6 nd ra suppress all
```

```
ipv6 dhcp relay destination 2001:DB8:A:B::7
end
```

After you remove this command with the **no ipv6 nd ra suppress all** command, PC1 successfully generates a global IPv6 address and identifies an IPv6 default gateway, as shown in Example 10-26.

Example 10-26 Verifying IPv6 Addressing on PC1

```
C:\PC1>ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . : TSHOOT.com
    IPv6 Address. . . . . : 2001:db8:a:a00:27ff:fe5d:6d6
    Link-local IPv6 Address . . . . . : fe80::a00:27ff:fe5d:6d6%11
    IPv4 Address. . . . . : 10.1.1.10
    Subnet Mask . . . . . : 255.255.255.192
    Default Gateway . . . . . : fe80::c80a:eff:fe3c:8%11
                                10.1.1.1
```

You confirm that IPv6 resources are accessible by pinging 2001:db8:d::1 in Example 10-27, and it is successful. You then call the user at PC2 and confirm that he can access the resources as well. He indicates that he is.

Example 10-27 Successful Ping from PC1 to Web Server at 2001:db8:d::1

```
C:\PC1>ping 2001:db8:d::1

Pinging 2001:db8:d::1 with 32 bytes of data:
Reply from 2001:db8:d::1: time=37ms
Reply from 2001:db8:d::1: time=35ms
Reply from 2001:db8:d::1: time=38ms
Reply from 2001:db8:d::1: time=38ms

Ping statistics for 2001:db8:d::1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 35ms, Maximum = 38ms, Average = 36ms
```

Trouble Ticket 10-2

Problem: PC1 is not able to access resources on the web server at 2001:db8:d::1.

Your network uses stateless autoconfiguration for IPv6 addressing and DHCPv6 for additional options such as a domain name, TFTP server addresses, and DNS server addresses.

You begin troubleshooting by verifying the issue with a ping from PC1 to 2001:db8:d::1. As shown in Example 10-28, the ping fails.

Example 10-28 *Failed Ping from PC1 to Web Server at 2001:db8:d::1*

```
C:\PC1>ping 2001:db8:d::1

Pinging 2001:db8:d::1 with 32 bytes of data:
PING: transmit failed. General failure.
PING: transmit failed. General failure.
PING: transmit failed. General failure.
PING: transmit failed. General failure.

Ping statistics for 2001:db8:d::1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

You ping the default gateway at 2001:db8:a:a::1, but the ping fails, as shown in Example 10-29.

Example 10-29 *Failed Ping from PC1 to Default Gateway at 2001:db8:a:a::1*

```
C:\PC1>ping 2001:db8:a:a::1

Pinging 2001:db8:a:a::1 with 32 bytes of data:
PING: transmit failed. General failure.
PING: transmit failed. General failure.
PING: transmit failed. General failure.
PING: transmit failed. General failure.

Ping statistics for 2001:db8:a:a::1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Next you verify the IPv6 addresses on PC1 using the **ipconfig** command. Example 10-30 indicates that PC1 is not generating its own global unicast address using stateless auto-configuration; however, it is identifying a default gateway on the network at the link-local address fe80::c80a:eff:fe3c:8.

Example 10-30 *Verifying IPv6 Addressing on PC1*

```
C:\PC1>ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . : TSHOOT.com
    Link-local IPv6 Address . . . . . : fe80::a00:27ff:fe5d:6d6%11
    IPv4 Address. . . . . : 10.1.1.10
```

```
Subnet Mask . . . . . : 255.255.255.192
Default Gateway . . . . . : fe80::c80a:eff:fe3c:8%11
                             10.1.1.1
```

Your phone rings, and the user at PC2 is indicating that she cannot access any of the IPv6-enabled resources. You access PC2 and issue the **ipconfig** command, as shown in Example 10-31, and notice that it is experiencing the same issues as PC1.

Example 10-31 Verifying IPv6 Addressing on PC2

```
C:\PC2>ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . : TSHOOT.com
    Link-local IPv6 Address . . . . . : fe80::a00:27ff:fe5d:ce47%9
    IPv4 Address. . . . . : 10.1.1.10
    Subnet Mask . . . . . : 255.255.255.192
    Default Gateway . . . . . : fe80::c80a:eff:fe3c:8%11
                                10.1.1.1
```

Recall that SLAAC relies on RAs. Therefore, R1's Gig0/0 interface needs to be sending RAs on the link for PC1 and PC2 to generate their own IPv6 address using SLAAC. You issue the command **show ipv6 interface gigabitEthernet0/0** on R1, as shown in Example 10-32. The output indicates that hosts will use SLAAC for addresses, and DHCP will be used for other configuration values. Also, there is no indication that RAs are being suppressed. This is also confirmed by the fact that PC1 and PC2 are identifying a default gateway. However, is it the right one? According to Examples 10-30 and 10-31, the default gateway is fe80::c80a:eff:fe3c:8. Based on Example 10-32, this is correct. Review Example 10-32 further; can you see the issue?

Example 10-32 Verifying Whether RAs Are Suppressed on R1

```
R1#show ipv6 interface gigabitEthernet 0/0
GigabitEthernet0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::C80A:EFF:FE3C:8
  No Virtual link-local address(es):
  Global unicast address(es):
    2001:DB8:A:A::1, subnet is 2001:DB8:A::/60
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:2
    FF02::1:FF00:1
    FF02::1:FF3C:8
  MTU is 1500 bytes
```

```

ICMP error messages limited to one every 100 milliseconds
ICMP redirects are enabled
ICMP unreachable are sent
ND DAD is enabled, number of DAD attempts: 1
ND reachable time is 30000 milliseconds (using 30000)
ND advertised reachable time is 0 (unspecified)
ND advertised retransmit interval is 0 (unspecified)
ND router advertisements are sent every 200 seconds
ND router advertisements live for 1800 seconds
ND advertised default router preference is Medium
Hosts use stateless autoconfig for addresses.
Hosts use DHCP to obtain other configuration.

```

If you have not spotted it, look at the global prefix assigned to interface Gig0/0: 2001:db8:a::/60. SLAAC works only if the prefix is /64.

You issue the command **show run interface gigabitEthernet0/0** to verify the configuration commands on the interface. As shown in Example 10-33, the interface is configured with the command **ipv6 address 2001:db8:a:a::1/60**. RAs are still generated, but SLAAC will not work unless the prefix is a /64.

Example 10-33 Verifying Interface Configuration on R1

```

R1#show run interface gigabitEthernet 0/0
Building configuration...

Current configuration : 216 bytes
!
interface GigabitEthernet0/0
  ipv6 address 2001:DB8:A:A::1/60
  ipv6 nd other-config-flag
  ipv6 dhcp relay destination 2001:DB8:A:B::7
end

```

After you remove this command with the **no ipv6 address 2001:db8:a:a::1/60** command, and issue the command **ipv6 address 2001:db8:a:a::1/64**, PC1 successfully generates a global IPv6 unicast address, as shown in Example 10-34.

Example 10-34 Verifying IPv6 Addressing on PC1

```

C:\PC1>ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . : TSHOOT.com
    IPv6 Address. . . . . : 2001:db8:a:a:a00:27ff:fe5d:6d6

```

```

Link-local IPv6 Address . . . . . : fe80::a00:27ff:fe5d:6d6%11
IPv4 Address. . . . . : 10.1.1.10
Subnet Mask . . . . . : 255.255.255.192
Default Gateway . . . . . : fe80::c80a:eff:fe3c:8%11
                           10.1.1.1

```

You confirm that IPv6 resources are accessible by pinging 2001:db8:d::1 in Example 10-35, and it is successful. In addition, you contact the user at PC2, and she indicates that everything is fine now.

Example 10-35 *Successful Ping from PC1 to Web Server at 2001:db8:d::1*

```

C:\PC1>ping 2001:db8:d::1
Pinging 2001:db8:d::1 with 32 bytes of data:
Reply from 2001:db8:d::1: time=37ms
Reply from 2001:db8:d::1: time=35ms
Reply from 2001:db8:d::1: time=38ms
Reply from 2001:db8:d::1: time=38ms

Ping statistics for 2001:db8:d::1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 35ms, Maximum = 38ms, Average = 36ms

```

Exam Preparation Tasks

As mentioned in the section “How to Use This Book” in the Introduction, you have a couple of choices for exam preparation: the exercises here; Chapter 22, “Final Preparation;” and the exam simulation questions on the CD-ROM.

Review All Key Topics

Review the most important topics in this chapter, noted with the Key Topic icon in the outer margin of the page. Table 10-3 lists a reference of these key topics and the page numbers on which each is found.



Table 10-3 *Key Topics for Chapter 10*

Key Topic Element	Description	Page Number
Paragraph	Explains the Neighbor Solicitation	371
Example 10-1	Verifying IPv6 multicast groups a router interface is listening to	372
Paragraph	Describes the EUI-64 process	373
Example 10-5	Verifying EUI-64 on a router interface	375
Example 10-7	Enabling SLAAC on a router interface	376
Paragraph	Explains the Router Advertisement process	377
Paragraph	Identifies how to verify SLAAC-generated IPv6 addresses	377
List	Describes issues that may occur while using SLAAC	378
Example 10-11	Verifying an interface is enabled for IPv6	379
Example 10-12	Verifying that RAs are not suppressed	379
Example 10-13	Verifying default gateways configured on a PC	380
Example 10-15	Sample DHCPv6 configuration on R1	381
Example 10-16	Verifying DHCPv6 Information on R1	381
Example 10-17	Verifying stateless DHCPv6	382
Step list	Describes the four-way negotiation process of DHCPv6	384
Example 10-19	Verifying IPv6 configuration on PC1	385

Define Key Terms

Define the following key terms from this chapter and check your answers in the glossary:

Neighbor Solicitation, Neighbor Advertisement, Neighbor Discovery, solicited node multicast addresses, EUI-64, stateless autoconfiguration (SLAAC), stateful DHCPv6, stateless DHCPv6, router solicitation, router advertisement, link-local address, global unicast address, SOLICIT message, ADVERTISE message, REQUEST message, REPLY message, DHCPv6 relay agent

Command Reference to Check Your Memory

This section includes the most important **show** commands covered in this chapter. It might not be necessary to memorize the complete syntax of every command, but you should be able to remember the basic keywords that are needed.

To test your memory of the commands, cover the right side of Table 10-4 with a piece of paper, read the description on the left side, and then see how much of the command you can remember.

The 300-135 TSHOOT exam focuses on practical, hands-on skills that are used by a networking professional. Therefore, you should be able to identify the commands needed to successfully troubleshoot the topic covered in this chapter.

Table 10-4 *show Commands*

Task	Command Syntax
Displays if IPv6 is enabled on an interface, displays the multicast groups the router interface is a member of, displays the global and link-local unicast addresses associated with an interface, indicates whether EUI-64 was used or stateless autoconfiguration was used to obtain the IPv6 address for the interface, displays whether RAs are suppressed for the interface, and displays how devices connected to the same link as the interface will obtain an IPv6 address and how they will obtain other options	show ipv6 interface <i>interface_type interface_number</i>
Displays the IPv6 addresses that are being used by each of the DHCPv6 clients.	show ipv6 dhcp binding
Displays which DHCPv6 pool is assigned to which interface on the router	show ipv6 dhcp interface
Displays the configured DHCPv6 pools on the router	show ipv6 dhcp pool



This chapter covers the following topics:

- **Troubleshooting IPv4 ACLs:** This section examines how you can read IPv4 ACLs so that you are more efficient at troubleshooting IPv4 ACL-related issues. You will also learn the commands and processes that you can use while troubleshooting IPv4 packet filtering with standard, extended, and time-based IPv4 ACLs.
- **IPv4 ACL Trouble Tickets:** This section provides trouble tickets that demonstrate how you can use a structured troubleshooting process to solve a reported problem.
- **Troubleshooting IPv6 ACLs:** This section examines how you can read IPv6 ACLs so that you are more efficient at troubleshooting IPv6 ACL-related issues. You will also discover the commands and processes that you can use while troubleshooting IPv6 packet filtering.
- **IPv6 ACL Trouble Tickets:** This section provides trouble tickets that demonstrate how you can use a structured troubleshooting process to solve a reported problem.
- **Troubleshooting Prefix Lists:** This section reviews how to efficiently examine a prefix list for troubleshooting purposes so that when you are dealing with an issue that has a prefix list associated with it, you can determine whether the prefix list is or is not the problem.
- **Prefix List Trouble Tickets:** This section provides trouble tickets that demonstrate how you can use a structured troubleshooting process to solve a reported problem.

Troubleshooting IPv4 and IPv6 ACLs and Prefix Lists

Access control lists (ACLs) and prefix lists are powerful tools that you need to be comfortable with as a troubleshooter. They enable you to classify traffic or routes, and then depending on how you apply them, take a specific action. One slight error in an ACL or prefix list will change the meaning of it and, as a result, how the service or feature that relies on it handles the route or traffic.

Therefore, you need to be able to read ACLs and prefix lists efficiently. You need a solid understanding of the way they are processed and how the devices using them make a decision based on the entries. Without this knowledge, you cannot successfully eliminate or prove that the ACL or prefix list is the problem.

This chapter covers the ins and outs of ACLs and prefix lists. You will learn the way they are processed, how they are read, and how you can identify issues related to them. In addition, this chapter explains how you can use ACLs for traffic filtering and how a prefix list can be used for route filtering.

“Do I Know This Already?” Quiz

The “Do I Know This Already?” quiz allows you to assess whether you should read this entire chapter thoroughly or jump to the “Exam Preparation Tasks” section. If you are in doubt about your answers to these questions or your own assessment of your knowledge of the topics, read the entire chapter. Table 11-1 lists the major headings in this chapter and their corresponding “Do I Know This Already?” quiz questions. You can find the answers in Appendix A, “Answers to the ‘Do I Know This Already?’ Quizzes.”

Table 11-1 “Do I Know This Already?” Section-to-Question Mapping

Foundation Topics Section	Questions
Troubleshooting IPv4 ACLs	1–4
Troubleshooting IPv6 ACLs	5–7
Troubleshooting Prefix Lists	8–10

Caution The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you do not know the answer to a question or are only partially sure of the answer, you should mark that question as wrong for purposes of the self-assessment. Giving yourself credit for an answer that you correctly guess skews your self-assessment results and might provide you with a false sense of security.

1. What is the correct order of operations for an IPv4 ACL?
 - a. Top-down processing, execute upon the longest match, implicit deny all
 - b. Execute upon the longest match, top down processing, implicit deny all
 - c. Implicit deny all, immediate execution upon a match, top-down processing
 - d. Top-down processing, immediate execution upon a match, implicit deny all
2. What occurs to a packet when an ACL is applied to an interface but the packet does not match any of the entries in the ACL?
 - a. It is forwarded.
 - b. It is flooded.
 - c. It is dropped.
 - d. It is buffered.
3. What will the following ACL entry accomplish when applied to an interface: 20 permit tcp 10.1.1.0 0.0.0.63 host 192.0.2.1 eq 23?
 - a. Permit Telnet traffic from the device with an IP address of 192.0.2.1 going to any device with an IP address from 10.1.1.0 to 10.1.1.63
 - b. Permit Telnet traffic from any device with an IP address from 10.1.1.0 to 10.1.1.63 going to the device with an IP address of 192.0.2.1
 - c. Permit SSH traffic from any device with an IP address from 10.1.1.0 to 10.1.1.63 going to the device with an IP address of 192.0.2.1
 - d. Permit SSH traffic from the device with an IP address of 192.0.2.1 going to any device with an IP address from 10.1.1.0 to 10.1.1.63
4. Which command will successfully filter ingress traffic using ACL 100 on an interface?
 - a. access-group 100 in
 - b. access-class 100 in
 - c. ip access-group 100 in
 - d. ip traffic-filter 100 in

5. What is the correct order of operations for an IPv6 ACL?
 - a. Immediate execution upon a match, implicit permit icmp nd, implicit deny all, top-down processing
 - b. Top-down processing, immediate execution upon a match, implicit permit icmp nd, implicit deny all
 - c. Top-down processing, implicit permit icmp nd, immediate execution upon a match, implicit deny all
 - d. Implicit permit icmp nd, top-down processing, immediate execution upon a match, implicit deny all
6. What will happen if you add the following entry to the end of an IPv6 ACL: **deny ipv6 any any log**? (Choose two answers.)
 - a. All traffic will be denied and logged.
 - b. All traffic that does not match an entry in the ACL will be denied and logged.
 - c. ICMP Neighbor Discovery messages will still be implicitly permitted.
 - d. ICMP Neighbor Discovery messages will be denied.
7. Which command will successfully filter egress traffic using an IPv6 ACL named TSHOOT on an interface?
 - a. `access-group TSHOOT out`
 - b. `access-class TSHOOT out`
 - c. `ipv6 access-group TSHOOT out`
 - d. `ipv6 traffic-filter TSHOOT out`
8. Which IP prefix list will match only the default route?
 - a. `ip prefix-list TSHOOT permit 0.0.0.0 le 32`
 - b. `ip prefix-list TSHOOT permit 0.0.0.0 ge 32`
 - c. `ip prefix-list TSHOOT permit 0.0.0.0 ge 1`
 - d. `ip prefix-list TSHOOT permit 0.0.0.0`
9. Which IP prefix list will match all routes?
 - a. `ip prefix-list TSHOOT permit 0.0.0.0 le 32`
 - b. `ip prefix-list TSHOOT permit 0.0.0.0 ge 32`
 - c. `ip prefix-list TSHOOT permit 0.0.0.0 ge 1`
 - d. `ip prefix-list TSHOOT permit 0.0.0.0`

- 10.** What routes match the following prefix list: **ip prefix-list TSHOOT seq 35 deny 192.168.0.0/20 ge 24 le 28?**
- a.** Routes with an address from 192.168.0.0 to 192.168.15.255 with a subnet mask of 24 to 28
 - b.** Routes within the 192.168.0.0/20 subnet with a subnet mask greater than 24 and less than 28
 - c.** Routes with the subnet ID and mask of 192.168.0.0/20
 - d.** Routes with an address from 192.168.0.0 to 192.168.15.255 with a subnet mask of 24 or 28

Foundation Topics

Troubleshooting IPv4 ACLs

The purpose of an access control list is to identify traffic based on different criteria such as source or destination IP address, source or destination port numbers, transport layer protocols, quality of service (QoS) markings, and so on. An ACL that has been created does nothing unless it is applied to a service, feature, or interface. For example, it can be used to identify the private IP addresses that will be translated to a public address with Network Address Translation (NAT) and Port Address Translation (PAT). It can also be used to control which routes will be redistributed, which packets will be policy-based routed, and which packets will be permitted or denied through the router. Therefore, as a troubleshooter, it is imperative that you can read an ACL to determine whether it was created correctly; otherwise, the services you are applying it to will fail to produce the results you want.

This section explains how to troubleshoot an IPv4 ACL to make sure that it is correctly created for the purpose it is intended for. The section also provides examples related to packet filtering. Other examples related to distribute lists, route maps, and policy-based routing (PBR) are covered in later chapters relating to those features.

Reading an IPv4 ACL

Being able to read an ACL and understand what it was created for is important for troubleshooting. However, understanding how an ACL functions is even more important as you troubleshoot because you need to identify why you are experiencing the issues that are occurring. Following is a list of steps that IPv4 ACLs use. You want to remember these steps because they will help you identify why an IPv4 ACL is behaving the way it is.



- Step 1. Top down processing:** An ACL is made up of various entries; these entries are processed from the top of the ACL to the bottom of the ACL in order.
- Step 2. Immediate execution upon a match:** The very first entry that matches the values in the packet that are being compared will be the entry that is used. This may be a permit entry or a deny entry and will dictate how the packet is treated based on the ACL implementation. If there is another entry later in the ACL that matches, it does not matter. Only the first entry that matches matters.
- Step 3. Implicit deny any:** If there is no matching entry for the packet, the packet is automatically denied based on the invisible implicit deny any entry at the end of an ACL.



Refer to Example 11-1, which displays a sample standard numbered ACL that uses only source IPv4 addresses. In this example, the ACL is numbered 1 and has four entries. The entries are listed from most specific to least specific. In earlier versions of the IOS, if you

did not create the entries from most specific to least specific, you ended up with generic entries earlier in the ACL that would cause issues by dropping or permitting traffic that should not be. In newer versions of the IOS, if you attempt to create an ACL entry that is more specific than an entry that already exists, the router will prevent the entry from being created and give an error message.

Notice how traffic sourced from 10.1.1.5 is denied in sequence 5. Even though the very next sequence of 10 permits 10.1.1.5, 10.1.1.5 will be denied because of top-down processing and then immediate execution upon a match. Likewise, even though sequence 30 permits all addresses from 10.1.1.0 through 10.1.1.255, 10.1.1.5 is denied by sequence 5, and 10.1.1.64 through 10.1.1.127 are denied by sequence 20. What about all other source IP addresses that do not match an entry in the ACL? For example, the IP address 192.168.2.1. They are all denied because of the implicit deny entry (you cannot see it) at the end of the ACL.

Example 11-1 Sample Standard Numbered ACL

```
Router#show access-lists
Standard IP access list 1
    5 deny    10.1.1.5
    10 permit 10.1.1.0, wildcard bits 0.0.0.63 (1 match)
    20 deny   10.1.1.64, wildcard bits 0.0.0.63
    30 permit 10.1.1.0, wildcard bits 0.0.0.255
```

Extended ACLs are a little more complicated to read and troubleshoot because they contain more parameters. The previous example was a standard ACL that only allows a source address to be specified. The extended ACL can take source and destination addresses, source and destination port numbers, protocols, and other parameters that give you granular control over what you are trying to match. Also remember that standard and extended IPv4 ACLs can be named instead of numbered.



Example 11-2 provides a sample extended numbered ACL. In this example, it is numbered 100. It has four entries, listed from most specific to least specific. Notice in sequence 10 that 10.1.1.5 is denied from accessing TCP services using port 80 on 192.0.2.1. At the same time, under sequence 20, 10.1.1.5 would be permitted to telnet to 192.0.2.1, and in sequence 40, it would be permitted to any destination on any port using any protocol. Therefore, you have much more granular control over how the traffic will be matched in an extended ACL.

Example 11-2 Sample Extended Numbered ACL

```
R1#show access-lists 100
Extended IP access list 100
    10 deny tcp host 10.1.1.5 host 192.0.2.1 eq www
    20 permit tcp 10.1.1.0 0.0.0.63 host 192.0.2.1 eq telnet
    30 deny ip 10.1.1.64 0.0.0.63 host 192.0.2.1
    40 permit ip 10.1.1.0 0.0.0.255 any
```

Using an IPv4 ACL for Filtering

Using an ACL for packet filtering requires you to apply the ACL to an interface. You can accomplish this with the **ip access-group {acl_number/name} {in|out}** command in interface configuration mode, as shown in Example 11-3. The direction you apply the ACL on an interface is significant. You need to consider this while you are creating the ACL. If you apply it to the wrong interface or in the wrong direction, you will not get the desired result. You can verify the ACLs that are applied to an interface using the **show ip interface interface_type interface_number** command. Example 11-3 shows how access list 1 is applied inbound on Gig0/0 and access list 100 is applied outbound on Gig0/0.



Example 11-3 Verifying Access Lists Applied to Interfaces

```
R1(config)#interface gigabitEthernet 0/0
R1(config-if)#ip access-group 100 out
R1(config-if)#ip access-group 1 in
R1(config-if)#end
R1#show ip interface gigabitEthernet 0/0
GigabitEthernet0/0 is up, line protocol is up
  Internet address is 10.1.1.1/24
  Broadcast address is 255.255.255.255
  Address determined by non-volatile memory
  MTU is 1500 bytes
  Helper address is 172.16.1.10
  Directed broadcast forwarding is disabled
  Multicast reserved groups joined: 224.0.0.5 224.0.0.6
  Outgoing access list is 100
  Inbound access list is 1
  Proxy ARP is enabled
  Local Proxy ARP is disabled
```

Using a Time-Based IPv4 ACL

By default, an ACL you apply is active the entire time it is applied. However, that might not be your goal. For example, perhaps you want to prevent traffic from going to the Internet after hours but allow it during hours. Or give a certain service or user the ability to back up files to a server from 9 p.m. to 1 a.m. Monday to Friday and prevent them from doing it any other time.

To accomplish these goals, you need to use time-based ACLs. Review Example 11-4, which provides a sample time-based ACL. Notice that the ACL entry with a sequence number of 10 has the time-range option added. The time range is based on values configured in the AFTERHOURS time range. It also states that it is active, meaning that the current entry will be denying WWW traffic from host 10.1.1.5 to 192.0.2.1. Because the ACL entry is attached to a time range, when troubleshooting time-based ACLs you will also have to review the configuration of the time range itself. Example 11-5 displays the AFTERHOURS time range with the **show time-range AFTERHOURS** command. It has

two *weekdays* entries, one from 5 p.m. to midnight and the other from midnight to 9 a.m. It also has a *weekend* entry that covers all day and all night. It also states that it is active and used in an ACL. When the access control entry is outside of the time range, it will display inactive.



Example 11-4 Sample Time-Based ACL

```
R1#show access-lists 100
Extended IP access list 100
 10 deny tcp host 10.1.1.5 host 192.0.2.1 eq www time-range AFTERHOURS (active)
 20 permit tcp 10.1.1.0 0.0.0.63 host 192.0.2.1 eq telnet
 30 deny ip 10.1.1.64 0.0.0.63 host 192.0.2.1
 40 permit ip 10.1.1.0 0.0.0.255 any
```

Example 11-5 Sample Time Range Configured on R1

```
R1#show time-range AFTERHOURS
time-range entry: AFTERHOURS (active)
  periodic weekdays 17:00 to 23:59
  periodic weekdays 0:00 to 8:59
  periodic weekend 0:00 to 23:59
used in: IP ACL entry
```

So far, you have seen that you have to troubleshoot the ACL, and the time range when dealing with issues related to time-based ACLs. However, there is one more item of troubleshooting: *time*!

Time-based ACLs are based on the router clock. If the router clock is not correct, the time-based ACL may be active or inactive at the wrong time. Example 11-6 shows how you can verify the current time on a router with the **show clock** command. Notice how it is Sunday May 25, 2014, at 10:53 a.m. Therefore, the time-based ACL entry should be active because it is AFTERHOURS. We only want to permit WWW traffic Monday to Friday 9 a.m. to 5 p.m. All other times, it is denied.

Example 11-6 Viewing the Time on a Cisco Router

```
R1#show clock
*10:53:50.067 UTC Sun May 25 2014
```

But wait, are we sure it is the right time? Are we using manually set clocks, have they changed? Or are we using a Network Time Protocol (NTP) server? You will want to verify with another time source that this is in fact the right time. In addition, if you are using NTP (which you should be), you need to check your NTP settings to make sure that the clocks are synchronized and that the time is right, and do not forget to consider daylight savings time.

IPv4 ACL Trouble Tickets

This section presents various trouble tickets relating to the topics discussed earlier in the chapter. The purpose of these trouble tickets is to give a process that you can follow when troubleshooting in the real world or in an exam environment. All trouble tickets in this section are based on the topology depicted in Figure 11-1.

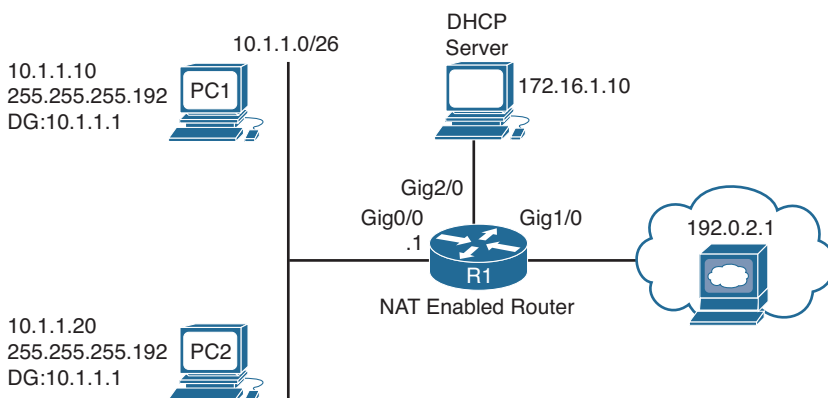


Figure 11-1 *IPv4 ACL Trouble Ticket Topology*

Trouble Ticket 11-1

Problem: A user at PC1 has indicated that he cannot telnet to 192.0.2.1 and he needs to. However, he can ping 192.0.2.1 and access web-enabled resources.

You start by verifying the problem. On PC1, you attempt to telnet to 192.0.2.1, but it fails, as shown in Example 11-7. You then ping 192.0.2.1, and it is successful, as also shown in Example 11-7.

Example 11-7 *Failed Telnet and Successful Ping from PC1 to 192.0.2.1*

```
C:\PC1>telnet 192.0.2.1
Connecting To 192.0.2.1...Could not open connection to the host, on port 23: Connect
failed

C:\PC1>ping 192.0.2.1
Reply from 192.0.2.1: bytes=32 time 1ms TTL=128
Reply from 192.0.2.1: bytes=32 time 1ms TTL=128
Reply from 192.0.2.1: bytes=32 time 1ms TTL=128
Reply from 192.0.2.1: bytes=32 time 1ms TTL=128

Ping statistics for 192.0.2.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

At this point, you should be thinking that the issue is related to either the Telnet service being disabled on 192.0.2.1 or an ACL. Why an ACL? This is because certain types of traffic are allowed through but others are not, which is accomplished with filtering.

First, let's verify whether there are any ACLs configured on R1 that may filter Telnet-related traffic. In Example 11-8, the **show ip access-lists** command is used to verify whether any ACLs are configured on R1. In this example, there is one extended IPv4 ACL identified as number 100. You can see that there are two entries related to Telnet. One is a permit entry with a sequence number of 10, and the other is a deny entry with a sequence number of 20. Notice how the deny entry has 9 matches and the permit entry has no matches. Read sequence 10 out loud:

Sequence 10 will permit tcp traffic related to telnet from 192.0.2.1 to 10.1.1.10.

Read it again and think about how the traffic is flowing based on this entry:

FROM 192.0.2.1 TO 10.1.1.10

PC1 is trying to establish a Telnet session to 192.0.2.1 (not the other way around). Therefore, sequence 10 does not match the Telnet traffic from PC1 to 192.0.2.1. It matches Telnet traffic from 192.0.2.1 to PC1.

Sequence 20 states that TCP traffic related to Telnet from the 10.1.1.0/26 network to any destination will be denied. Therefore, using the top-down processing and immediate execution upon a match flow, sequence 20 matches the Telnet traffic from PC1 to 192.0.2.1, and as a result, the traffic is denied.

Example 11-8 Verifying ACLs Configured on R1

```
R1#show ip access-lists
Extended IP access list 100
 10 permit tcp host 192.0.2.1 host 10.1.1.10 eq telnet
 20 deny tcp 10.1.1.0 0.0.0.63 any eq telnet (9 matches)
 30 deny tcp 10.1.1.0 0.0.0.63 any eq ftp
 40 permit tcp 10.1.1.0 0.0.0.63 any eq 22
 50 deny tcp 10.1.1.0 0.0.0.63 any eq smtp
 60 permit ip any any (2 matches)
```

The best way to fix this is to remove sequence 10 and replace it with the correct entry. We can use named ACL configuration mode to accomplish this. Example 11-9 displays how you can use named ACL configuration mode to edit a numbered ACL and the output of **show ip access-lists**, which verifies that the changes were made.

Example 11-9 Using Named ACL Configuration Mode to Modify Numbered ACL

```
R1#config t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#ip access-list extended 100
R1(config-ext-nacl)#no 10
R1(config-ext-nacl)#10 permit tcp host 10.1.1.10 host 192.0.2.1 eq 23
```

```

R1(config-ext-nacl)#end
R1#
R1#show access-lists
Extended IP access list 100
 10 permit tcp host 10.1.1.10 host 192.0.2.1 eq telnet
 20 deny tcp 10.1.1.0 0.0.0.63 any eq telnet (9 matches)
 30 deny tcp 10.1.1.0 0.0.0.63 any eq ftp
 40 permit tcp 10.1.1.0 0.0.0.63 any eq 22
 50 deny tcp 10.1.1.0 0.0.0.63 any eq smtp
 60 permit ip any any (4 matches)

```

As shown in Example 11-10, you issue the **telnet 192.0.2.1** command from PC1, and the connection is successful.

Example 11-10 Successful Telnet Connection from PC1 to 192.0.2.1

```

C:\PC1>telnet 192.0.2.1
User Access Verification
Password:

```

Reviewing the output of **show ip access-lists** on R1, as shown in Example 11-11, reveals the matches associated with sequence 10 now.

Example 11-11 Verifying Packet Matches For an ACL Entry

```

R1#show ip access-lists
Extended IP access list 100
 10 permit tcp host 10.1.1.10 host 192.0.2.1 eq telnet (25 matches)
 20 deny tcp 10.1.1.0 0.0.0.63 any eq telnet (9 matches)
 30 deny tcp 10.1.1.0 0.0.0.63 any eq ftp
 40 permit tcp 10.1.1.0 0.0.0.63 any eq 22
 50 deny tcp 10.1.1.0 0.0.0.63 any eq smtp
 60 permit ip any any (5 matches)

```

Troubleshooting IPv6 ACLs

IPv6 ACLs play an important role in our IPv6 networks. They allow us to classify traffic for many different reasons. For example, we might need to classify traffic that will be policy-based routed, or we may need to classify the traffic that will be filtered as it passes through the router.

IPv6 traffic filtering can be done on an interface-by-interface basis with IPv6 access lists. This section explains how to read an IPv6 access list so that you can troubleshoot them efficiently and identify whether they have been applied correctly to an interface for filtering purposes.

Reading an IPv6 ACL

Being able to read an IPv6 ACL and understand what it was created for is important for troubleshooting. However, understanding how an IPv6 ACL functions is even more important as you troubleshoot because you need to identify why you are experiencing the issues that are occurring. Following is a list of steps that IPv6 ACLs use; these are the same as IPv4 ACLs. You want to remember these steps because they will help you identify why an IPv6 ACL is behaving the way it is:



- Step 1. Top-down processing:** An ACL is made up of various entries; these entries are processed from the top of the ACL to the bottom of the ACL in order.
- Step 2. Immediate execution upon a match:** The very first entry that matches the values in the packet that are being compared will be the entry that is used. This may be a permit entry or a deny entry and will dictate how the packet is treated based on the ACL implementation. If there is another entry later in the ACL that matches, it does not matter. Only the first entry that matches matters.
- Step 3. Implicit permit icmp nd:** If the packet is an NA or NS message, permit it.
- Step 4. Implicit deny any:** If there is no matching entry for the packet, the packet is automatically denied based on the invisible implicit deny any entry at the end of an ACL.

Pause here for a moment. Did you notice the steps differ a little from IPv4? There is an added step before the implicit deny any. Recall that IPv6 relies on the Neighbor Discovery Protocol (NDP) NA and NS messages to determine the MAC address associated with an IPv6 address. Therefore, the *implicit permit icmp nd* entries for NA and NS messages as follows have been added before the implicit deny any, so they are not denied:

```
permit icmp any any nd-na
permit icmp any any nd-ns
```

However, because these are implicit permit statements, all statically entered commands come before them. Therefore, if you issue the **deny ipv6 any any log** command at the end of your IPv6 ACL like you might be accustomed to doing in IPv4, you will break the NDP process because NA and NS messages will be denied now. Therefore, when troubleshooting NDP, an ACL might be the reason why it is not working.

With IPv4 ACLs, a clear separation existed between standard and extended IPv4 ACLs. However, with IPv6, you have just one type, which would be similar to an IPv4 extended ACL. Therefore, within an IPv6 ACL entry, you provide as little or as much information as you need to accomplish your goal.

Refer to Example 11-12, which provides a sample IPv6 ACL that was created on R1. The IPv6 access list is named TSHOOT, and you read it exactly like you read an IPv4 ACL. For example, sequence 20 states that TCP traffic related to Telnet will be denied from any device going to 2001:DB8:A:B::7/128. Sequence 30 states that TCP traffic related to WWW from 2001:DB8:A:A::20/128 to 2001:DB8:D::1/128 will be permitted.

Key Topic
Example 11-12 *Sample IPv6 ACL*

```
R1#show ipv6 access-list
IPv6 access list TSHOOT
    permit tcp host 2001:DB8:A:A::20 host 2001:DB8:A:B::7 eq telnet sequence 10
    deny tcp any host 2001:DB8:A:B::7 eq telnet sequence 20
    permit tcp host 2001:DB8:A:A::20 host 2001:DB8:D::1 eq www sequence 30
    deny ipv6 2001:DB8:A:A::/80 any sequence 40
    permit ipv6 2001:DB8:A:A::/64 any sequence 50
```

Notice how there are no wildcard masks with IPv6. Instead, you specify a prefix, as shown in sequence 40 and 50 of Example 11-12, which accomplishes the same goal as the wildcard mask (defining a range of addresses). For example, a prefix of /128 is like having the all 0s wildcard mask, which would mean this exact address or host (match all bits in the address). A /0 prefix is like having the all 255s wildcard mask (do not match any bits in the address). A /64 prefix would indicate that the first 64 bits must match and that the last 64 bits do not have to match. As a result, this would include all interface IDs within a /64 network. What if the prefix is /80? This means the first 80 bits must match and the last 48 bits do not have to match. As a result, the prefix is defining which bits of the IPv6 address must match.

Using an IPv6 ACL for Filtering

Using an IPv6 ACL for packet filtering requires you to apply the IPv6 ACL to an interface. You can accomplish this with the **ipv6 traffic-filter *acl_name* {in|out}** command in interface configuration mode, as shown in Example 11-13. The direction you apply the IPv6 ACL on an interface is significant. It needs to be considered while you are creating the ACL. If you apply it to the wrong interface or in the wrong direction, you will not get the desired result. You can verify the IPv6 ACLs that are applied to an interface using the **show ipv6 interface *interface_type interface_number*** command. Example 11-13 shows how the IPv6 access-list TSHOOT is applied inbound on interface Gig0/0.

Key Topic
Example 11-13 *Verifying IPv6 Access Lists Applied to Interfaces*

```
R1(config)#interface gigabitEthernet 0/0
R1(config-if)#ipv6 traffic-filter TSHOOT in
R1(config-if)#end
R1#show ipv6 interface gigabitEthernet 0/0
GigabitEthernet0/0 is up, line protocol is up
    IPv6 is enabled, link-local address is FE80::C808:3FF:FE78:8
    No Virtual link-local address(es):
    Global unicast address(es):
        2001:DB8:A:A::1, subnet is 2001:DB8:A:A::/64
    Joined group address(es):
        FF02::1
        FF02::2
        FF02::1:2
```

```

FF02::1:FF00:1
FF02::1:FF78:8
MTU is 1500 bytes
ICMP error messages limited to one every 100 milliseconds
ICMP redirects are enabled
ICMP unreachable are sent
Input features: Access List
Inbound access list TSHOOT
ND DAD is enabled, number of DAD attempts: 1
ND reachable time is 30000 milliseconds (using 30000)
ND advertised reachable time is 0 (unspecified)
ND advertised retransmit interval is 0 (unspecified)
ND router advertisements are sent every 200 seconds
ND router advertisements live for 1800 seconds
ND advertised default router preference is Medium
Hosts use stateless autoconfig for addresses.
Hosts use DHCP to obtain other configuration.

```

IPv6 ACL Trouble Tickets

This section presents various trouble tickets relating to the topics discussed earlier in the chapter. The purpose of these trouble tickets is to give a process that you can follow when troubleshooting in the real world or in an exam environment. All trouble tickets in this section are based on the topology depicted in Figure 11-2.

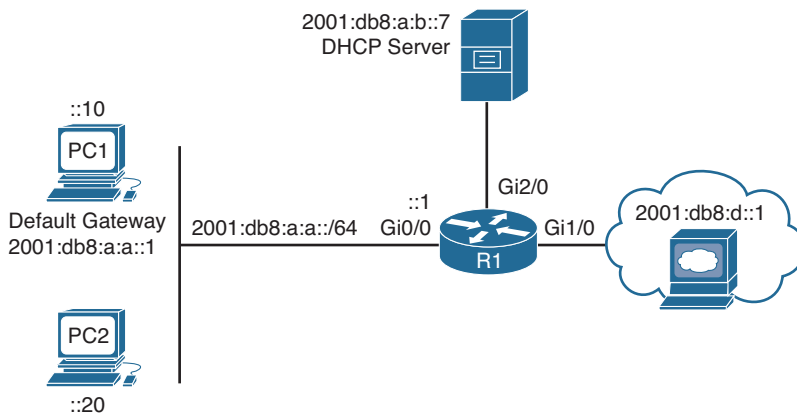


Figure 11-2 IPv6 ACL Trouble Ticket Topology

Trouble Ticket 11-2

Problem: A user at PC2 has indicated that she is not able to telnet to 2001:db8:a:b::7 and she needs to. However, she can ping 2001:db8:a:b::7 and receive DHCP-related information from the DHCP server.

You start by verifying the problem. On PC2, you attempt to telnet to 2001:db8:a:b::7, but it fails, as shown in Example 11-14. You then ping 2001:db8:a:b::7, and it is successful, as also show in Example 11-14.

Example 11-14 *Failed Telnet and Successful Ping from PC2 to 2001:db8:a:b::7*

```
C:\PC2>telnet 2001:db8:a:b::7
Connecting To 2001:db8:a:B::7...Could not open connection to the host, on port 23:
Connect failed

C:\PC2>ping 2001:db8:a:b::7

Pinging 2001:db8:a:b::7 with 32 bytes of data:
Reply from 2001:db8:a:b::7: time=46ms
Reply from 2001:db8:a:b::7: time=40ms
Reply from 2001:db8:a:b::7: time=40ms
Reply from 2001:db8:a:b::7: time=40ms

Ping statistics for 2001:db8:a:b::7:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 40ms, Maximum = 46ms, Average = 41ms
```

What could allow pings yet deny Telnet? At this point in time, you should be thinking the issue is related to either the Telnet service being disabled on 2001:db8:a:b::7 or an IPv6 ACL filtering traffic in or out of an interface. This is because certain traffic is allowed while others are denied. Most times, this is because of traffic filtering.

First, you verify whether the Telnet service is running by using Telnet from R1 to 2001:db8:a:b::7. As shown in Example 11-15, it is successful. If it was not successful, you could then access the server or contact the users responsible for the server to see whether Telnet is enabled.

Example 11-15 *Successful Telnet from R1 to 2001:db8:a:b::7*

```
R1#telnet 2001:db8:a:b::7
Trying 2001:DB8:A:B::7 ... Open

User Access Verification

Password:
```

Next you check whether there are any ACLs associated with interface Gi2/0 on R1 using the command **show ipv6 interface gigabitethernet2/0**. As shown in Example 11-16, there are no IPv6 ACLs.

Example 11-16 *Verifying ACLs on Gig2/0 of R1*

```

R1#show ipv6 interface gigabitEthernet 2/0
GigabitEthernet2/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::C808:3FF:FE78:38
  No Virtual link-local address(es):
  Global unicast address(es):
    2001:DB8:A:B::1, subnet is 2001:DB8:A:B::/64
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF00:1
    FF02::1:FF78:38
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ICMP unreachables are sent
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds (using 30000)
  ND advertised reachable time is 0 (unspecified)
  ND advertised retransmit interval is 0 (unspecified)
  ND router advertisements are sent every 200 seconds
  ND router advertisements live for 1800 seconds
  ND advertised default router preference is Medium
  Hosts use stateless autoconfig for addresses.

```

Next you check whether there are any ACLs associated with interface Gi0/0 on R1 by using the command **show ipv6 interface gigabitethernet0/0**. As shown in Example 11-17, there is an inbound IPv6 ACL named TSHOOT attached to the interface.

Example 11-17 *Verifying ACLs on Gig0/0 of R1*

```

R1#show ipv6 interface gigabitEthernet 0/0
GigabitEthernet0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::C808:3FF:FE78:8
  No Virtual link-local address(es):
  Global unicast address(es):
    2001:DB8:A:A::1, subnet is 2001:DB8:A:A::/64
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:2
    FF02::1:FF00:1
    FF02::1:FF78:8
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled

```

```

ICMP unreachable are sent
Input features: Access List
Inbound access list TSHOOT
ND DAD is enabled, number of DAD attempts: 1
ND reachable time is 30000 milliseconds (using 30000)
ND RAs are suppressed (all)
Hosts use stateless autoconfig for addresses.
Hosts use DHCP to obtain other configuration.

```

Now you need to verify the IPv6 ACL named TSHOOT using the **show ipv6 access-list TSHOOT** command. Example 11-18 displays this output. Notice sequence 20. It is a permit statement allowing PC2 to telnet to 2001:db8:a:b::7. However, notice sequence 10. It is a deny statement preventing all devices from using Telnet to 2001:db8:a:b::7. Remember that IPv6 ACLs are processed from top down, and then once a match is found, it is immediately executed on. That is what is happening here. Sequence 10 matches PC2's Telnet and denies it.

(Notice for IPv6 that the router allowed a more specific entry to be placed after a more general entry, this differs from the behavior witnessed with IPv4 ACLs earlier.)

Example 11-18 TSHOOT IPv6 ACL on R1

```

R1#show ipv6 access-list TSHOOT
IPv6 access list TSHOOT
  deny tcp any host 2001:DB8:A:B::7 eq telnet (6 matches) sequence 10
  permit tcp host 2001:DB8:A:A::20 host 2001:DB8:A:B::7 eq telnet sequence 20
  permit tcp host 2001:DB8:A:A::20 host 2001:DB8:D::1 eq www sequence 30
  permit ipv6 2001:DB8:A:A::/64 any (67 matches) sequence 40

```

To solve this issue, you connect to R1, enter IPv6 ACL configuration mode for the ACL named TSHOOT, and then you remove sequence 20 and add the same entry with a sequence number of 5 so that it is before sequence 10, as shown in Example 11-19. In addition, you verify the changes with the **show ipv6 access-list TSHOOT** command.

Example 11-19 Modifying TSHOOT IPv6 ACL on R1

```

R1#config t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#ipv6 access-list TSHOOT
R1(config-ipv6-acl)#no sequence 20
R1(config-ipv6-acl)#seq 5 permit tcp host 2001:DB8:A:A::20 host 2001:DB8:A:B::7 eq
telnet

R1#show ipv6 access-list TSHOOT
IPv6 access list TSHOOT
  permit tcp host 2001:DB8:A:A::20 host 2001:DB8:A:B::7 eq telnet sequence 5
  deny tcp any host 2001:DB8:A:B::7 eq telnet (6 matches) sequence 10
  permit tcp host 2001:DB8:A:A::20 host 2001:DB8:D::1 eq www sequence 30
  permit ipv6 2001:DB8:A:A::/64 any (67 matches) sequence 40

```

Now you go back to PC2 and attempt to telnet to 2001:db8:a:b::7. In Example 11-20, it is successful.

Example 11-20 *Successful Telnet from PC2 to 2001:db8:a:b::7*

```
C:\PC2>telnet 2001:db8:a:b::7

User Access Verification

Password:
```

Troubleshooting Prefix Lists

Although an ACL can give you extreme granular control of the traffic you want to match, it lacks the ability to identify routes based on a subnet mask. Therefore, ACLs do not give you granular control when matching routes for route filtering. This is why prefix lists exist. They allow you to define the route and prefix that you want to match. This section explains how to read a prefix list so that when you are troubleshooting features that call upon a prefix list you will have the ability to eliminate the prefix list as the cause of the issue or prove that the prefix list is the cause of the issue.

Note that this discussion applies to both IPv4 prefix lists and IPv6 prefix lists. The only difference is that in an IPv4 prefix list you will have IPv4 addresses and masks and in an IPv6 prefix list you will have IPv6 addresses and masks. However, the same principles and concepts apply. As a result, all the examples in this section are based on IPv4.

Reading a Prefix List

Let's begin with an example. Example 11-21 displays the commands used to create a sample prefix list called TSHOOT and the output of **show ip prefix-list**, which you can use to verify the IPv4 prefix lists configured on a router. To verify IPv6 prefix lists you use the command **show ipv6 prefix-list**.

Example 11-21 *Sample IPv4 Prefix List*

```
R1#config t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#ip prefix-list TSHOOT seq 10 deny 10.1.1.0/26
R1(config)#ip prefix-list TSHOOT seq 20 permit 10.1.1.0/24 le 32
R1(config)#ip prefix-list TSHOOT seq 30 permit 0.0.0.0/0
R1(config)#ip prefix-list TSHOOT seq 35 deny 192.168.0.0/20 ge 24 le 28
R1(config)#end
R1#show ip prefix-list
ip prefix-list TSHOOT: 3 entries
    seq 10 deny 10.1.1.0/26
    seq 20 permit 10.1.1.0/24 le 32
    seq 30 permit 0.0.0.0/0
    seq 35 deny 192.168.0.0/20 ge 24 le 28
```

There are two different ways to read a prefix list entry. The way you read a prefix list entry is based on whether there is a *le* (less than or equal to) or *ge* (greater than or equal to) at the end of the prefix list entry or not.



No ge or le: If the entry does not contain a *ge* or *le*, the prefix is treated as an address and a subnet mask. Refer to the entry with a sequence number of 10 in Example 11-21. There is no *ge* or *le*; therefore, the network 10.1.1.0/26 is matched exactly. For example, if you are using the prefix list to filter routing updates, the 10.1.1.0/26 network will be denied (meaning that it will be filtered and not used).

There is a ge or le: If the entry does contain a *ge* or *le*, the prefix is treated as an address and a *wildcard mask*. Refer to the entry with a sequence number of 20 in Example 11-21. Because there is a *ge* or *le*, the entry is defining a range of values. 10.1.1.0/24 really means 10.1.1.0 0.0.0.255 (where 0.0.0.255 is the inverse of the subnet mask), which indicates a range of addresses from 10.1.1.0 through 10.1.1.255 (just like an ACL). The *le* at the end means less than or equal to, and the 32 is referring to a subnet mask. Therefore, this entry is permitting any address from 10.1.1.0 through 10.1.1.255 with a subnet mask less than or equal to 32 (0 to 32). For example, if you are using the prefix list to filter routing updates, the 10.1.1.0/24, 10.1.1.64/26, and 10.1.1.128/30 networks would all be permitted because they fall within the prefix range and subnet mask range.

Refer to sequence 30 in Example 11-21. Because there is no *ge* or *le*, it will be an exact match to the address and mask listed. In this case, the address and mask are 0.0.0.0/0, which is the default route. Therefore, if this prefix list is being used to filter routing updates, the filter would permit the default route.

Refer to sequence 35 in Example 11-21. Because there is a *ge* or *le*, the address and mask are treated as an address and wildcard mask to define a range. Therefore, 192.168.0.0/20 is 192.168.0.0 0.0.15.255, which defines a range of 192.168.0.0 through 192.168.15.255. The *ge* 24 *le* 28 values specify a subnet mask range from 24 to 28. Therefore, if this prefix entry was used to filter routes, all routes with an address from 192.168.0.0 to 192.168.15.255 with a subnet mask of 24 to 28 will be denied.

Now it is your turn. Which routes will match the following prefix list:

```
ip prefix-list EXAMPLE permit 10.1.1.0/24 ge 26
```

Before you read any further, try to determine it on your own.

Because there is a *ge*, the /24 is treated as a wildcard mask of 0.0.0.255. Therefore, the range of routes are from 10.1.1.0 to 10.1.1.255. (The first 24 bits must match.) However, the *ge* 26 indicates that the routes also must have a subnet mask from 26 to 32. So, to sum up the prefix list, any route from 10.1.1.0 to 10.1.1.255 with a subnet mask from 26 to 32 will match this prefix list.

Prefix List Processing

Following is a list of steps that prefix lists use. You want to remember these steps as they will help you identify why a prefix list is behaving the way it is.



- Step 1. Top -down processing:** A prefix list is made up of various entries; these entries are processed from the top of the prefix list to the bottom of the prefix list in order of sequence number. In Example 11-21, sequence 10 is processed first, then 20, 30, 40.
- Step 2. Immediate execution upon a match:** The very first entry that matches will be the entry that is used. This may be a permit entry or a deny entry and will dictate how the information is treated. If there is another entry later in the prefix list that matches, it does not matter. Only the first entry that matches matters. For example, even though in Example 11-21 the 10.1.1.0/26 network falls within the range defined in sequence 20, which would permit it, it is denied in sequence 10, which is processed first. Therefore, 10.1.1.0/26 is denied.
- Step 3. Implicit deny any:** If there is no matching entry, the information is automatically denied based on the invisible implicit deny any entry at the end of a prefix list. For example, if the prefix list in Example 11-21 is used to filter routing updates, and an update is received for 172.16.32.0/29, it is denied because it does not match sequence 10, 20, 30, or 40.

Because there is an implicit deny any at the end of a prefix list, you need at least one permit statement in a prefix list or everything will be denied. For example, if you are creating a prefix list to deny a specific route or two (for example, 10.1.1.0/24 and 10.1.2.0/24) you would create the following entries:

```
ip prefix-list NAME seq 10 deny 10.1.1.0/24
ip prefix-list NAME seq 20 deny 10.1.2.0/24
```

Although this denies both prefixes, it also denies every other prefix because of the implicit deny any at the end. Therefore, to permit everything else, you need to include an entry that does so. The following entry would do just that:

```
ip prefix-list NAME seq 30 permit 0.0.0.0/0 le 32
```



Do not confuse this with the default route entry (seq 30) from Example 11-21. That did not have an `le` or `ge`. This example does. Let's review it. Because there is an `le`, it means address and wildcard mask. So, 0.0.0.0/0 is really 0.0.0.0 255.255.255.255. Therefore, the range is all/any addresses. The subnet mask will be `le 32`, which is 0 to 32. Therefore, we are permitting all routes in this entry. For IPv6, the equivalent permit all is as follows:

```
ipv6 prefix-list NAME seq 30 permit ::/0 le 128
```

Prefix List Trouble Tickets

This section presents various trouble tickets relating to the topics discussed earlier in the chapter. The purpose of these trouble tickets is to give a process that you can follow when troubleshooting in the real world or in an exam environment. All trouble tickets in this section are based on the topology depicted in Figure 11-3.

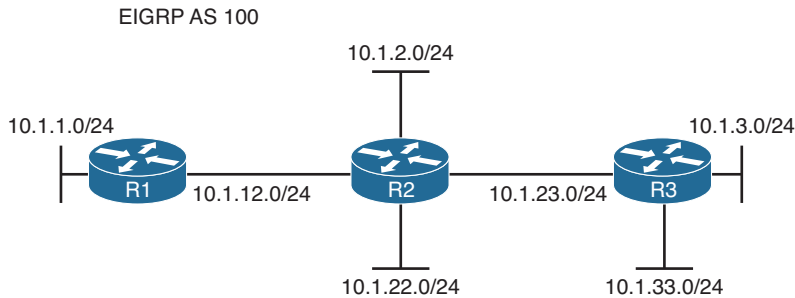


Figure 11-3 IPv4 Prefix List Trouble Ticket Topology

Trouble Ticket 11-3

Problem: Your junior admin has contacted you indicating that R1 is not learning any routes via Enhanced Interior Gateway Routing Protocol (EIGRP), as shown in Example 11-22. They have confirmed that neighbor relationships are being formed, interfaces are participating in the routing process, and that other routers are learning about the routes. They have come to you for help. With your extensive knowledge, you ask your junior admin if he checked for any route filters. He says no.

Example 11-22 Verifying Routes in R1's Routing Table

```

R1#show ip route
...output omitted...
Gateway of last resort is not set

    10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C       10.1.1.0/24 is directly connected, GigabitEthernet0/0
L       10.1.1.1/32 is directly connected, GigabitEthernet0/0
C       10.1.12.0/24 is directly connected, GigabitEthernet1/0
L       10.1.12.1/32 is directly connected, GigabitEthernet1/0
  
```

You execute the **show ip protocols** command on R1, as shown in Example 11-23. The output indicates that there is an inbound route filter using a prefix list called **FILTER_10.1.3.0**.

Example 11-23 Verifying Whether There Are Any Route Filters on R1

```

R1#show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "eigrp 100"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is (prefix-list) FILTER_10.1.3.0
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
...output omitted...
  
```

Next you issue the **show ip prefix-list** command on R1 to review the prefix list called **FILTER_10.1.3.0**, as shown in Example 11-24. In this output, you can see that the 10.1.3.0/24 prefix is being denied. Your junior admin states that this is not the problem, because 10.1.3.0/24 is supposed to be denied based on the documentation while all others are permitted. You respond by saying that you are very sure that it is the problem. You remind your junior admin about how prefix lists are processed: 1) top down, 2) immediate execution upon a match, 3) implicit deny any at the end. Therefore, this prefix list denies all prefixes not just 10.1.3.0/24.

Example 11-24 *Reviewing the Prefix List on R1*

```
R1#show ip prefix-list
ip prefix-list FILTER_10.1.3.0: 1 entries
seq 5 deny 10.1.3.0/24
```

To fix this problem you create another entry for the **FILTER_10.1.3.0** prefix list that permits all other routes as follows:

```
ip prefix-list FILTER_10.1.3.0 seq 10 permit 0.0.0.0/0 le 32
```

Example 11-25 displays the updated prefix list on R1, and Example 11-26 shows the updated routing table, which has all the routes except for 10.1.3.0/24, which is denied.

Example 11-25 *Reviewing the Updated Prefix List on R1*

```
R1#show ip prefix-list
ip prefix-list FILTER_10.1.3.0: 2 entries
seq 5 deny 10.1.3.0/24
seq 10 permit 0.0.0.0/0 le 32
```

Example 11-26 *Verifying Updated Routes in R1's Routing Table*

```
R1#show ip route
...output omitted...
Gateway of last resort is not set

    10.0.0.0/8 is variably subnetted, 8 subnets, 2 masks
C       10.1.1.0/24 is directly connected, GigabitEthernet0/0
L       10.1.1.1/32 is directly connected, GigabitEthernet0/0
D       10.1.2.0/24 [90/130816] via 10.1.12.2, 00:01:32, GigabitEthernet1/0
C       10.1.12.0/24 is directly connected, GigabitEthernet1/0
L       10.1.12.1/32 is directly connected, GigabitEthernet1/0
D       10.1.22.0/24 [90/130816] via 10.1.12.2, 00:01:32, GigabitEthernet1/0
D       10.1.23.0/24 [90/3072] via 10.1.12.2, 00:01:32, GigabitEthernet1/0
D       10.1.33.0/24 [90/131072] via 10.1.12.2, 00:01:32, GigabitEthernet1/0
```

Exam Preparation Tasks

As mentioned in the section “How to Use This Book” in the Introduction, you have a couple of choices for exam preparation: the exercises here; Chapter 22, “Final Preparation;” and the exam simulation questions on the CD-ROM.

Review All Key Topics

Review the most important topics in this chapter, noted with the Key Topic icon in the outer margin of the page. Table 11-2 lists a reference of these key topics and the page numbers on which each is found.



Table 11-2 *Key Topics for Chapter 11*

Key Topic Element	Description	Page Number
Step list	Outlines the IPv4 ACL order of operations	401
Paragraph	Identifies how to read an IPv4 standard ACL	401
Paragraph	Identifies how to read an IPv4 extended ACL	402
Example 11-3	Verifying access lists applied to interfaces	403
Example 11-4	Sample time-based ACL	404
Step list	Outlines the IPv6 ACL order of operations	408
Example 11-12	Sample IPv6 ACL	409
Example 11-13	Verifying IPv6 access lists applied to interfaces	409
Paragraphs	Reviews how to read a prefix list	415
Step list	Outlines the prefix list order of operations	416
Paragraphs	Displays how to create an explicit permit all prefix list entry	416

Define Key Terms

Define the following key terms from this chapter and check your answers in the glossary:

standard ACL, extended ACL, named ACL, time-based ACL, IPv6 ACL, implicit deny, implicit permit, prefix list, ge, le

Command Reference to Check Your Memory

This section includes the most important **show** commands covered in this chapter. It might not be necessary to memorize the complete syntax of every command, but you should be able to remember the basic keywords that are needed.

To test your memory of the commands, cover the right side of Table 11-3 with a piece of paper, read the description on the left side, and then see how much of the command you can remember.

The 300-135 TSHOOT exam focuses on practical, hands-on skills that are used by a networking professional. Therefore, you should be able to identify the show commands needed to verify and troubleshoot topics presented in this chapter.

Table 11-3 show Commands

Task	Command Syntax
Displays all the access lists configured on the device	show access-lists
Displays all the IPv4 access lists configured on the device	show ip access-lists
Displays all the IPv6 access lists configured on the device	show ipv6 access-list
Displays the inbound and outbound IPv4 access lists applied to an interface	show ip interface <i>interface_type interface_number</i>
Displays the inbound and outbound IPv6 access lists applied to an interface	show ipv6 interface <i>interface_type interface_number</i>
Displays any time ranges that have been configured on the device	show time-range
Displays the date and time on the device	show clock
Displays the IPv4 prefix lists configured on the device	show ip prefix-list
Displays the IPv6 prefix lists configured on the device	show ipv6 prefix-list
Displays the IPv4 routing protocols running on the router/multilayer switch and can identify whether there are any filters (such as prefix lists) applied inbound or outbound	show ip protocols

This page intentionally left blank



This chapter covers the following topics:

- **Packet-Forwarding Process:** This section covers the processes that are involved during the packet-forwarding process and the commands that you can use while troubleshooting issues related to the packet-forwarding process.
- **Troubleshooting Routing Information Sources:** This section covers how a router can learn from multiple sources and how it chooses which source is the most reliable. You will also learn how you can identify the administrative distance associated with a route.
- **Troubleshooting Static Routes:** This section explains how IPv4 and IPv6 static routes are created. You will also learn the key characteristics of each and how a misconfiguration can cause suboptimal routing or routing loops.
- **Static Routing Trouble Tickets:** This section provides trouble tickets that demonstrate how you can use a structured troubleshooting process to solve a reported problem.
- **Troubleshooting GRE Tunnels:** This section explains how to configure a GRE tunnel over an IPv4 network to transport IPv4 and IPv6 traffic so that you are able to troubleshoot misconfiguration issues with GRE tunnels. You will also discover the benefits of using IPsec with GRE tunnels for security.

Troubleshooting Basic IPv4/IPv6 Routing and GRE Tunnels

Before you can explore how to troubleshoot static routing or dynamic routing, you need a solid understanding of the packet-delivery process (also known as the routing process). This is the process that a router goes through when a packet arrives at an ingress interface and needs to be packet switched to an egress interface. It does not matter whether the packet is an IPv4 or IPv6 packet. The router will go through the same steps to successfully take a packet from an ingress interface and packet switch it to the egress interface.

As a troubleshooter, you also need to be familiar with how a router populates the routing table with “the best” routes. What classifies those routes as the best? Is an Enhanced Interior Gateway Routing Protocol (EIGRP)-learned route better than a static route? What about an Open Shortest Path First (OSPF)-learned route or a Border Gateway Protocol (BGP)-learned route? How do they compare to the other sources of routing information. When multiple sources provide the same routing information, as a troubleshooter you need to be able to identify why the router made the decision it made.

Static routes are part of every network. However, because they are manually configured, they are prone to human error, which can produce suboptimal routing or routing loops. As a troubleshooter, you need to be able to identify issues related to static routes.

Also, it is common to take one protocol like IPv6 and transport it over another protocol like IPv4. This is accomplished by using a tunneling protocol such as generic routing encapsulation (GRE). GRE is used to encapsulate various types of network layer packets inside a transport protocol (GRE) so that they can be transported over an IP network. Therefore, you need to be able to troubleshoot issues related to GRE tunnels.

This chapter covers the packet-delivery process and the various commands that you can use to troubleshoot issues related to the process. You will learn how a router chooses which sources of routing information are more believable so that only the best routes are in the routing table. You will also learn how to recognize and troubleshoot issues related to static routing and GRE tunnels.

“Do I Know This Already?” Quiz

The “Do I Know This Already?” quiz allows you to assess whether you should read this entire chapter thoroughly or jump to the “Exam Preparation Tasks” section. If you are in doubt about your answers to these questions or your own assessment of your knowledge of the topics, read the entire chapter. Table 12-1 lists the major headings in this chapter and their corresponding “Do I Know This Already?” quiz questions. You can find the answers in Appendix A, “Answers to the ‘Do I Know This Already?’ Quizzes.”

Table 12-1 “Do I Know This Already?” Section-to-Question Mapping

Foundation Topics Section	Questions
Packet-Forwarding Process	1–3
Troubleshooting Routing Information Sources	4–8
Troubleshooting Static Routes	9–10
Troubleshooting GRE Tunnels	11–13

Caution The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you do not know the answer to a question or are only partially sure of the answer, you should mark that question as wrong for purposes of the self-assessment. Giving yourself credit for an answer that you correctly guess skews your self-assessment results and might provide you with a false sense of security.

1. Which two data structures does a router use during the packet-forwarding process?
 - a. Routing table
 - b. Layer 3 to Layer 2 mapping table
 - c. Topology table
 - d. Link-state database
2. Which two data structures reside at the routers data plane?
 - a. IP routing table
 - b. ARP cache
 - c. Forwarding Information Base
 - d. Adjacency table
3. Which command enables you to verify routes in the FIB?
 - a. `show ip route`
 - b. `show ip arp`
 - c. `show ip cef`
 - d. `show adjacency detail`
4. Which are capable of populating a routing protocols data structure, such as the EIGRP topology table? (Choose three answers.)
 - a. Updates from a neighbor
 - b. Redistributed routes
 - c. Interfaces enabled for the routing process
 - d. Static routes

5. Which of the following has the lowest default administrative distance?
 - a. OSPF
 - b. EIGRP (internal)
 - c. RIP
 - d. eBGP
6. What is the default administrative distance of an OSPF intra-area route?
 - a. 90
 - b. 110
 - c. 115
 - d. 120
7. What is the default administrative distance of a static route?
 - a. 0
 - b. 1
 - c. 5
 - d. 20
8. How can you create a floating static route?
 - a. Provide the static route with a metric higher than the preferred source of the route
 - b. Provide the static route with a metric lower than the preferred source of the route
 - c. Provide the static route with an AD higher than the preferred source of the route
 - d. Provide the static route with an AD lower than the preferred source of the route
9. What occurs when you create an IPv4 static route with an Ethernet interface designated instead of a next hop IP address?
 - a. The router ARPs for the MAC address of the directly connected routers IP address.
 - b. The router forwards the packet with a destination MAC address of FFFF:FFFF:FFFF.
 - c. The router ARPs for the MAC address of the IP address in the source of the packet.
 - d. The router ARPs for the MAC address of the IP address in the destination of the packet.

- 10.** What occurs when you create an IPv6 static route with a global unicast address as the next hop?
- a.** The router ARPs for the MAC address of the directly connected routers IP address.
 - b.** The router forwards the packet with a destination MAC address of FFFF:FFFF:FFFF.
 - c.** The router uses NDP to determine the MAC address associated with the global unicast address.
 - d.** The router uses NDP to determine the MAC address associated with the destination IPv6 address in the packet.
- 11.** Which statements are true about GRE tunnels?
- a.** The original packet is encapsulated in a GRE header only.
 - b.** The original packet is encapsulated in a GRE header, and then a new IP header is added.
 - c.** The original packet is encapsulated in an IP header, and then a GRE header is added.
 - d.** The original packet header is rewritten by the GRE header.
- 12.** When creating a virtual tunnel interface, what is the default tunnel mode?
- a.** GRE/IP
 - b.** GRE/IPv6
 - c.** IPv6/IP
 - d.** GRE/Multipoint
- 13.** Which of the following are true about IPsec transport mode? (Choose two answers.)
- a.** It creates a new tunnel IP packet.
 - b.** It reuses the GRE IP header, which reduces overhead.
 - c.** It encrypts the original IP packet, the GRE header, and the GRE IP header.
 - d.** It encrypts the original IP packet and the GRE header only.

Foundation Topics

Packet-Forwarding Process

When troubleshooting connectivity issues for an IP-based network, the network layer (Layer 3) of the OSI reference model is often an appropriate place to begin your troubleshooting efforts (*divide-and-conquer method*). For example, if you are experiencing connectivity issues between two hosts on a network, you could check Layer 3 by pinging between the hosts. If the pings are successful, you can conclude that the issue resides at upper layers of the OSI reference model (Layers 4–7). However, if the pings fail, you can focus your troubleshooting efforts on Layers 1–3. If you ultimately determine that there is a problem at Layer 3, your efforts may be centered on the packet-forwarding process of a router.

This section discusses the packet-forwarding process and the commands that you can use to verify the entries in the data structures that are used for this process. It also provides you with a collection of Cisco IOS Software commands that could prove to be useful when troubleshooting-related issues.

Reviewing Layer 3 Packet-Forwarding Process

To review basic routing processes, consider Figure 12-1. In this topology, PC1 needs to access HTTP resources on Server1. Notice that PC1 and Server1 are on different networks. So, the question becomes, how does a packet from a source IP address of 192.168.1.2 get routed to a destination IP address of 192.168.3.2?

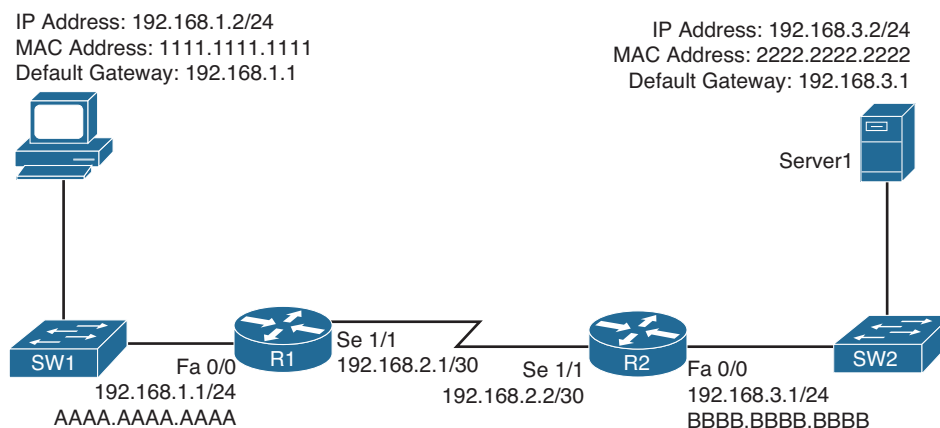


Figure 12-1 Basic Routing Topology

Consider the following walkthrough of this process, step by step:

Step 1. PC1 compares its IP address and subnet mask of 192.168.1.2/24 with the destination IP address 192.168.3.2, as discussed in Chapter 9, “Troubleshooting IPv4 Addressing and Addressing Technologies.” PC1 determines the network portion of its own IP address. It then compares these binary bits with the same binary bits of the destination address. If they are the same, the destination is on the same subnet. If they differ, the destination is on a remote subnet. PC1 concludes that the destination IP address resides on a remote subnet in this example. Therefore, PC1 needs to send the frame to its default gateway, which could have been manually configured on PC1 or dynamically learned via Dynamic Host Configuration Protocol (DHCP). In this example, PC1 has a default gateway of 192.168.1.1 (that is, router R1). To construct a proper Layer 2 frame, PC1 needs the MAC address of the frame’s destination, which is PC1’s default gateway in this example. If the MAC address is not in PC1’s Address Resolution Protocol (ARP) cache, PC1 uses ARP to discover it. Once PC1 receives an ARP reply from router R1, PC1 adds router R1’s MAC address to its ARP cache. PC1 now sends its data destined for Server1 in a frame addressed to R1, as shown in Figure 12-2.

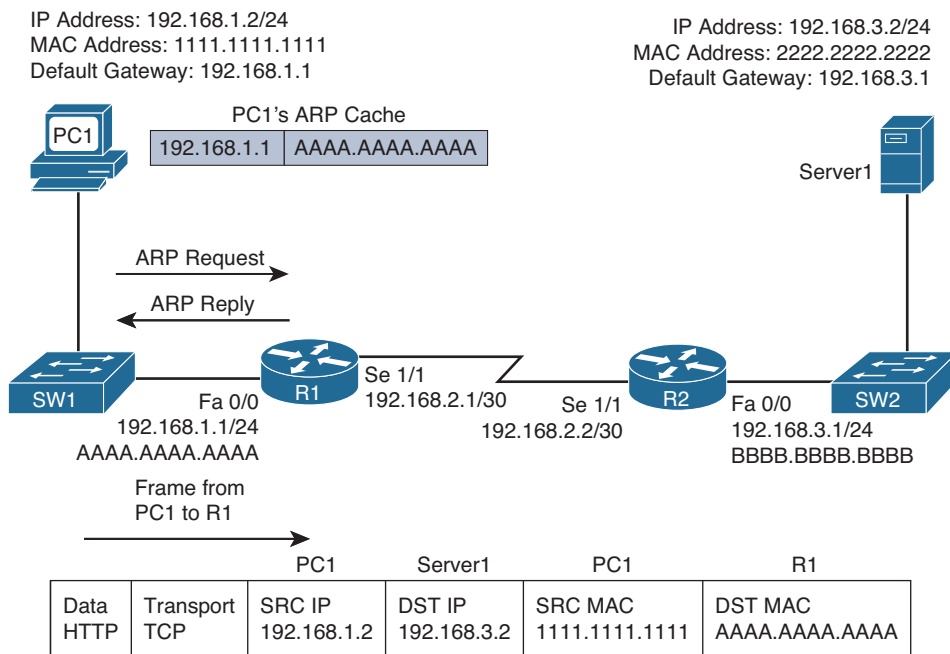


Figure 12-2 Basic Routing Step 1

Step 2. Router R1 receives the frame sent from PC1, and because the destination MAC is R1’s, R1 tears off the Layer 2 header and interrogates the IP (Layer 3) header. An IP header contains a time-to-live (TTL) field, which is decremented once for each router hop. Therefore, router R1 decrements the packet’s TTL

field. If the value in the TTL field is reduced to zero, the router discards the packet and sends a *time-exceeded* Internet Control Message Protocol (ICMP) message back to the source. Assuming the TTL is not decremented to zero, router R1 checks its routing table to determine the best path to reach the IP address 192.168.3.2. In this example, router R1's routing table has an entry stating that network 192.168.3.0/24 is accessible via interface Serial 1/1. Note that ARPs are not required for serial interfaces because these interface types do not have MAC addresses. Therefore, router R1 forwards the frame out of its Serial 1/1 interface, as shown in Figure 12-3, using the PPP Layer 2 framing header.

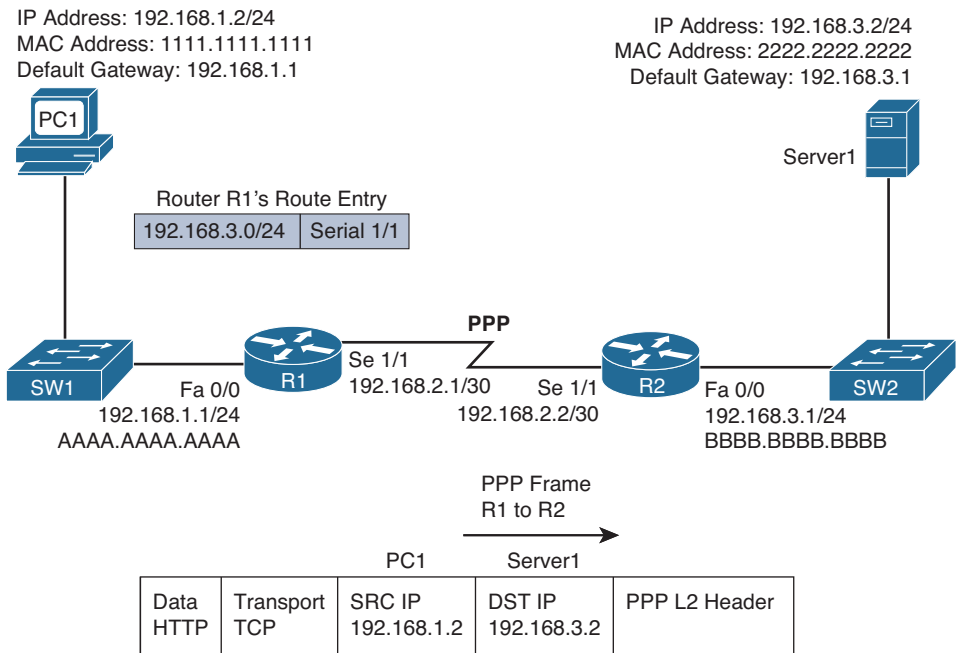


Figure 12-3 Basic Routing Step 2

Step 3. When router R2 receives the frame, it removes the PPP header, and then decrements the TTL in the IP header, just as router R1 did. Again, assuming the TTL did not get decremented to zero, router R2 interrogates the IP header to determine the destination network. In this case, the destination network of 192.168.3.0/24 is directly attached to router R2's Fast Ethernet 0/0 interface. Similar to how PC1 sent out an ARP request to determine the MAC address of its default gateway, router R2 sends an ARP request to determine the MAC address of Server1 if it is not already known in the ARP cache. Once an ARP reply is received from Server1, router R2 forwards the frame out of its Fast Ethernet 0/0 interface to Server1, as shown in Figure 12-4.

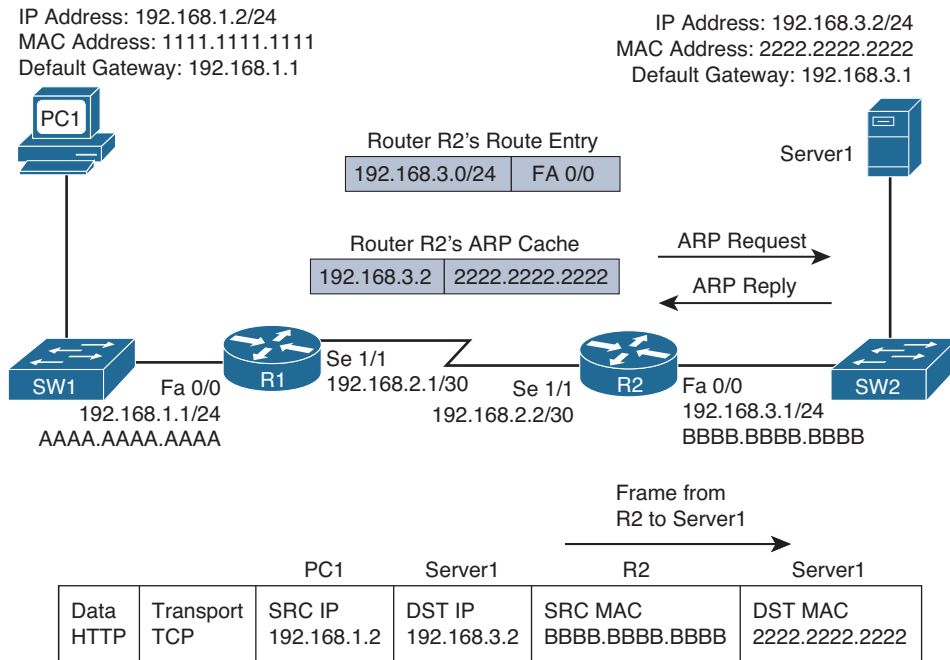


Figure 12-4 Basic Routing Step 3

The previous steps identified two router data structures:



- **IP routing table:** When a router needed to route an IP packet, it consulted its IP routing table to find the best match. The best match is the route that has the *longest prefix*. For example, suppose that a router has a routing entry for network 10.0.0/8, 10.1.1/24, and 10.1.1/26. Also, suppose that the router is trying to forward a packet with the destination IP address 10.1.1.10. The router would select the 10.1.1/26 route entry as the best match for 10.1.1.10 because that route entry has the longest prefix of /26 (matches the most number of bits).
- **Layer 3 to Layer 2 mapping table:** In the previous figure, router R2's ARP cache contained Layer 3 to Layer 2 mapping information. Specifically, the ARP cache had a mapping that said a MAC address of 2222.2222.2222 corresponded to an IP address of 192.168.3.2. An ARP cache is the Layer 3 to Layer 2 mapping data structure used for Ethernet-based networks, but similar data structures are used for Multipoint Frame Relay networks and dynamic multipoint virtual private networks (DMVPNs). However, for point-to-point links such as PPP or HDLC, because there is only one other possible device connected to the other end of the link, no mapping information is needed to determine the next-hop device.

Continually querying a router's routing table and its Layer 3 to Layer 2 mapping data structure (for example, an ARP cache) is less than efficient. Fortunately, Cisco Express Forwarding (CEF), as introduced in Chapter 3, "Troubleshooting Device Performance,"

makes lookups much more efficient. CEF gleans its information from the router's IP routing table and Layer 3 to Layer 2 mapping tables. Then, CEF's data structures can be referenced when forwarding packets. The two primary CEF data structures are as follows:



- **Forwarding Information Base (FIB):** The FIB contains Layer 3 information, similar to the information found in an IP routing table. In addition, an FIB contains information about multicast routes and directly connected hosts.
- **Adjacency table:** When a router is performing a route lookup using CEF, the FIB references an entry in the adjacency table. The adjacency table entry contains the frame header information required by the router to properly form a frame. Therefore, an egress interface and a next-hop MAC address would be in an adjacency entry for a multipoint interface, whereas a point-to-point interface would require only egress interface information.

As a reference, Figure 12-5 shows the router data structures previously discussed.

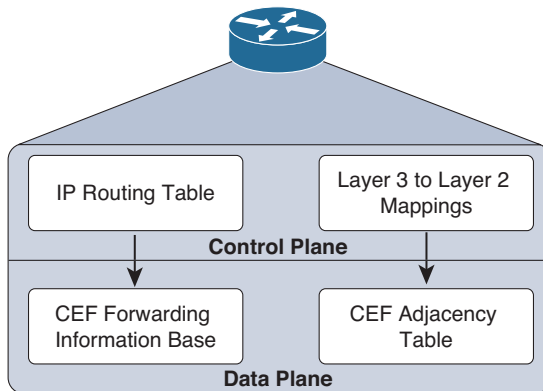


Figure 12-5 A Router's Data Structures

Troubleshooting the Packet-Forwarding Process

When troubleshooting packet-forwarding issues, you will examine a router's IP routing table. If the traffic's observed behavior is not conforming to information in the IP routing table, remember that the IP routing table is maintained by a router's control plane and is used to build the tables at the data plane. CEF is operating in the data plane and uses the FIB. Therefore, you want to view the CEF data structures (that is, the FIB and the adjacency table) that contain all the information required to make packet-forwarding decisions.

Example 12-1 provides sample output from the **show ip route** [*ip_address*] command. The output shows that the next-hop IP address to reach an IP address of 192.168.1.11 is 192.168.0.11, which is accessible via interface Fast Ethernet 0/0. Because this information is coming from the control plane, it includes information about the routing protocol, which is OSPF in this case.

**Example 12-1** *show ip route ip_address Command Output*

```
Router#show ip route 192.168.1.11
Routing entry for 192.168.1.0/24
Known via "ospf 1", distance 110, metric 11, type intra area
Last update from 192.168.0.11 on FastEthernet0/0, 00:06:45 ago
Routing Descriptor Blocks:
192.168.0.11, from 10.1.1.1, 00:06:45 ago, via FastEthernet0/0
Route metric is 11, traffic share count is 1
```

Example 12-2 provides sample output from the **show ip route ip_address subnet_mask** command. The output indicates that the entire network 192.168.1.0/24 is accessible out of interface Fast Ethernet 0/0, with a next-hop IP address of 192.168.0.11.

Example 12-2 *show ip route ip_address subnet_mask Command Output*

```
Router#show ip route 192.168.1.0 255.255.255.0
Routing entry for 192.168.1.0/24
Known via "ospf 1", distance 110, metric 11, type intra area
Last update from 192.168.0.11 on FastEthernet0/0, 00:06:57 ago
Routing Descriptor Blocks:
192.168.0.11, from 10.1.1.1, 00:06:57 ago, via FastEthernet0/0
Route metric is 11, traffic share count is 1
```

Example 12-3 provides sample output from the **show ip route ip_address subnet_mask longer-prefixes** command, with and without the **longer-prefixes** option. Notice that the router responds that the subnet 172.16.0.0 255.255.0.0 is not in the IP routing table. However, after adding the **longer-prefixes** option, two routes are displayed, because these routes are subnets of the 172.16.0.0/16 network.

Example 12-3 *show ip route ip_address subnet_mask longer-prefixes Command Output*

```
Router#show ip route 172.16.0.0 255.255.0.0
% Subnet not in table
R2#show ip route 172.16.0.0 255.255.0.0 longer-prefixes
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
- ODR, P - periodic downloaded static route

Gateway of last resort is not set

172.16.0.0/30 is subnetted, 2 subnets
C      172.16.1.0 is directly connected, Serial1/0.1
C      172.16.2.0 is directly connected, Serial1/0.2
```

Example 12-4 provides sample output from the **show ip cef ip_address** command. The output indicates that, according to CEF, an IP address of 192.168.1.11 is accessible out of interface Fast Ethernet 0/0, with a next-hop IP address of 192.168.0.11.

Key Topic

Example 12-4 show ip cef ip_address Command Output

```
Router#show ip cef 192.168.1.11
192.168.1.0/24, version 42, epoch 0, cached adjacency 192.168.0.11
0 packets, 0 bytes
via 192.168.0.11, FastEthernet0/0, 0 dependencies
next hop 192.168.0.11, FastEthernet0/0
valid cached adjacency
```

Example 12-5 provides sample output from the **show ip cef ip_address subnet_mask** command. The output indicates that network 192.168.1.0/24 is accessible off of interface Fast Ethernet 0/0, with a next-hop IP address of 192.168.0.11.

Example 12-5 show ip cef ip_address subnet_mask Command Output

```
Router#show ip cef 192.168.1.0 255.255.255.0
192.168.1.0/24, version 42, epoch 0, cached adjacency 192.168.0.11
0 packets, 0 bytes
via 192.168.0.11, FastEthernet0/0, 0 dependencies
next hop 192.168.0.11, FastEthernet0/0
valid cached adjacency
```

Example 12-6 provides sample output from the **show ip cef exact-route source_address destination_address** command. The output indicates that a packet sourced from an IP address of 10.2.2.2 and destined for an IP address of 192.168.1.11 will be sent out of interface Fast Ethernet 0/0 to a next-hop IP address of 192.168.0.11.

Example 12-6 show ip cef exact-route source_address destination_address Command Output

```
Router#show ip cef exact-route 10.2.2.2 192.168.1.11
10.2.2.2 -> 192.168.1.11 : FastEthernet0/0 (next hop 192.168.0.11)
```

For a multipoint interface such as point-to-multipoint Frame Relay or Ethernet, after a router knows the next-hop address for a packet, it needs appropriate Layer 2 information (for example, next-hop MAC address, or data-link connection identifier [DLCI]) to properly construct a frame. Example 12-7 provides sample output from the **show ip arp** command, which displays the ARP cache that is stored in the control plane on a router. The output shows the learned or configured MAC addresses along with their associated IP addresses.

Key Topic

Example 12-7 show ip arp Command Output

```
Router#show ip arp
Protocol Address      Age (min)  Hardware Addr  Type   Interface
Internet 192.168.0.11         0    0009.b7fa.d1e1  ARPA   FastEthernet0/0
Internet 192.168.0.22         -    c001.0f70.0000  ARPA   FastEthernet0/0
```

Example 12-8 provides sample output from the **show frame-relay map** command. The output shows the Frame Relay interfaces, the corresponding DLCIs associated with the interfaces, and the next-hop IP address that is reachable out the interface using the permanent virtual circuit (PVC) associated with the listed DLCI. In this case, if R2 needs to send data to the next-hop IP address 172.16.33.6, it would use the PVC associated with DLCI 406 to get there.



Example 12-8 show frame-relay map Command Output

```
Router#show frame-relay map
Serial1/0 (up): ip 172.16.33.5 dlci 405(0x195,0x6450), static,broadcast,
CISCO, status defined, active
Serial1/0 (up): ip 172.16.33.6 dlci 406(0x196,0x6460), static,broadcast,
CISCO, status defined, active
```

Example 12-9 provides sample output from the **show ip nhrp** command. This command displays the Next Hop Resolution Protocol cache that is used with DMVPNs. In this example, if a packet needs to be sent to the 192.168.255.2 next-hop IP address, the non-broadcast multiaccess (NBMA) address of 198.51.100.2 is used to reach it.

Example 12-9 show ip nhrp Command Output

```
HUBRouter#show ip nhrp
192.168.255.2/32 via 192.168.255.2
Tunnel0 created 00:02:35, expire 01:57:25
Type: dynamic, Flags: unique registered
NBMA address: 198.51.100.2
192.168.255.3/32 via 192.168.255.3
Tunnel0 created 00:02:36, expire 01:57:23
Type: dynamic, Flags: unique registered
NBMA address: 203.0.113.2
```

Example 12-10 provides sample output from the **show adjacency detail** command. The output shows the CEF information used to construct frame headers needed to reach the next-hop IP addresses through the various router interfaces. Notice the value 64510800 for Serial 1/0. This is a hexadecimal representation of information that is needed by the router to successfully forward the packet to the next hop IP address 172.16.33.5, including the DLCI of 405. Notice the value CA1B01C4001CCA1C164000540800 for Fast Ethernet 3/0. This is the destination MAC, source MAC, and the EtherType code for an Ethernet frame. The first 12 hex values are the destination MAC, the next 12 are the source MAC, and 0800 is the IPv4 EtherType code.

Example 12-10 show adjacency detail *Command Output*

```

Router#show adjacency detail
Protocol Interface      Address
IP        Serial1/0              172.16.33.5(7)
           0 packets, 0 bytes
           epoch 0
           sourced in sev-epoch 1
           Encap length 4
           64510800
           FR-MAP
IP        Serial1/0      172.16.33.6(7)
           0 packets, 0 bytes
           epoch 0
           sourced in sev-epoch 1
           Encap length 4
           64610800
           FR-MAP
IP        FastEthernet3/0 203.0.113.1(7)
           0 packets, 0 bytes
           epoch 0
           sourced in sev-epoch 1
           Encap length 14
           CA1B01C4001CCA1C164000540800
           L2 destination address byte offset 0
           L2 destination address byte length 6
           Link-type after encap: ip
           ARP

```

Troubleshooting Routing Information Sources

When designing a routed network, you have many options to choose from when determining what will be the source of routing information: connected, static, EIGRP, OSPF, BGP, to name a few. With all these different options, you need to be able to recognize what is more trustworthy (believable). This is extremely important when you are using multiple sources because only one source of information can be used to populate the routing table for any given route. As a result, it is important as a troubleshooter to understand how the best source of route information is determined and placed in the routing table.

This section explains which sources of route information are the most believable and how the routing table interacts with various data structures to populate itself with the best information.

Data Structures and the Routing Table

To better troubleshoot routing information sources, consider, generically, how dynamic routing protocols' data structures interact with a router's IP routing table. Figure 12-6 shows the interaction between the data structures of an IP routing protocol and a router's IP routing table.

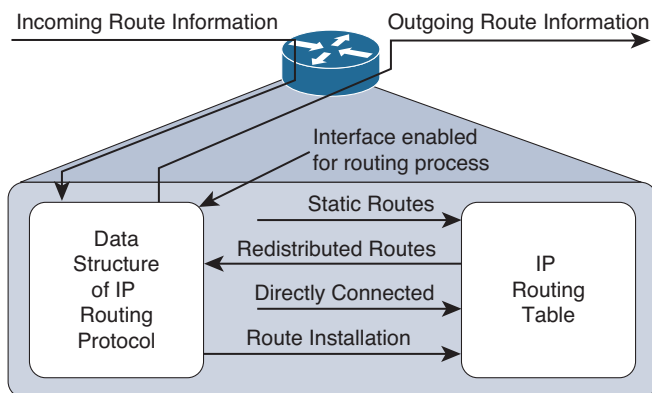


Figure 12-6 *Interaction Between the IP Routing Table and a Routing Protocol Data Structure*

As a router receives route information from a neighboring router, the information is stored in the data structures of the IP routing protocol and analyzed by the routing protocol to determine the best path based on metrics. An IP routing protocol's data structure can also be populated by the local router. For example, a router might be configured for route redistribution, where route information is redistributed from the routing table into the IP routing protocols data structure. The router might be configured to have specific interfaces participate in an IP routing protocol process. Therefore, the network that the interface belongs to is placed into the routing protocol data structure as well.

However, what goes in the routing table? Reviewing Figure 12-6 again, you can see that the routing protocol data structure can populate the routing table, a directly connected route can populate the routing table, and static routes can populate the routing table. These are all known as sources of routing information.

Sources of Route Information

Your router could conceivably receive route information from the following routing sources all at the same time:

- Connected interface
- Static route
- RIP
- EIGRP
- OSPF
- BGP

If the route information received from all these sources is for different destination networks, each one will be used for its respectively learned destination networks and be placed in the routing table. However, what if the route received from RIP and OSPF were the exact same? For example, both protocols have informed the router about the 10.1.1.0/24 network. How does the router choose which is the most believable, or the best source of routing information? It cannot use both; it has to pick one and install that information in the routing table.

Routing information sources have each been assigned an *administrative distance* (AD). An administrative distance of a routing information source can be thought of as the *believability* or *trustworthiness* of that routing source when comparing it to the other routing information sources. Table 12-2 lists the default ADs of routing information sources. The lower the AD, the more preferred the source of information.

For instance, RIP has a default AD of 120, whereas OSPF has a default AD of 110. Therefore, if both RIP and OSPF have knowledge of a route to a specific network (10.1.1.0/24 as an example), the OSPF route would be injected into the router's IP routing table because OSPF has a more believable AD. Therefore, the best route selected by an IP routing protocol's data structure is only a *candidate* to be injected into the router's IP routing table. The route is only injected into the routing table if the routing table concludes that it came from the best routing source. As you will see in later chapters when you troubleshoot specific routing protocols, routes might be missing in the routing table from a specific routing protocol, or suboptimal routing may be occurring because a different routing source with a lower AD is being used.



Table 12-2 *Default Administrative Distance of Route Sources*

Source of Route Information	AD
Connected interface	0
Static route	1
EIGRP summary route	5
eBGP	20
EIGRP (internal)	90
OSPF	110
IS-IS	115
RIP	120
EGP	140
ODR	160
EIGRP (external)	170
iBGP	200
Unknown (not believable)	255

You can verify the AD of a route in the routing table by using the **show ip route ip_address** command as shown in Example 12-11. You can see in the example that the route to 10.1.1.0 has an AD of 0 and the route to 10.1.23.0 has an AD of 90.



Example 12-11 Verifying the Administrative Distance of a Route in the Routing Table

```
R1#show ip route 10.1.1.0
Routing entry for 10.1.1.0/26
Known via "connected", distance 0, metric 0 (connected, via interface)
Redistributing via eigrp 100
Routing Descriptor Blocks:
  directly connected, via GigabitEthernet1/0
Route metric is 0, traffic share count is 1

R1#show ip route 10.1.23.0
Routing entry for 10.1.23.0/24
Known via "eigrp 100", distance 90, metric 3072, type internal
Redistributing via eigrp 100
Last update from 10.1.13.3 on GigabitEthernet2/0, 09:42:20 ago
Routing Descriptor Blocks:
  10.1.13.3, from 10.1.13.3, 09:42:20 ago, via GigabitEthernet2/0
Route metric is 3072, traffic share count is 1
Total delay is 20 microseconds, minimum bandwidth is 1000000 Kbit
Reliability 255/255, minimum MTU 1500 bytes
Loading 1/255, Hops 1
```

If you ever need to make sure that the route information or subset of route information received from a particular source is never used, you can change the AD of specific routes or all routes from that source to 255, which means “do not believe.”

AD can also be used to manipulate path selection. For example you may have two different paths to the same destination learned from two different sources (for example, EIGRP and a static route). In this case, the static route is preferred. However, this static route may be pointing to a backup link that is slower than the EIGRP path. Therefore, you want the EIGRP path to be installed in the routing table because the static route is causing sub-optimal routing. But, you are not allowed to remove the static route. To solve this issue, you can create a floating static route. This static route has a higher AD than the preferred route. Because we want EIGRP to be preferred, we modify the static route so that it has an AD higher than EIGRP, which is 90. As a result, the EIGRP-learned route is installed in the routing table, and the static route will be installed only if the EIGRP-learned route goes away.

Troubleshooting Static Routes

Static routes are manually configured by administrators, and by default are the second most trustworthy source of routing information, with an AD of 1. They allow an administrator to precisely control how to route packets for a particular destination. This section

discusses the syntax of IPv4 and IPv6 static routes and explains what to look for while troubleshooting.

IPv4 Static Routes

To create an IPv4 static route, you use the `ip route prefix mask {ip_address | interface_type interface_number} [distance]` command in global configuration mode. Example 12-12 displays the configuration of a static route on R1, as shown in Figure 12-7. The static route is training R1 about the 10.1.3.0/24 network. To get to the network, it is reachable via the next-hop address of 10.1.12.2, which is R2, and it has been assigned an AD of 8. (The default is 1.)

Example 12-12 Configuring a Static Route on R1 with Next-Hop Option

```
R1#config t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#ip route 10.1.3.0 255.255.255.0 10.1.12.2 8
```

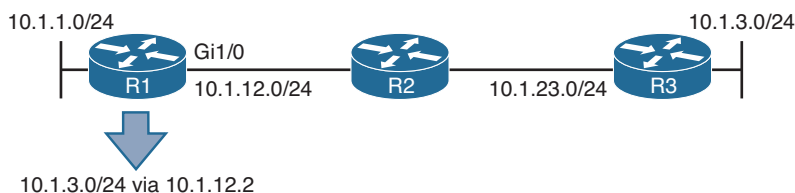


Figure 12-7 Configuring a Static Route on R1 with Next-Hop Option

Example 12-13, which shows the output of `show ip route static` on R1, indicates that the 10.1.3.0/24 network was learned by a static route, it is reachable via the next-hop IP address of 10.1.12.2, it has an AD of 8, and the metric is 0 because there is no way to know how far the destination truly is like a dynamic routing protocol does.

Example 12-13 Verifying a Static Route on R1

```
R1#show ip route static
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
...output omitted...

10.0.0.0/8 is variably subnetted, 7 subnets, 2 masks
S    10.1.3.0/24 [8/0] via 10.1.12.2
```

When troubleshooting IPv4 static routes, you need to be able to recognize why the static route may not be providing the results you want. For example, are the network and mask accurate? If either of these is incorrect, your static route will not route the packets you are expecting it to route. The router might drop packets because it does not match the static route or any other route. It might end up forwarding packets via the default route, which may be pointing the wrong way. In addition, if the static route includes networks that it should not, you could be routing packets the wrong way.



Consider this: If you were to configure the following static route on R2 in Figure 12-7, **ip route 10.1.3.0 255.255.255.0 10.1.12.1**, packets destined to 10.1.3.0 would be sent to R1, which is the wrong way. However, notice in Example 12-13 that R1 points to R2 (10.1.12.2) for the network 10.1.3.0/24. Therefore, R1 and R2 will simply bounce packets back and forth that are destined for 10.1.3.0/24 until the TTL expires.

As you can see, the next-hop IP address is a very important parameter for the static route. It tells the local router where to send the packet. For instance, in Example 12-13, the next hop is 10.1.12.2. Therefore, a packet destined to 10.1.3.0 has to go to 10.1.12.2 next. R1 now does a recursive lookup in the routing table for 10.1.12.2 to determine how to reach it, as shown in Example 12-14. This example displays the output of the **show ip route 10.1.12.2** command on R1. You can see that 10.1.12.2 is directly connected out Gigabit Ethernet 1/0.

Example 12-14 Recursive Lookup on R1 for Next-Hop Address

```
R1#show ip route 10.1.12.2
Routing entry for 10.1.12.0/24
Known via "connected", distance 0, metric 0 (connected, via interface)
Routing Descriptor Blocks:
  directly connected, via GigabitEthernet1/0
Route metric is 0, traffic share count is 1
```

Because the exit interface to reach 10.1.12.2 is Gigabit Ethernet 1/0, the Ethernet frame requires a source and destination MAC address. As a result, R1 looks in its ARP cache as shown in Example 12-15 and finds the MAC for 10.1.12.2 is ca08.0568.0008.

Example 12-15 MAC Address Lookup in ARP Cache

Protocol	Address	Age (min)	Hardware Addr	Type	Interface
Internet	10.1.1.1	-	ca07.0568.0008	ARPA	GigabitEthernet0/0
Internet	10.1.12.1	-	ca07.0568.001c	ARPA	GigabitEthernet1/0
Internet	10.1.12.2	71	ca08.0568.0008	ARPA	GigabitEthernet1/0

As you can see in this case, the MAC address of the next-hop address is used for the Layer 2 frame. It is not the MAC address of the IP address in the packet. The benefit of this is that the router only has to find the MAC address of the next-hop once using the ARP process and then store the results in the ARP cache. Now, any packet that has to go to the next-hop of 10.1.12.2 does not require an ARP request to be sent, just a look up in the ARP cache, making the overall routing process more efficient.

Now that we understand the next-hop IP address, there is another option we need to know about. The **ip route** syntax shown earlier indicated that you can specify an exit interface instead of a next hop IP address. There is a right time and a wrong time to use the exit interface. The right time is when it's a pure point-to-point interface such as DSL, or Serial. Point-to-Point Ethernet links are not pure point-to-point, they are still multi-access, and since they are Ethernet they require a source and destination MAC address. If

you specify an Ethernet interface as the next-hop you will be making your router ARP for the MAC address of every destination IP address in every packet. Let's look at this.

You configure the following static route on R1: **ip route 10.1.3.0 255.255.255.0 gigabit Ethernet 1/0**. Example 12-16 displays how the static route appears in the routing table. It states 10.1.3.0/24 is directly connected to Gigabit Ethernet 1/0. But is it? Refer to Figure 12-8 to know for sure. It is clear in Figure 12-8 that 10.1.3.0/24 is not directly connected. But, the way the static route is configured, R1 thinks that it is.

Example 12-16 Static Route with Exit Interface Specified

```
R1#show ip route static
...output omitted...

10.0.0.0/8 is variably subnetted, 7 subnets, 2 masks
S      10.1.3.0/24 is directly connected, GigabitEthernet1/0
```

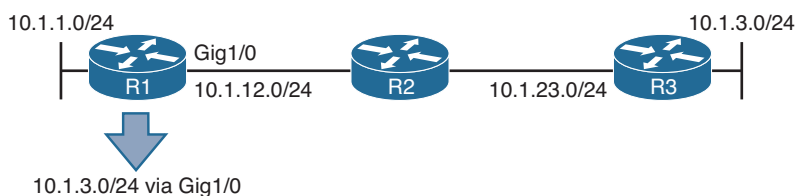


Figure 12-8 Configuring a Static Route on R1 with Exit Interface Option

Imagine users in the 10.1.1.0/24 network are trying to access resources in the 10.1.3.0/24 network. Specifically, they are accessing resources on 10.1.3.1 through 10.1.3.8. R1 receives the packets, and it looks in the routing table and the longest match is the following entry:

```
S      10.1.3.0/24 is directly connected, GigabitEthernet1/0
```

**Key
Topic**

R1 believes the network is directly connected; therefore, the destination IP address in the packet is on the network connected to Gig1/0. However, we know better because we have shown in Figure 12-8 that it is not. So, because it is an Ethernet interface, R1 will use ARP to determine the MAC of the IP address in the destination field of the packet. (This is different from what occurred when the next-hop IP address was specified. When the next-hop was specified, the MAC of the next-hop address was used.) Review Example 12-17 now. It displays the ARP cache on R1. Notice that every destination IP address has an entry in the ARP cache. How can that be since ARPs are not forwarded by routers? It is because of Proxy ARP, which is on by default on our routers. Proxy ARP allows a router to respond to ARP requests with its own MAC address if it has a route in the routing table to the IP address in the ARP request. Notice how the MAC addresses listed are all the same. In addition, they match the MAC address of the 10.1.12.2 entry. Therefore, because R2 has a route to reach the IP address of the ARP request, it responds back with its MAC address to use.

Example 12-17 *ARP Cache on R1 with R2 Proxy ARP Enabled*

```
R1#show ip arp
```

Protocol	Address	Age (min)	Hardware Addr	Type	Interface
Internet	10.1.1.1	-	ca07.0568.0008	ARPA	GigabitEthernet0/0
Internet	10.1.3.1	0	ca08.0568.0008	ARPA	GigabitEthernet1/0
Internet	10.1.3.2	0	ca08.0568.0008	ARPA	GigabitEthernet1/0
Internet	10.1.3.3	3	ca08.0568.0008	ARPA	GigabitEthernet1/0
Internet	10.1.3.4	0	ca08.0568.0008	ARPA	GigabitEthernet1/0
Internet	10.1.3.5	1	ca08.0568.0008	ARPA	GigabitEthernet1/0
Internet	10.1.3.6	0	ca08.0568.0008	ARPA	GigabitEthernet1/0
Internet	10.1.3.7	0	ca08.0568.0008	ARPA	GigabitEthernet1/0
Internet	10.1.3.8	1	ca08.0568.0008	ARPA	GigabitEthernet1/0
Internet	10.1.12.1	-	ca07.0568.001c	ARPA	GigabitEthernet1/0
Internet	10.1.12.2	139	ca08.0568.0008	ARPA	GigabitEthernet1/0

Example 12-18 shows how you can verify whether Proxy ARP is enabled by using the `show ip interfaces` command.

Example 12-18 *Verifying Whether Proxy ARP Is Enabled*

```
R2#show ip interface gigabitEthernet 0/0
GigabitEthernet0/0 is up, line protocol is up
Internet address is 10.1.12.2/24
Broadcast address is 255.255.255.255
Address determined by non-volatile memory
MTU is 1500 bytes
Helper address is not set
Directed broadcast forwarding is disabled
Multicast reserved groups joined: 224.0.0.5 224.0.0.6
Outgoing access list is not set
Inbound access list is not set
Proxy ARP is enabled
Local Proxy ARP is disabled
Security level is default
Split horizon is enabled
ICMP redirects are always sent
```

If Proxy ARP was not enabled, the ARP cache on R1 would appear as shown in Example 12-19. Notice how R1 is still sending ARP requests; however, it is not getting any ARP replies. Therefore, it cannot build the Layer 2 frame, and the result is an *encapsulation failure*, which you would be able to see if you were debugging IP packets.

Example 12-19 *ARP Cache on R1 with R2 Proxy ARP Disabled*

```
R1#show ip arp
```

Protocol	Address	Age (min)	Hardware Addr	Type	Interface
Internet	10.1.1.1	-	ca07.0568.0008	ARPA	GigabitEthernet0/0
Internet	10.1.3.1	0	Incomplete	ARPA	

Internet	10.1.3.2	0	Incomplete	ARPA	
Internet	10.1.3.3	0	Incomplete	ARPA	
Internet	10.1.3.4	0	Incomplete	ARPA	
Internet	10.1.3.5	0	Incomplete	ARPA	
Internet	10.1.3.6	0	Incomplete	ARPA	
Internet	10.1.3.7	0	Incomplete	ARPA	
Internet	10.1.3.8	0	Incomplete	ARPA	
Internet	10.1.12.1	-	ca07.0568.001c	ARPA	GigabitEthernet1/0
Internet	10.1.12.2	139	ca08.0568.0008	ARPA	GigabitEthernet1/0

Because of the fact that R1 will use ARP to determine the MAC address of every destination IP address in every packet, you should never specify an Ethernet interface in a static route. This results in an excessive use of router resources, such as processor and memory, as the control plane gets involved during the forwarding process to determine the appropriate Layer 2 MAC address using ARP.

As you can see, being able to recognize misconfigured static routes and the issues that arise is an important skill to have when troubleshooting because a misconfigured static route will cause traffic to be misrouted or suboptimally routed. In addition, remember that static routes have an AD of 1; therefore, they will be preferred over other sources of routing information to the same destination.

IPv6 Static Routes

To create an IPv6 static route, you use the **ipv6 route** *{ipv6_prefix/prefix_length} {ipv6_address | interface_type interface_number} [administrative_distance] [next_hop_address]* command in global configuration mode.

Example 12-20 displays the configuration of a static route on R1, as shown in Figure 12-9. The static route is training R1 about the 2001:DB8:0:3::/64 network. To get to the network, it is reachable via the next-hop address of FE80::2, which is R2's link-local address, and it has been assigned an AD of 8. (The default is 1.) Notice how the exit Ethernet interface is specified. This is mandatory when using the link-local address as the next-hop because the same link-local address can be used on multiple local router interfaces. In addition, multiple remote router interfaces can have the same link-local address as well. However, as long as the link-local addresses are unique between the devices within the same local network, then communication occurs as intended. If you are using a global unicast address as the next hop, you do not have to specify the exit interface.

Example 12-20 Configuring an IPv6 Static Route on R1 with Next-Hop Option

```
R1#config t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#ipv6 route 2001:DB8:0:3::/64 gigabitEthernet 1/0 FE80::2 8
```

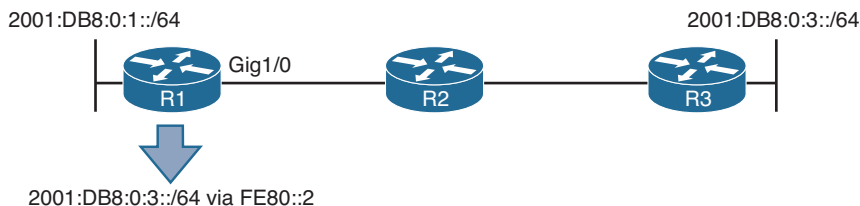



Figure 12-9 Configuring an IPv6 Static Route on R1 with Next-Hop Option

Example 12-21, which shows the output of `show ipv6 route static` on R1, indicates that the 2001:DB8:0:3::/64 network was learned by a static route, it is reachable via the next-hop IP address of FE80::2, it has an AD of 8, and the metric is 0 because there is no way to know how far the destination truly is like a dynamic routing protocol does.

Example 12-21 Verifying an IPv6 Static Route on R1

```
R1#show ipv6 route static
...output omitted...
S   2001:DB8:0:3::/64 [8/0]
    via FE80::2, GigabitEthernet1/0
```

Now recall that there are no broadcasts with IPv6. Therefore, IPv6 does not use ARP. It uses NDP (Neighbor Discovery Protocol), which is multicast based, to determine a neighboring device's MAC address. In this case, if R1 needs to route packets to 2001:DB8:0:3::/64, the routing table says to use the next-hop FE80::2, which is out Gig1/0. Therefore, it consults its IPv6 neighbor table, as shown in Example 12-22, to determine whether there is a MAC address for FE80::2 out Gig 1/0. It is imperative that the table has an entry mapping the *link-local address* and the *interface*. If only one matches, it is not the correct entry. If there is no entry in the IPv6 neighbor table, a neighbor solicitation message is sent to discover the MAC of FE80::2 on Gig1/0.

Example 12-22 Viewing the IPv6 Neighbor Table on R1

```
R1#show ipv6 neighbors
IPv6 Address                               Age Link-layer Addr State Interface
FE80::2                                   0 ca08.0568.0008 REACH Gil/0
```

As you discovered earlier with IPv4, it is not acceptable to use the interface option in a static route when the interface is an Ethernet interface because of Proxy ARP consuming an excessive amount of router resources. Note that Proxy ARP does not exist in IPv6. Therefore, if you use the interface option with an Ethernet interface, it will work only if the destination IPv6 address is directly attached to the router interface specified. This is because the destination IPv6 address in the packet is used as the next-hop address and the MAC address will need to be discovered using NDP. If the destination is not in the directly connected network, ND will fail, and Layer 2 encapsulation will ultimately fail. Consider Figure 12-9 again. On R1, if you configured the following IPv6 static route (which is called a directly attached static route), what would happen?

```
ipv6 route 2001:DB8:0:3::/64 gigabitEthernet 1/0
```



When R1 receives a packet destined for 2001:db8:0:3::3, it determines based on the static route that it is directly connected to Gig1/0 (which it is not according to the figure). Therefore, R1 sends an NS out Gig1/0 for the MAC address associated with 2001:db8:0:3::3 using the solicited-node multicast address FF02::1:FF00:3. If no device attached to Gig1/0 is using the solicited-node multicast address FF02::1:FF00:3 and the IPv6 address 2001:db8:0:3::3, the NS goes unanswered, and Layer 2 encapsulation fails.

As you can see, being able to recognize misconfigured static routes and the issues that arise is an important skill to have when troubleshooting because a misconfigured static route will cause traffic to be misrouted or suboptimally routed. In addition, remember that static routes have an AD of 1 by default; therefore, they are preferred over other sources of routing information to the same destination.

Static Routing Trouble Tickets

This section presents various trouble tickets relating to the topics discussed earlier in the chapter. The purpose of these trouble tickets is to give a process that you can follow when troubleshooting in the real world or in an exam environment. All trouble tickets in this section are based on the topology depicted in Figure 12-10.

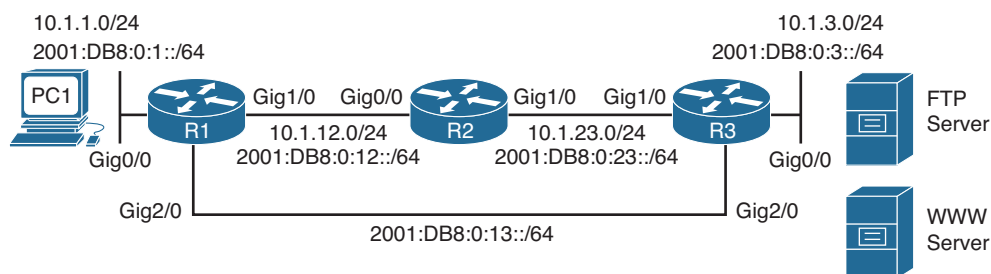


Figure 12-10 *Static Routing Trouble Tickets Topology*

Trouble Ticket 12-1

Problem: Users in the 10.1.1.0/24 network have indicated that they are not able to access resources on the FTP server in the 10.1.3.0/24 network. The FTP server uses a static IPv4 address of 10.1.3.10. They also indicate that they are able to access the web server at 10.1.3.5. (Note: This network only uses static routes.)

You start your troubleshooting efforts by verifying the problem with a ping to 10.1.3.10 from PC1 in the 10.1.1.0/24 network. As shown in Example 12-23 the ping is not successful. R1 is responding with a destination unreachable message. This indicates that R1 does not know how to route the packet destined for 10.1.3.10. In addition, you ping 10.1.3.5 from PC1, and it is successful, as shown in Example 12-23 as well.

Example 12-23 *Failed Ping From PC1 to 10.1.3.10 and Successful Ping to 10.1.3.5*

```

C:\PC1>ping 10.1.3.10

Pinging 10.1.3.10 with 32 bytes of data:

Reply from 10.1.1.1: Destination host unreachable.
Reply from 10.1.1.1: Destination host unreachable.
Reply from 10.1.1.1: Destination host unreachable.
Reply from 10.1.1.1: Destination host unreachable.

Ping statistics for 10.1.3.10:
Packets: Sent = 4, Received = 4, lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\PC1>ping 10.1.3.5

Pinging 10.1.3.5 with 32 bytes of data:

Reply from 10.1.3.5: bytes=32 time 1ms TTL=128
Reply from 10.1.3.5: bytes=32 time 1ms TTL=128
Reply from 10.1.3.5: bytes=32 time 1ms TTL=128
Reply from 10.1.3.5: bytes=32 time 1ms TTL=128

Ping statistics for 10.1.3.5:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms

```

Next you access R1 and issue the **show ip route** command on R1 to verify whether it knows how to route the packet to 10.1.3.10. In Example 12-24, the closest entry that would match 10.1.3.10 is the entry for 10.1.3.0/29. However, does 10.1.3.10 fall within that subnet?

Example 12-24 *Verifying Routing Table Entries*

```

R1#show ip route
...output omitted...

Gateway of last resort is not set

10.0.0.0/8 is variably subnetted, 6 subnets, 3 masks
C       10.1.1.0/24 is directly connected, GigabitEthernet0/0
L       10.1.1.1/32 is directly connected, GigabitEthernet0/0
S       10.1.3.0/29 [1/0] via 10.1.12.2
C       10.1.12.0/24 is directly connected, GigabitEthernet1/0
L       10.1.12.1/32 is directly connected, GigabitEthernet1/0
S       10.1.23.0/24 [1/0] via 10.1.12.2

```

The network 10.1.3.0/29 would have a range of addresses from 10.1.3.0 to 10.1.3.7. Therefore, 10.1.3.10 does not fall within that subnet, but 10.1.3.5 does. This explains why the users can reach one address and not the other in the 10.1.3.0/24 network. If you execute the **show ip route 10.1.3.10** and **show ip route 10.1.3.5** commands on R1, the output will verify this further. As shown in Example 12-25, there is no match for 10.1.3.10, but there is one for 10.1.3.5.

Example 12-25 *Verifying Specific Routes*

```
R1#show ip route 10.1.3.10
% Subnet not in table
R1#show ip route 10.1.3.5
Routing entry for 10.1.3.0/29
Known via "static", distance 1, metric 0
Routing Descriptor Blocks:
 10.1.12.2
Route metric is 0, traffic share count is 1
```

Because the network in Figure 12-10 is 10.1.3.0/24, and the entry in the routing table is 10.1.3.0/29, it is possible that the static route was misconfigured. You need to verify this by examining the running configuration using the **show run | include ip route** command, as shown in Example 12-26. Notice the command **ip route 10.1.3.0 255.255.255.248 10.1.12.2**. This is the command that is producing the 10.1.3.0/29 entry in the routing table. If you look closely, you will notice that the subnet mask was not configured correctly.

Example 12-26 *Examining the Static Routes on R1 in the Running Configuration*

```
R1#show run | include ip route
ip route 10.1.3.0 255.255.255.248 10.1.12.2
ip route 10.1.23.0 255.255.255.0 10.1.12.2
```

To solve this issue, you need to remove the static route with the command **no ip route 10.1.3.0 255.255.255.248 10.1.12.2** and create a new static route with the **ip route 10.1.3.0 255.255.255.0 10.1.12.2** command.

After you do this, you issue the **show ip route** command on R1 and confirm that the entry in the routing table is 10.1.3.0/24, as shown in Example 12-27.

Example 12-27 *Verifying Updated Static Route in the Routing Table on R1*

```
R1#show ip route
...output omitted...

Gateway of last resort is not set

10.0.0.0/8 is variably subnetted, 6 subnets, 2 masks
C       10.1.1.0/24 is directly connected, GigabitEthernet0/0
L       10.1.1.1/32 is directly connected, GigabitEthernet0/0
```

```

S      10.1.3.0/24 [1/0] via 10.1.12.2
C      10.1.12.0/24 is directly connected, GigabitEthernet1/0
L      10.1.12.1/32 is directly connected, GigabitEthernet1/0
S      10.1.23.0/24 [1/0] via 10.1.12.2

```

Next you issue the **show ip route 10.1.3.10** command, as shown in Example 12-28, and notice that the IP 10.1.3.10 now matches an entry in the routing table.

Example 12-28 *Verifying an Entry Exists for 10.1.3.10*

```

R1#show ip route 10.1.3.10
Routing entry for 10.1.3.0/24
Known via "static", distance 1, metric 0
Routing Descriptor Blocks:
 10.1.12.2
Route metric is 0, traffic share count is 1

```

Finally, you ping from PC1 to the IP address 10.1.3.10, and the ping is successful, as shown in Example 12-29.

Example 12-29 *Successful Ping from PC1 to 10.1.3.10*

```

C:\PC1>ping 10.1.3.10

Pinging 10.1.3.10 with 32 bytes of data:

Reply from 10.1.3.10: bytes=32 time 1ms TTL=128
Reply from 10.1.3.10: bytes=32 time 1ms TTL=128
Reply from 10.1.3.10: bytes=32 time 1ms TTL=128
Reply from 10.1.3.10: bytes=32 time 1ms TTL=128

Ping statistics for 10.1.3.10:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms

```

Trouble Ticket 12-2

Problem: Your proactive traffic monitoring indicates that all traffic from 2001:DB8:0:1::/64 destined to 2001:DB8:0:3::/64 is going through R2 when it should be going directly to R3 over the Gig2/0 link. R2 should only be used to forward traffic from 2001:DB8:0:1::/64 to 2001:DB8:0:3::/64 if the Gig2/0 link fails, which it has not. You need to determine why traffic is being forwarded the wrong way and fix it. (Note: This network only uses static routes.)

You start by confirming the problem with a trace, as shown in Example 12-30, from PC1 to 2001:DB8:0:3::3, which is the IPv6 address of the Gig0/0 interface on R3. The trace confirms that the packets are being sent though R2.

Example 12-30 *Trace from PC1 to R3s Gig0/0 Interface*

```
C:\PC1>tracert 2001:DB8:0:3::3
Tracing route to 2001:DB8:0:3::3 over a maximum of 30 hops

 1      6 ms      1 ms      2 ms    2001:DB8:0:1::1
 2      5 ms      1 ms      2 ms    2001:DB8:0:12::2
 3      5 ms      1 ms      2 ms    2001:DB8:0:23::3

Trace complete.
```

Now you issue the **show ipv6 route 2001:DB8:0:3::/64** command on R1, as shown in Example 12-31, and confirm that the next-hop IPv6 address for 2001:DB8:0:3::/64 is 2001:DB8:0:12::2, which is the IPv6 address of R2s Gig0/0 interface. The next-hop IPv6 address should be 2001:DB8:0:13::3 which is R3's Gig2/0 interface.

Example 12-31 *Verifying the IPv6 Route to 2001:DB8:0:3::/64 on R1*

```
R1#show ipv6 route 2001:DB8:0:3::/64
Routing entry for 2001:DB8:0:3::/64
Known via "static", distance 10, metric 0
Backup from "static [11]"
Route count is 1/1, share count 0
Routing paths:
2001:DB8:0:12::2
Last updated 00:09:07 ago
```

It appears that someone provided the incorrect next-hop IPv6 address in the static route. You verify the static route configured on R1 for the 2001:DB8:0:3::/64 network using the **show run | include ipv6 route** command, as shown in Example 12-32. You notice that there are two commands for the network 2001:DB8:0:3::/64. One has a next hop of 2001:DB8:0:12::2, and the other has a next hop of 2001:DB8:0:13::3.

Example 12-32 *Verifying the IPv6 Static Routes Configured on R1*

```
R1#show run | include ipv6 route
ipv6 route 2001:DB8:0:3::/64 2001:DB8:0:12::2 10
ipv6 route 2001:DB8:0:3::/64 2001:DB8:0:13::3 11
ipv6 route 2001:DB8:0:23::/64 2001:DB8:0:12::2
```

Why is the **ipv6 route** command with the next hop of 2001:DB8:0:12::2 being preferred over the command with a next hop of 2001:DB8:0:13::3? If you look closely at both commands in Example 12-32, you will notice that the one with a next hop of 2001:DB8:0:12::2 is configured with an AD of 10 and that the other, which has a next hop of 2001:DB8:0:13::3, is configured with an AD of 11. Because lower AD is preferred, the static route with the AD of 10 is more trustworthy and therefore used.

To solve this issue, you need to configure the static route with the next hop of 2001:DB8:0:13::3 with a lower AD. In this case, you change the AD to 1, which is the

default for static routes, with the **ipv6 route 2001:DB8:0:3::/64 2001:DB8:0:13::3 1** command. After the change, you revisit the routing table with the **show ipv6 route 2001:DB8:0:3::/64** command to verify that the static route with the next hop of 2001:DB8:0:13::3 is now in the routing table. Example 12-33 confirms that the change was successful.

Example 12-33 Verifying IPv6 Routing Table on R1

```
R1#show ipv6 route 2001:DB8:0:3::/64
Routing entry for 2001:DB8:0:3::/64
Known via "static", distance 1, metric 0
Backup from "static [11]"
Route count is 1/1, share count 0
Routing paths:
2001:DB8:0:13::3
Last updated 00:01:14 ago
```

Now you perform a trace from PC1 to 2001:DB8:0:3::3, as shown in Example 12-34, and it confirms that R2 is no longer being used. The traffic is now flowing across the link between R1 and R3.

Example 12-34 Trace from PC1 to R3's Gig0/0 Interface

```
C:\PC1>tracert 2001:DB8:0:3::3
Tracing route to 2001:DB8:0:3::3 over a maximum of 30 hops

 1      6 ms      1 ms      2 ms  2001:DB8:0:1::1
 2      5 ms      1 ms      2 ms  2001:DB8:0:13::3

Trace complete.
```

Troubleshooting GRE Tunnels

Generic routing encapsulation (GRE) is a tunneling protocol that is used to encapsulate various types of network layer packets inside a transport protocol (GRE) so that they can be transported over an IP network. For example, you could take IPv6 network layer packets and encapsulate them in GRE so that they can be transported over an IPv4 network. Discussions related to GRE can be extensive. However, this book is focused on getting you ready for the TSHOOT certification exam. Therefore, we focus our GRE discussion on exam preparation. As a result, this section covers the benefits of using GRE for site-to-site VPNs in addition to the issues that could cause your GRE tunnels for IPv4 and IPv6 packets not to function as expected.

With GRE, you can create virtual point-to-point links between remote Cisco routers across an IP network, as shown in Figure 12-11. HQ, which is in San Francisco, and Branch, which is in Toronto, are both connected to the Internet and are not directly connected to each other. However, by using a GRE tunnel, you can virtually directly connect

the two routers. Example 12-35 provides a sample configuration that would be used on HQ to configure the GRE tunnel, and Example 12-36 provides a sample configuration that would be used on Branch to configure the GRE tunnel. The tunnel IP address is 172.16.1.1 for HQ and 172.16.1.2 for Branch. HQ is using Fa3/0 as the source of the tunnel and the IP address of 203.0.113.1 as the tunnel destination. Branch is using Fa1/0 as the source of the tunnel and the IP address of 192.0.2.1 as the tunnel destination. The tunnel mode is not listed, which means that the default tunnel mode (point-to-point GRE) is being used. For Cisco devices, the default tunnel mode is GRE/IP. If you had changed the mode and now you want to revert back to the default mode, you use the command **tunnel mode gre ip** in interface tunnel configuration mode.

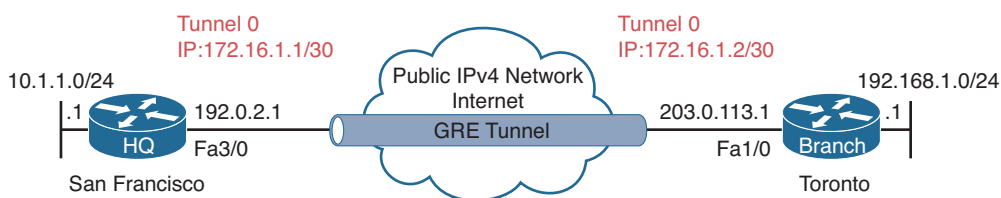


Figure 12-11 GRE Tunnel Example

Example 12-35 GRE Tunnel Configuration on HQ

```
HQ#show run int tunnel 0
Building configuration...

Current configuration : 127 bytes
!
interface Tunnel0
ip address 172.16.1.1 255.255.255.252
tunnel source FastEthernet3/0
tunnel destination 203.0.113.1
end
```

Example 12-36 GRE Tunnel Configuration on Branch

```
Branch#show run int tunnel 0
Building configuration...

Current configuration : 125 bytes
!
interface Tunnel0
ip address 172.16.1.2 255.255.255.252
tunnel source FastEthernet1/0
tunnel destination 192.0.2.1
end
```

An advantage of using GRE in this scenario is that HQ and Branch can dynamically learn about the private IPv4 networks at each site by exchanging routing information over the

tunnel. Without the tunnel, this would not occur because the public IPv4 network would not allow for the exchange of such information between HQ and Branch. GRE accomplishes this by adding a GRE header (carrier protocol) to encapsulate the original packet (passenger protocol) and then adding a new IP header (transport protocol), which will be used to transport the GRE encapsulated packet over the public IPv4 network, as shown in Figure 12-12. As a result of this, users in HQ can successfully access resources in Branch (through the tunnel), and users in Branch can successfully access resources in HQ (through the tunnel), as shown in Example 12-37.

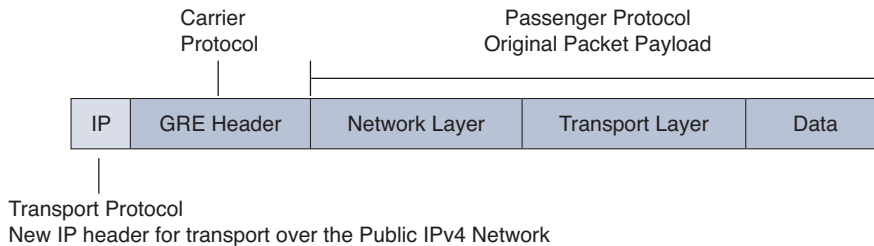


Figure 12-12 GRE Encapsulated Packet

Example 12-37 Verifying Routes via Tunnel Interface on HQ and Branch

```
HQ#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
- ODR, P - periodic downloaded static route, H - NHRP, l - LISP
+ - replicated route, % - next hop override

Gateway of last resort is 192.0.2.2 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 192.0.2.2
10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C      10.1.1.0/24 is directly connected, GigabitEthernet0/0
L      10.1.1.1/32 is directly connected, GigabitEthernet0/0
172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C      172.16.1.0/30 is directly connected, Tunnel0
L      172.16.1.1/32 is directly connected, Tunnel0
192.0.2.0/24 is variably subnetted, 2 subnets, 2 masks
C      192.0.2.0/30 is directly connected, FastEthernet3/0
L      192.0.2.1/32 is directly connected, FastEthernet3/0
D      192.168.1.0/24 [90/26880256] via 172.16.1.2, 00:14:08, Tunnel0

Branch#show ip route
```

```
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
- ODR, P - periodic downloaded static route, H - NHRP, l - LISP
+ - replicated route, % - next hop override
```

```
Gateway of last resort is 203.0.113.2 to network 0.0.0.0
```

```
S*    0.0.0.0/0 [1/0] via 203.0.113.2
10.0.0.0/24 is subnetted, 1 subnets
D     10.1.1.0 [90/26880256] via 172.16.1.1, 00:15:29, Tunnel0
172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C     172.16.1.0/30 is directly connected, Tunnel0
L     172.16.1.2/32 is directly connected, Tunnel0
192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C     192.168.1.0/24 is directly connected, GigabitEthernet0/0
L     192.168.1.1/32 is directly connected, GigabitEthernet0/0
203.0.113.0/24 is variably subnetted, 2 subnets, 2 masks
C     203.0.113.0/30 is directly connected, FastEthernet1/0
L     203.0.113.1/32 is directly connected, FastEthernet1/0
```

When troubleshooting GRE issues, you need to consider the following:

- **Are the remote devices reachable across the public network?** To form a GRE tunnel between the two remote devices, they must be able to reach each other's public IP address. You can verify this using the **ping** command on each router, as shown in Example 12-38.
- **Are the tunnel IP addresses in the same subnet?** With GRE, you are creating a virtual point-to-point connection between two remote devices; therefore, they need to be in the same subnet. You can verify the IP address on a tunnel interface, as shown in Example 12-39, using the **show interfaces tunnel *tunnel_number*** command or the **show ip interface brief** command.
- **Are the correct tunnel source and destination IP addresses specified?** The tunnel needs to know where it starts and where it ends. This is based on a source IP address and a destination IP address. These addresses need to be reachable, need to be accurate, and need to be symmetrical (source on each router must match destination on other router and vice versa). You can verify the tunnel source and destination IP addresses using the **show interfaces tunnel *tunnel_number*** command, as shown in Example 12-39.
- **Is the correct tunnel mode specified?** To transport IPv4 or IPv6 packets using GRE over an IPv4 network a tunnel mode of GRE IP is required. You can verify the tunnel mode used on a tunnel interface with the **show interfaces tunnel *tunnel_number*** command, as shown in Example 12-39.

- **Is an access control list (ACL) blocking GRE packets?** GRE uses IP protocol number 47. If an ACL exists along the path between the remote devices that is denying protocol 47 or not permitting protocol 47, GRE packets will be dropped in transit. You can verify whether an ACL is applied to an interface with the **show ip interface interface_type interface_number** command, and you can verify the entries in an ACL with the **show access-list** command.
- **Is fragmentation occurring due to insufficient maximum transmission unit (MTU)?** Because the GRE header is 24 bytes, this limits the original packet payload to 1476 bytes for a total of 1500 bytes, which equals the typical MTU of an interface. This can become an issue if large packets (original packet payload) bigger than 1476 bytes have to cross the GRE tunnel. Because the combined GRE header and original packet payload will be larger than 1500 bytes, fragmentation will occur. This results in processing delays and high CPU usage. To overcome this issue, you have to implement a consistent MTU from end to end. You can verify the MTU on a tunnel interface by using the **show interface tunnel tunnel_number** command, as shown in Example 12-39.
- **Is the recursive routing table lookup pointing back to the tunnel?** If you receive the syslog message *%TUN-5-RECURDOWN: Tunnel0 temporarily disabled due to recursive routing*, it is indicating that the router is trying to route to the tunnel destination (the public IPv4 address on the Internet in our example) using the virtual tunnel interface instead of the physical interface connected to the Internet. This can be temporary due to a flapping route elsewhere in the network, or it could be permanent due to a misconfiguration that is causing the router to route packets to the tunnel destination through the virtual tunnel interface.
- **Is the routing protocol enabled on the tunnel interface?** For routes to be shared dynamically over the tunnel, the tunnel interface needs to be participating in a routing process (for example, RIP, EIGRP, OSPF).

Example 12-38 Verifying Connectivity Between Remote Routers

```
HQ#ping 203.0.113.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 203.0.113.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 36/67/104 ms

Branch#ping 192.0.2.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.0.2.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/67/84 ms
```

Key
Topic**Example 12-39** *Verifying Tunnel Addresses, Mode, and MTU*

```

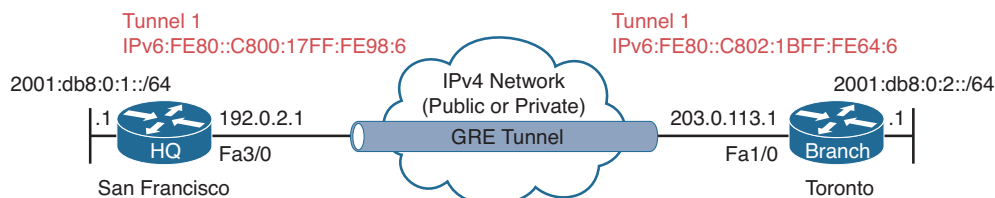
HQ#show interfaces tunnel 0
Tunnel0 is up, line protocol is up
Hardware is Tunnel
Internet address is 172.16.1.1/30
MTU 17916 bytes, BW 100 Kbit/sec, DLY 50000 usec,
reliability 255/255, txload 1/255, rxload 1/255
Encapsulation TUNNEL, loopback not set
Keepalive not set
Tunnel source 192.0.2.1 (FastEthernet3/0), destination 203.0.113.1
Tunnel Subblocks:
src-track:
Tunnel0 source tracking subblock associated with FastEthernet3/0
Set of tunnels with source FastEthernet3/0, 1 member (includes iterators), on inter-
face <OK>
Tunnel protocol/transport GRE/IP
Key disabled, sequencing disabled
Checksumming of packets disabled
Tunnel TTL 255, Fast tunneling enabled
Tunnel transport MTU 1476 bytes
Tunnel transmit bandwidth 8000 (kbps)
Tunnel receive bandwidth 8000 (kbps)
...output omitted...

HQ#show ip interface brief

```

Interface	IP-Address	OK?	Method	Status	Protocol
Ethernet0/0	unassigned	YES	unset	administratively down	down
GigabitEthernet0/0	10.1.1.1	YES	manual	up	up
FastEthernet3/0	192.0.2.1	YES	manual	up	up
Tunnel0	172.16.1.1	YES	manual	up	up

Refer to Figure 12-13; the GRE tunnel is being used to transport IPv6 packets over an IPv4 network. Examples 12-40 and 12-41 display the configurations required on HQ and Branch to accomplish this. Notice that the tunnel source and tunnel destination are IPv4 addresses and that interface Tunnel 1 is using an IPv6 address. The IPv6 address is a link-local address in this case; however, it could have been a global unicast address as well. The tunnel mode is not displayed because the default is being used. The default tunnel mode on Cisco routers is point-to-point GRE/IP.

**Figure 12-13** *GRE Tunnel Example (for IPv6 Traffic)*

Example 12-40 GRE Tunnel Configuration on HQ for IPv6

```
HQ#show run interface tunnel 1
Building configuration...

Current configuration : 132 bytes
!
interface Tunnell
no ip address
ipv6 enable
ipv6 eigrp 100
tunnel source FastEthernet3/0
tunnel destination 203.0.113.1
end
```

Example 12-41 GRE Tunnel Configuration on Branch for IPv6

```
Branch#show run interface tunnel 1
Building configuration...

Current configuration : 130 bytes
!
interface Tunnell
no ip address
ipv6 enable
ipv6 eigrp 100
tunnel source FastEthernet1/0
tunnel destination 192.0.2.1
end
```

The **show interfaces tunnel 1** command is displayed in Example 12-42. You can see from this output that the GRE tunnel mode is GRE/IP even though you are transporting IPv6 packets over the IPv4 network. The tunnel source and destination are IPv4 addresses. The output of **show ipv6 interface brief** is displayed in Example 12-43 and allows you to verify the IPv6 addresses on an interface.

Example 12-42 Verifying GRE Tunnel Configuration with the **show interface tunnel Command**

```
HQ#show interface tunnel 1
Tunnell is up, line protocol is up
Hardware is Tunnel
MTU 17916 bytes, BW 100 Kbit/sec, DLY 50000 usec,
reliability 255/255, txload 1/255, rxload 1/255
Encapsulation TUNNEL, loopback not set
Keepalive not set
Tunnel source 192.0.2.1 (FastEthernet3/0), destination 203.0.113.1
Tunnel Subblocks:
```

```
src-track:
Tunnel1 source tracking subblock associated with FastEthernet3/0
Set of tunnels with source FastEthernet3/0, 2 members (includes iterators), on
interface <OK>
Tunnel protocol/transport GRE/IP
Key disabled, sequencing disabled
Checksumming of packets disabled
Tunnel TTL 255, Fast tunneling enabled
Tunnel transport MTU 1476 bytes
Tunnel transmit bandwidth 8000 (kbps)
Tunnel receive bandwidth 8000 (kbps)
...output omitted...
```

Example 12-43 Verifying IPv6 Addresses of Tunnel Interface

```
HQ#show ipv6 interface brief
Ethernet0/0          [administratively down/down]
unassigned
GigabitEthernet0/0   [up/up]
FE80::C800:17FF:FE98:8
2001:DB8:0:1::1
FastEthernet3/0      [up/up]
unassigned
Tunnel0              [up/up]
unassigned
Tunnel1              [up/up]
FE80::C800:17FF:FE98:6
```

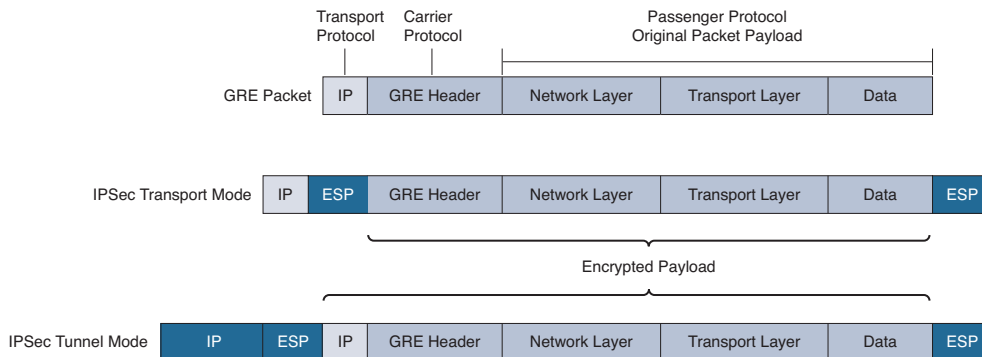
GRE is great at tunneling, but lousy at security. Its main purpose is to provide simple yet powerful tunneling for multiple network layer protocols. It provides basic plaintext authentication between the remote devices using a tunnel key, which is not a valid security solution when using GRE over an untrusted network such as the Internet.

When using GRE over an untrusted network, you want to provide confidentiality, authentication, and data integrity. You can accomplish this with IPsec. Confidentiality can be provided with symmetric algorithms, and authentication and integrity can be provided with hash message authentication codes (HMACs).

When using IPsec with GRE, GRE encapsulates the original packet payload first, and then encryption occurs next with IPsec to protect the GRE packet.



Two different IPsec modes exist that you can use to encapsulate the GRE packet (see Figure 12-14). IPsec tunnel mode will encapsulate and encrypt the entire GRE packet, including the Transport Protocol header. Because the Transport Protocol header is being encapsulated and encrypted, IPsec has to include a new IP header. IPsec transport mode will only encapsulate and encrypt the carrier protocol and the passenger protocol. Therefore, the Transport Protocol header can be reused by IPsec and reduce overhead.

**Figure 12-14** *IPsec Modes*

Benefits of using GRE and IPsec for site-to-site VPNs include the following:

- In addition to supporting IPv4 as the passenger protocol, it provides support for other Layer 3 protocols.
- It provides support for multicast and routing traffic across the IPsec VPN.
- With a hub-and-spoke topology, it reduces the management overhead needed to maintain IPsec tunnels because a minimum number of tunnels is used to provide full connectivity.

Exam Preparation Tasks

As mentioned in the section “How to Use This Book” in the Introduction, you have a couple of choices for exam preparation: the exercises here; Chapter 22, “Final Preparation;” and the exam simulation questions on the CD-ROM.

Review All Key Topics

Review the most important topics in this chapter, noted with the Key Topic icon in the outer margin of the page. Table 12-3 lists a reference of these key topics and the page numbers on which each is found.



Table 12-3 *Key Topics for Chapter 12*

Key Topic Element	Description	Page Number
List	Describes the routing table and Layer 3 to Layer 2 mapping table	430
List	Describes the FIB and adjacency table	431
Example 12-1	show ip route <i>ip_address</i> command output	432
Example 12-4	show ip cef <i>ip_address</i> command output	433
Example 12-7	show ip arp command output	433
Example 12-8	show frame-relay map command output	434
Table 12-2	Administrative distance of route sources	437
Example 12-11	Verifying the administrative distance of a route in the routing table	438
Paragraph	Outlines the importance of the next-hop address in an IPv4 static route	440
Section	Describes what occurs when an Ethernet interface is used in an IPv4 static route	441
Section	Describes what occurs when an Ethernet interface is used in an IPv6 static route	445
Example 12-39	Verifying tunnel addresses, mode, and MTU	455
Paragraph	Describes the difference between IPsec tunnel mode and transport mode	457
List	Identifies the benefits of using GRE and IPsec for site-to-site VPNs	458

Define Key Terms

Define the following key terms from this chapter and check your answers in the glossary:

packet forwarding, ARP, TTL, routing table, ARP cache, CEF, FIB, adjacency table, control plane, data plane, administrative distance, static route, proxy ARP, GRE tunnel

Complete Tables and Lists from Memory

Print a copy of Appendix C, “Memory Tables,” (found on the disc), or at least the section for this chapter, and complete the tables and lists from memory. Appendix D, “Memory Tables Answer Key,” also on the disc, includes completed tables and lists to check your work.

Command Reference to Check Your Memory

This section includes the most important **show** commands covered in this chapter. It might not be necessary to memorize the complete syntax of every command, but you should be able to remember the basic keywords that are needed.

To test your memory of the commands, cover the right side of Table 12-4 with a piece of paper, read the description on the left side, and then see how much of the command you can remember.

The 300-135 TSHOOT exam focuses on practical, hands-on skills that are used by a networking professional. Therefore, you should be able to identify the commands needed to verify and troubleshoot the topics covered in this chapter.

Table 12-4 *show Commands*

Task	Command Syntax
Displays a router's best route to the specified IP address.	show ip route <i>ip_address</i>
Displays only the static routes in a routers routing table.	show ip route static
Displays a router's best route to the specified network if the specific route (with a matching subnet mask length) is found in the router's IP routing table.	show ip route <i>ip_address subnet_mask</i>
Displays all routes in a router's IP routing table that are encompassed by the specified network address and subnet mask. (This command often proves useful when troubleshooting route summarization issues.)	show ip route <i>ip_address subnet_mask longer- prefixes</i>

Task	Command Syntax
Displays information (for example, next-hop IP address and egress interface) required to forward a packet, similar to the output of the show ip route <i>ip_address</i> command. (The output of this command comes from CEF. Therefore, routing protocol information is not presented in the output.)	show ip cef <i>ip_address</i>
Displays information from a router's FIB showing the information needed to route a packet to the specified network with the specified subnet mask.	show ip cef <i>ip_address subnet_mask</i>
Displays the adjacency that will be used to forward a packet from the specified source IP address to the specified destination IP address. (This command is useful if the router is load balancing across multiple adjacencies and you want to see which adjacency will be used for a certain combination of source and destination IP addresses.)	show ip cef exact-route <i>source_address destination_address</i>
Displays the static IPv6 routes configured on a device.	show ipv6 route static
Displays the Layer 3 IPv6 address to Layer 2 MAC address mappings.	show ipv6 neighbors
Displays a router's ARP cache, containing IPv4 address to MAC address mappings.	show ip arp
Displays Frame Relay PVC DLCIs associated with next-hop IP addresses.	show frame-relay map
Displays the Layer 2 frame header information in a router's CEF adjacency table that is used to encapsulate a frame being sent to an adjacent router.	show adjacency detail
Displays whether Proxy ARP is enabled on an interface as well as the IPv4 address and mask assigned to the interface.	show ip interface <i>interface_type interface_number</i>
Displays the configuration of a tunnel interface in the running configuration.	show run interface tunnel <i>tunnel_number</i>
Displays the status of a tunnel, the IP address of the tunnel, the tunnel source and destination, along with the tunnel mode and the tunnel transport MTU.	show interfaces tunnel <i>tunnel_number</i>



This chapter covers the following topics:

- **Troubleshooting RIPv2:** This section covers the different issues that could cause routes to be missing in RIPv2 domains. You will also learn the different commands that you can use to identify and troubleshoot these issues.
- **Troubleshooting RIPng:** This section explains the different commands that you can use to identify and troubleshoot issues related to RIPng.
- **RIPv2 and RIPng Troubleshooting:** This section provides trouble tickets that demonstrate how you can use a structured troubleshooting process to solve a reported problem.

Troubleshooting RIPv2 and RIPv6

Routing Information Protocol (RIP) is one of the oldest dynamic routing protocols. It is a distance vector routing protocol that relies on hop count as the routing metric. It is not scalable like Enhanced Interior Gateway Routing Protocol (EIGRP) and Open Shortest Path First (OSPF), and it takes a considerable amount of time to fully converge after a topology change. Therefore, if used, it is used in small and simple routed networks. RIPv2 is designed for IPv4 routed networks, and RIP next generation (RIPv6) is designed for IPv6 routed networks.

This chapter focuses on the issues you may have to troubleshoot in a RIPv2 and RIPv6 domain. This includes how you would recognize the issues based on the presented symptoms and the commands you would use to successfully verify the reason why the issue exists.

“Do I Know This Already?” Quiz

The “Do I Know This Already?” quiz allows you to assess whether you should read this entire chapter thoroughly or jump to the “Exam Preparation Tasks” section. If you are in doubt about your answers to these questions or your own assessment of your knowledge of the topics, read the entire chapter. Table 13-1 lists the major headings in this chapter and their corresponding “Do I Know This Already?” quiz questions. You can find the answers in Appendix A, “Answers to the ‘Do I Know This Already?’ Quizzes.”

Table 13-1 “Do I Know This Already?” Section-to-Question Mapping

Foundation Topics Section	Questions
Troubleshooting RIPv2	1–5, 10
Troubleshooting RIPv6	6–10

Caution The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you do not know the answer to a question or are only partially sure of the answer, you should mark that question as wrong for purposes of the self-assessment. Giving yourself credit for an answer that you correctly guess skews your self-assessment results and might provide you with a false sense of security.

1. Which command enables you to verify all RIPv2 routes learned from directly connected devices?
 - a. `show ip route`
 - b. `show ip route rip`
 - c. `show ip protocols`
 - d. `show ip rip database`
2. Which command enables you to verify the interfaces that are sending and receiving RIPv2 routing updates?
 - a. `show ip route`
 - b. `show ip route rip`
 - c. `show ip protocols`
 - d. `show ip rip database`
3. Which command will enable the RIPv2 routing process on the interface with an IP address of 10.1.1.7/28?
 - a. `network 10.0.0.0`
 - b. `network 10.1.1.7 0.0.0.0`
 - c. `ip rip enable`
 - d. `ip ripv2 enable`
4. What occurs when you configure a passive interface with RIPv2? (Choose two answers.)
 - a. The interface will send RIP updates.
 - b. The interface will receive RIP updates.
 - c. The interface will suppress the sending of RIP updates.
 - d. The interface will suppress the receiving of RIP updates.
5. What is the maximum hop count for RIPv2?
 - a. 5
 - b. 10
 - c. 15
 - d. 16

6. Which command enables you to verify all RIPv6 routes learned from directly connected devices?
 - a. `show ipv6 rip`
 - b. `show ipv6 route rip`
 - c. `show ipv6 protocols`
 - d. `show ipv6 rip database`
7. Which commands enable you to verify the interfaces that are participating in the RIPv6 routing process? (Choose two answers.)
 - a. `show ipv6 rip`
 - b. `show ipv6 route rip`
 - c. `show ipv6 protocols`
 - d. `show ipv6 rip database`
8. Which command enables you to verify the number of paths that will be used by RIPv6 for load balancing?
 - a. `show ipv6 rip`
 - b. `show ipv6 route rip`
 - c. `show ipv6 protocols`
 - d. `show ipv6 rip database`
9. Which command enables you to verify whether default routes are being generated by RIPv6?
 - a. `show ipv6 rip`
 - b. `show ipv6 route rip`
 - c. `show ipv6 protocols`
 - d. `show ipv6 rip database`
10. Which of the following are reasons why a RIPv6 route may be missing from a router running RIPv2 or RIPv6? (Choose three answers.)
 - a. Bad or missing network statement
 - b. Max hop count exceeded
 - c. ACLs
 - d. Neighbor relationship not formed

Foundation Topics

Troubleshooting RIPv2

Routing Information Protocol Version 2 (RIPv2) does not establish neighbor adjacencies; therefore, you will not have to troubleshoot neighbor-related issues like you do with Enhanced Interior Gateway Routing Protocol (EIGRP), Open Shortest Path First (OSPF), and Border Gateway Protocol (BGP). However, with RIPv2, you will be troubleshooting issues related to routing updates. This section covers the reasons why a RIPv2 router may not be receiving the routes that you expect it to receive.

Missing RIPv2 Routes

A RIPv2 route may be missing from the RIP database or the routing table for many reasons. As a troubleshooter, it is important that you can recognize the reasons why routes are missing and resolve the issue quickly and efficiently.

Following is a listing of reasons as to why RIPv2 routes might be missing either in the RIP database or the routing table:



- **Interface is shut down:** The RIP-enabled interface must be up/up for the network associated with the interface to be advertised.
- **Wrong subnet:** The sender of RIP updates must be in the same subnet as the receiver of RIP updates; otherwise, updates are ignored.
- **Bad or missing network statement:** The **network** command enables the RIP process on an interface and injects the network the interface is part of into the RIP process.
- **Passive interface:** Suppresses the sending of RIP updates out an interface.
- **Wrong version:** The sender of RIP updates must be using the same RIP version as the receiver of RIP updates.
- **Max hop count exceeded:** When the maximum hop count is exceeded, the route is unreachable and not used.
- **Authentication:** If authentication parameters do not match, routing updates are ignored.
- **Route filtering:** A filter might be set up that is preventing a route from being advertised or learned.
- **Split horizon:** Loop-prevention feature that prevents a router from advertising routes out the same interface they were learned on.
- **Autosummarization:** Summarizes classless networks at classful boundaries, which is problematic in discontinuous networks.

- **Better source of information:** If the exact same network is learned from a more reliable source (as determined by the administrative distance), it is used instead of the RIP-learned information.
- **ACLs:** If an access control list (ACL) is denying RIP packets in an interface, routes will not be learned.
- **Load balancing:** If the maximum paths value is incorrectly set, equal metric paths for certain routes will be missing from the routing table.

Key Topic

To verify all routes that have been learned from neighboring routers and the directly connected routes that have been injected into the RIP process, use the **show ip rip database** command. This command displays the RIP database, as shown in Example 13-1. In this example, you can see two directly connected networks (10.1.1.0/24 and 10.1.12.0/24) that have been injected into the RIP process and two networks (10.1.3.0/24 and 10.1.23.0/24) that have been learned from the neighbor with an IP address of 10.1.12.2, which is also the next hop to reach those networks. The 10.1.3.0/24 network has a hop count of 2, and the 10.1.23.0/24 network has a hop count of 1. Remember that autosummarization is on by default with RIP; as a result, the router will automatically create a classful summary route, as seen with the 10.0.0.0/8 network.

Example 13-1 Viewing the RIP Database with the `show ip rip database` Command

```
R1#show ip rip database
10.0.0.0/8      auto-summary
10.1.1.0/24     directly connected, GigabitEthernet0/0
10.1.3.0/24
    [2] via 10.1.12.2, 00:00:03, GigabitEthernet1/0
10.1.12.0/24    directly connected, GigabitEthernet1/0
10.1.23.0/24
    [1] via 10.1.12.2, 00:00:03, GigabitEthernet1/0
```

To verify the RIP routes that have been installed in the routing table, use the **show ip route rip** command, as shown in Example 13-2. In this example, there are two RIP routes with a next-hop address of 10.1.12.2. Because they are learned via RIP, they have an administrative distance (AD) of 120 by default. The metric (hop count) for 10.1.3.0/24 is 2, as shown in the brackets, and 1 for 10.1.23.0/24.

Key Topic

Example 13-2 Viewing the RIP Installed Routes in the Routing Table with the `show ip route rip` Command

```
R1#show ip route rip
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
```



```
+ - replicated route, % - next hop override
```

```
Gateway of last resort is not set
```

```
10.0.0.0/8 is variably subnetted, 6 subnets, 2 masks
```

```
R    10.1.3.0/24 [120/2] via 10.1.12.2, 00:00:06, GigabitEthernet1/0
```

```
R    10.1.23.0/24 [120/1] via 10.1.12.2, 00:00:06, GigabitEthernet1/0
```



To verify various RIP parameters and settings, use the **show ip protocols** command, as shown in Example 13-3. With **show ip protocols**, you can verify route filters that have been applied, timers, redistribution, which versions of RIP are being sent and received, the status of automatic summarization, maximum paths for load balancing, the address that was used for the **network** command, any passive interfaces, who the router has learned routes from, and finally, the AD. You will examine most of these as we go through examples.

Example 13-3 Viewing RIP Settings with the show ip protocols Command

```
R1#show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "rip"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Sending updates every 30 seconds, next due in 21 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Redistributing: rip
  Default version control: send version 2, receive version 2
    Interface          Send Recv Triggered RIP Key-chain
  GigabitEthernet1/0    2      2
  Automatic network summarization is not in effect
  Maximum path: 4
  Routing for Networks:
    10.0.0.0
  Passive Interface(s):
    Ethernet0/0
    GigabitEthernet0/0
    GigabitEthernet2/0
  Routing Information Sources:
    Gateway         Distance      Last Update
    10.1.12.2        120          00:00:14
  Distance: (default is 120)
```

Let's examine each of the issues previously listed on an individual basis and identify how we can troubleshoot them.

Interface Is Shut Down

For an interface to participate in the RIP routing process, it must be up/up. You can verify the status of an interface with the **show ip interface brief** command, as shown in Example 13-4.

Example 13-4 Verifying the Status of an Interface

```
R1#show ip interface brief
```

Interface	IP-Address	OK?	Method	Status	Protocol
Ethernet0/0	unassigned	YES	NVRAM	administratively down	down
GigabitEthernet0/0	10.1.1.1	YES	NVRAM	up	up
GigabitEthernet1/0	10.1.12.1	YES	NVRAM	up	up
GigabitEthernet2/0	unassigned	YES	NVRAM	up	up

Wrong Subnet

RIP routers exchanging RIP updates must be in the same subnet. If they are not, they will ignore the RIP updates that they receive from each other. See Figure 13-1, which displays a sample RIP domain where the link between R1 and R2 is not addressed properly. Notice that R1's interface connected to R2 is in the 10.1.10.0/24 network and the interface on R2 connected to R1 is in the 10.1.12.0/24 network. When either R1 or R2 receive a RIP update from each other, they will ignore it, as shown in Example 13-5, which displays the output of **debug ip rip** on R1.



Figure 13-1 Example of Wrong Addressing Between R1 and R2

Example 13-5 Output of debug ip rip on R1

```
R1#debug ip rip
RIP protocol debugging is on
R1#
RIP: ignored v2 update from bad source 10.1.12.2 on GigabitEthernet1/0
```

As a result, both the IP address and the subnet mask assigned to an interface must be correct. In this case, the link between R1 and R2 is the 10.1.12.0/24 network. Reviewing the output of **show ip interface gigabitethernet1/0** on R1 in Example 13-6 reveals that it is configured with a 10.1.10.1/24 address and mask. Therefore, it would have to be changed so that the address is 10.1.12.1/24 by using the interface command **ip address 10.1.12.1 255.255.255.0**.

Example 13-6 *Output of show ip interface on R1*

```
R1#show ip interface gigabitethernet1/0
GigabitEthernet1/0 is up, line protocol is up
  Internet address is 10.1.10.1/24
  Broadcast address is 255.255.255.255
  Address determined by setup command
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Multicast reserved groups joined: 224.0.0.9
  ...output omitted...
```

Bad or Missing Network Statement

The RIP process is enabled on interfaces using the classful **network** command in router RIP configuration mode. Once an interface is enabled for RIP, the network the interface is part of will be injected into the RIP routing process and advertised to directly connected routers out RIP-enabled interfaces. If the network command is missing, or is incorrect, the RIP process will not be enabled on the interface, and routes will be missing in the RIP domain.



To verify the interfaces enabled for RIP, use the **show ip protocols** command, as shown in Example 13-7. In this example, interfaces Gigabit Ethernet 0/0 and Gigabit Ethernet 1/0 are participating in the RIP routing process. In addition, you will notice the *Routing for Networks:* area that indicates 10.0.0.0. This is really the **network** command that was used to enable the RIP routing process on the interfaces. It states that the RIP routing process will be enabled on any interface that has a first octet with 10 in it. Reviewing the running configuration with the command **show run | section rip**, as shown in Example 13-8, confirms the **network** command is **network 10.0.0.0**. Using the command **show ip interface brief**, as shown in Example 13-9, indicates that Gigabit Ethernet 0/0 and Gigabit Ethernet 1/0 both have an IP address that begin with a 10 in the first octet and therefore will participate in the RIP routing process.

Example 13-7 *Verifying Interfaces Participating in the RIP Process With show ip protocols*

```
R1#show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "rip"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Sending updates every 30 seconds, next due in 28 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Redistributing: rip
  Default version control: send version 2, receive version 2
    Interface                Send Recv Triggered RIP Key-chain
```

```
GigabitEthernet0/0    2    2
GigabitEthernet1/0    2    2
Automatic network summarization is not in effect
Maximum path: 4
Routing for Networks:
  10.0.0.0
Routing Information Sources:
  Gateway         Distance      Last Update
Distance: (default is 120)
```

Example 13-8 Verifying RIP Configurations in the Running Configuration

```
R1#show run | section router rip
router rip
version 2
network 10.0.0.0
no auto-summary
```

Example 13-9 Verifying Interface IP Addresses

```
R1#show ip interface brief
```

Interface	IP-Address	OK?	Method	Status	Protocol
Ethernet0/0	unassigned	YES	NVRAM	administratively down	down
GigabitEthernet0/0	10.1.1.1	YES	NVRAM	up	up
GigabitEthernet1/0	10.1.12.1	YES	NVRAM	up	up
GigabitEthernet2/0	unassigned	YES	NVRAM	up	up

Passive Interface

The passive interface feature is a must have for all routing domains. It does two things: reduces the RIP related traffic on a LAN, and improves RIP security.

The passive interface feature for RIP will disable the sending of RIP updates out of the interface that is passive. Therefore, it eliminates the RIP-related traffic on the LAN leaving the router interface. This slightly improves the security of RIP because the router is not advertising RIP information out an interface that could be captured by a malicious user. However, the interface will still receive RIP updates and use them; as a result, it is possible for rogue RIP routes to be introduced into the RIP domain. Remember that when an interface is passive, the network/subnet the interface is part of will still be injected into the RIP routing process and advertised to other RIP routers.

Consider Figure 13-2, where Gigabit Ethernet 1/0 of R1 has been configured as a passive interface. In this case, R1 will not send RIP updates out Gig1/0 to R2; however, it will still receive RIP updates from R2. Therefore, R1 will know how to get to the 10.1.3.0/24 network, but R2 and R3 will not know how to reach the 10.1.1.0/24 network. Therefore, end-to-end routing is broken in this RIP domain because the passive interface feature was enabled on the wrong interface.

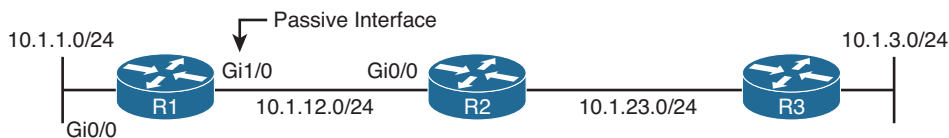


Figure 13-2 Example of Passive Interface Configured on Wrong Interface

Example 13-10 displays the output of **show ip route** on R1 and R2. Notice how R2 lacks the route 10.1.1.0/24 but R1 knows about 10.1.3.0/24 and 10.1.23.0/24.

Example 13-10 Verifying RIP Routes on R1 and R2

```

R1#show ip route
...output omitted
    10.0.0.0/8 is variably subnetted, 6 subnets, 2 masks
C       10.1.1.0/24 is directly connected, GigabitEthernet0/0
L       10.1.1.1/32 is directly connected, GigabitEthernet0/0
R       10.1.3.0/24 [120/2] via 10.1.12.2, 00:00:04, GigabitEthernet1/0
C       10.1.12.0/24 is directly connected, GigabitEthernet1/0
L       10.1.12.1/32 is directly connected, GigabitEthernet1/0
R       10.1.23.0/24 [120/1] via 10.1.12.2, 00:00:04, GigabitEthernet1/0

R2#show ip route
...output omitted...
    10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
R       10.1.3.0/24 [120/1] via 10.1.23.3, 00:00:04, GigabitEthernet1/0
C       10.1.12.0/24 is directly connected, GigabitEthernet0/0
L       10.1.12.2/32 is directly connected, GigabitEthernet0/0
C       10.1.23.0/24 is directly connected, GigabitEthernet1/0
L       10.1.23.2/32 is directly connected, GigabitEthernet1/0
  
```

To verify whether there are any passive interfaces configured, you use **show ip protocols**, as shown in Example 13-11. In this case, Gigabit Ethernet 1/0 is a passive interface when it should not be.



Example 13-11 Verifying RIP Passive Interfaces with show ip protocols

```

R1#show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "rip"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Sending updates every 30 seconds, next due in 6 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Redistributing: rip
  Default version control: send version 2, receive version 2
    Interface          Send Recv Triggered RIP Key-chain
  
```

```

GigabitEthernet0/0    2    2
Automatic network summarization is not in effect
Maximum path: 4
Routing for Networks:
  10.0.0.0
Passive Interface(s):
  GigabitEthernet1/0
Routing Information Sources:
  Gateway          Distance      Last Update
  10.1.12.2         120          00:00:07
Distance: (default is 120)

```

Wrong Version

By default, RIPv1 is enabled when you start the RIP routing process with the **router rip** global configuration command. To enable RIPv2, you issue the **version 2** command in router RIP configuration mode. If directly connected routers are not using the same version, they will not share routing information. See Figure 13-3, which shows that R1 is using RIPv1 and R2 is using RIPv2.

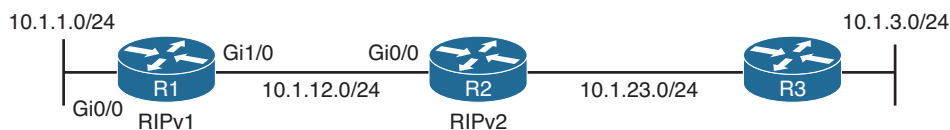


Figure 13-3 Example of Routers Using Incorrect RIP Versions

Example 13-12 displays the output of **debug ip rip** on R1 and R2. Notice that they ignore the RIP routing updates from each other as they are “illegal version.” As a result, R1 in this case will not learn any RIP routes and will only have directly connected routes in the routing table. R2 will not learn any RIP routes from R1 either.

Example 13-12 Using debug ip rip to Determine Why Routes Are Not Received

```

R1#debug ip rip
RIP: ignored v2 packet from 10.1.12.2 (illegal version)
R1#u all

R2#debug ip rip
RIP: ignored v1 packet from 10.1.12.1 (illegal version)
R2#u all

```

You can verify which version of RIP is being used on a router with the **show ip protocols** command. As shown in Example 13-13, RIPv1 is being used for both sent updates and received updates on R1, and RIPv2 is being used for both sent and received updates on R2.

Example 13-13 *Verifying the Version of RIP Being Used*

```

R1#show ip protocols
...output omitted...
Invalid after 180 seconds, hold down 180, flushed after 240
Redistributing: rip
Default version control: send version 1, receive version 1
  Interface          Send  Recv  Triggered RIP  Key-chain
  GigabitEthernet0/0  1    1
  GigabitEthernet1/0  1    1
Automatic network summarization is not in effect
...output omitted...

R2#show ip protocols
...output omitted...
Invalid after 180 seconds, hold down 180, flushed after 240
Redistributing: rip
Default version control: send version 2, receive version 2
  Interface          Send  Recv  Triggered RIP  Key-chain
  GigabitEthernet0/0  2    2
  GigabitEthernet1/0  2    2
Automatic network summarization is not in effect
...output omitted...

```

Note that with RIP you can control on an interface-by-interface basis with the **ip rip send version** and **ip rip receive version** commands which version of RIP is used to send and receive updates, regardless of the version specified in router RIP configuration mode. Therefore, it is possible to have version 1 running but send v2 updates out an interface, or version 2 running and have v1 updates sent out an interface, as shown in Example 13-14. Notice that version 2 is running but Gigabit Ethernet 0/0 is using version 1 for sending and receiving updates, which in this case would align with R1 using version 1 and routes being exchanged successfully.

Example 13-14 *Controlling RIP Version on an Interface Basis*

```

R2#show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "rip"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Sending updates every 30 seconds, next due in 13 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Redistributing: rip
  Default version control: send version 2, receive version 2
    Interface          Send  Recv  Triggered RIP  Key-chain
    GigabitEthernet0/0  1    1

```

```
GigabitEthernet1/0    2    2
Automatic network summarization is in effect
Maximum path: 4
Routing for Networks:
  10.0.0.0
Routing Information Sources:
  Gateway          Distance      Last Update
  10.1.12.1         120          00:00:26
  10.1.23.3         120          00:00:12
Distance: (default is 120)
```

Therefore, it is important to check the version being used on the router as a whole as well as the interfaces when troubleshooting RIP related issues.

Max Hop Count Exceeded

RIP has a maximum hop count of 15. Any routes that are 16 hops or further are considered unreachable. Therefore, they will not be installed in the routing table or shared with neighboring routers. Here is a listing of reasons as to why the max hop count may be exceeded:



- **The physical topology is too large:** If there are too many physical routers (15 or more) from the local router to the destination network, the hop count will be exceeded. You will need to review your network topologies and consult your documentation to verify this.
- **The seed metric during redistribution was set to high:** When routes are redistributed into RIP, you must manually set a seed metric. If you set it too high, it is possible that the route will not get advertised to the furthest RIP routers in the domain because the max hop count is reached before the routers can learn about the redistributed route. RIP redistribution is covered in a later chapter.
- **There is an offset list applied:** You can use an offset list to manipulate the metric of RIP routes by adding hops before the route is advertised or once it is received. If the offset is set to high, it is possible that the route will not get advertised to the furthest RIP routers in the domain because the max hop count is reached before the routers can learn about the route.

You can verify whether an offset list is applied using **show ip protocols**, as shown in Example 13-15. In this example, it states that routes received inbound on Gigabit Ethernet 1/0 that match ACL 1 will have 14 hops added to their metric. ACL 1 is shown in Example 13-16 with the **show ip access-list 1** command. It matches routes that have an address of 10.1.3.0.

Example 13-15 Verifying Applied Offset Lists

```
R2#show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "rip"
```



```

Outgoing update filter list for all interfaces is not set
Incoming update filter list for all interfaces is not set
Incoming routes in GigabitEthernet1/0 will have 14 added to metric if on list 1
Sending updates every 30 seconds, next due in 11 seconds
Invalid after 180 seconds, hold down 180, flushed after 240
Redistributing: rip
Default version control: send version 2, receive version 2
  Interface          Send  Recv  Triggered RIP  Key-chain
  GigabitEthernet0/0    2    2
  GigabitEthernet1/0    2    2
Automatic network summarization is in effect
Maximum path: 4
Routing for Networks:
  10.0.0.0
Routing Information Sources:
  Gateway            Distance      Last Update
  10.1.12.1           120          00:00:02
  10.1.23.3           120          00:00:07
Distance: (default is 120)

```

Example 13-16 *Verifying Access List 1 on R2*

```

R2#show ip access-list 1
Standard IP access list 1
  10 permit 10.1.3.0 (14 matches)

```

Reviewing the output of **show ip route** on R2 indicates that 10.1.3.0/24 is now 15 hops away, as shown in Example 13-17.

Example 13-17 *Verifying RIP Metric for the 10.1.3.0/24 Route*

```

R2#show ip route
...output omitted...
  10.0.0.0/8 is variably subnetted, 6 subnets, 2 masks
R      10.1.1.0/24 [120/1] via 10.1.12.1, 00:00:08, GigabitEthernet0/0
R      10.1.3.0/24 [120/15] via 10.1.23.3, 00:00:28, GigabitEthernet1/0
C      10.1.12.0/24 is directly connected, GigabitEthernet0/0
L      10.1.12.2/32 is directly connected, GigabitEthernet0/0
C      10.1.23.0/24 is directly connected, GigabitEthernet1/0
L      10.1.23.2/32 is directly connected, GigabitEthernet1/0

```

Now take a look at the **debug ip rip** output on R1 in Example 13-18. Notice that when R1 receives the RIP update from R2, 10.1.3.0/24 is 16 hops away (inaccessible). Therefore, it will not be installed in R1's routing table. Because it is not installed in R1's routing table, it will not be advertised out any other RIP-enabled interfaces. You can verify this in the same **debug** output. When the RIP update is sent out Gigabit Ethernet 0/0, the 10.1.3.0/24 network is missing.

Example 13-18 *Reviewing debug ip rip Output on R1*

```

R1#debug ip rip
RIP protocol debugging is on
RIP: received v2 update from 10.1.12.2 on GigabitEthernet1/0
    10.1.3.0/24 via 0.0.0.0 in 16 hops (inaccessible)
    10.1.23.0/24 via 0.0.0.0 in 1 hops
R1#
RIP: sending v2 update to 224.0.0.9 via GigabitEthernet0/0 (10.1.1.1)
RIP: build update entries
    10.1.12.0/24 via 0.0.0.0, metric 1, tag 0
    10.1.23.0/24 via 0.0.0.0, metric 2, tag 0

```

Authentication

When authentication is configured on a RIP-enabled interface, it will only accept RIP updates that pass authentication, which improves security. As shown in the **debug ip rip** output of Example 13-19, R1 is ignoring the update from 10.1.12.2 because of invalid authentication.

Example 13-19 *Ignored Update Due to Invalid Authentication*

```

R1#debug ip rip
RIP protocol debugging is on
RIP: ignored v2 packet from 10.1.12.2 (invalid authentication)

```

When troubleshooting authentication, you need to consider all three of the following:

- Key chain configuration
- Key chain association
- Authentication Mode

RIP uses key chains for authentication; therefore, you need to be able to troubleshoot key chain configurations when troubleshooting RIP authentication. Example 13-20 displays the output of **show key chain** on R1. The key chain in this example is called RIP, it has 1 key with an ID of 1, and the key string (*text*) is TSHOOT. For RIP authentication to be successful, the key ID and the key string have to match between the router interface sending the updates and the router interface receiving the updates. However, the name of the key chain does not have to match. In addition, notice the accept and send lifetime. These are used to specify when the key will be used when sending updates and when the key will be used for received updates. By default, they are always valid (never expire). However, you can modify the lifetimes to control when keys will be used. This is important if you specify multiple keys for key rotation to enhance security.

Example 13-20 *Verifying Key Chain*

```
R1#show key chain
Key-chain RIP:
  key 1 -- text "TSHOOT"
    accept lifetime (always valid) - (always valid) [valid now]
    send lifetime (always valid) - (always valid) [valid now]
```

To assign a key chain to an interface using RIP, you use the **ip rip authentication key-chain *key_chain*** command in interface configuration mode. To specify the mode, you type **ip rip authentication mode [text|md5]**. It is imperative that neighboring RIP routers are using the same keys and the same mode. Example 13-21 displays the output of **show ip protocols**. You can see that Gigabit Ethernet 1/0 is configured to use the key chain named *RIP*. Remember that the key chain name does not have to match between the directly connected RIP routers. Therefore, once you determine the key chain applied to the interface, you would have to execute the **show key chain** command, as shown before in Example 13-20, to review the key ID, key string, and the validity of the keys to make sure that they match.

Example 13-21 *Verifying RIP Authentication*

```
R1#show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "rip"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Sending updates every 30 seconds, next due in 15 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Redistributing: rip
  Default version control: send version 2, receive version 2
    Interface          Send Recv Triggered RIP Key-chain
    GigabitEthernet1/0  2     2              RIP
  Automatic network summarization is not in effect
  Maximum path: 4
...output omitted...
```

So far, you have verified which key chain is applied and the settings of the key chain. However, you have yet to confirm the mode of authentication that is being used. RIP supports message digest 5 (MD5) authentication and simple password authentication. You are encouraged to use MD5 authentication and avoid simple password authentication in the real world. Regardless of what you use, to verify the mode you need to review the interface configuration in the running config using the **show run interface *interface_type interface_number*** command. As shown in Example 13-22, R1 is using MD5 authentication.

Example 13-22 *Verifying RIP Authentication Mode*

```
R1#show run interface gigabitethernet1/0
Building configuration...

Current configuration : 193 bytes
!
interface GigabitEthernet1/0
 ip address 10.1.12.1 255.255.255.0
 ip rip authentication mode md5
 ip rip authentication key-chain RIP
 negotiation auto
 ipv6 address 2001:DB8:0:12::1/64
end
```

The best approach when troubleshooting authentication is to compare the authentication configurations of the two devices in question and spot the difference.

Route Filtering

A distribute list applied to the RIP process controls which routes are advertised to neighbors or which routes are received from neighbors. The distribute list is applied in RIP configuration mode either inbound or outbound, and the routes sent or received are controlled by ACLs, prefix lists, or route maps, as specified by the distribute list. So, when troubleshooting route filtering, you need to consider the following:



- Is the distribute list applied in the correct direction?
- Is the distribute list applied to the correct interface?
- If the distribute list is using an ACL, is the ACL correct?
- If the distribute list is using a prefix list, is the prefix list correct?
- If the distribute list is using a route map, is the route map correct?

The **show ip protocols** command identifies whether a distribute list is applied to all interfaces or an individual interface, as shown in Example 13-23. This example indicates that there are no outbound filters and that there is an inbound filter on Gig 1/0.

Example 13-23 *Verifying Route Filters with show ip protocols*

```
R1#show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "rip"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  GigabitEthernet1/0 filtered by (prefix-list) TSHOOT_RIP (per-user), default is
  not set
```

```

Sending updates every 30 seconds, next due in 17 seconds
Invalid after 180 seconds, hold down 180, flushed after 240
Redistributing: rip
Default version control: send version 2, receive version 2
  Interface          Send  Recv  Triggered RIP  Key-chain
  GigabitEthernet1/0    2    2
...output omitted...

```

The inbound filter in Example 13-23 on Gig 1/0 is filtered by prefix list TSHOOT_RIP. To verify entries in the prefix list, you would need to issue the **show ip prefix-list TSHOOT_RIP** command. To verify entries in an ACL, you would need to issue the **show access-lists [access_list_number | access_list_name]** command. If a route map was applied, you would issue the **show route-map [map_name]** command.

As displayed in Example 13-24, you can verify the command that was used to apply the distribute list in the running configuration.

Example 13-24 Verifying the RIP Distribute List Command

```

R1#show run | section router rip
router rip
  version 2
  passive-interface default
  no passive-interface GigabitEthernet1/0
  network 10.0.0.0
  distribute-list prefix TSHOOT_RIP in GigabitEthernet1/0
  no auto-summary

```

Split Horizon



The RIP split-horizon rule states that any routes learned inbound on an interface will not be advertised out the same interface. This rule is designed to prevent routing loops. Refer to the **debug ip rip** output in Example 13-25, which shows how R1 only sends the 10.1.1.0/24 network to R2 and not 10.1.3.0/24, 10.1.23.0/24, or 10.1.12.0/24 as shown in Figure 13-4.

Example 13-25 Verifying Advertised Routes

```

R1#debug ip rip
RIP: sending v2 update to 224.0.0.9 via GigabitEthernet1/0 (10.1.12.1)
RIP: build update entries
  10.1.1.0/24 via 0.0.0.0, metric 1, tag 0
R1#

```

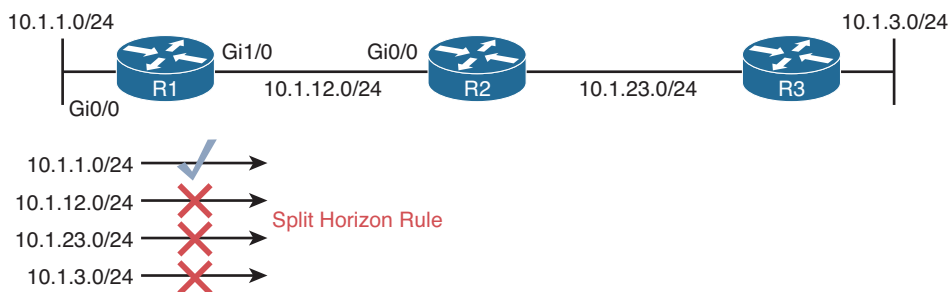


Figure 13-4 Sample Topology for Split-Horizon Rule

You can verify whether split horizon is enabled on an interface with the `show ip interface interface_type interface_number` command, as shown in Example 13-26.

Example 13-26 Verifying That Split Horizon Is Enabled

```
R1#show ip interface gigabitethernet1/0
GigabitEthernet1/0 is up, line protocol is up
  Internet address is 10.1.12.1/24
  Broadcast address is 255.255.255.255
  Address determined by setup command
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Multicast reserved groups joined: 224.0.0.9
  Outgoing access list is not set
  Inbound access list is not set
  Proxy ARP is enabled
  Local Proxy ARP is disabled
  Security level is default
  Split horizon is enabled
  ICMP redirects are always sent
  ICMP unreachable are always sent
  ICMP mask replies are never sent
```

The split-horizon rule becomes an issue in multiaccess hub-and-spoke topologies such as Frame Relay. Figure 13-5 depicts such a network. In this case, when R2 sends a RIP update about 10.1.2.0/24 to R1, R1 will not send the 10.1.2.0/24 network in its routing update out Serial 1/0 because of the split-horizon rule. Therefore, R3 never learns about 10.1.2.0/24. The same will be true about R3's update about 10.1.3.0/24. R1 will not send 10.1.3.0/24 in the update out Serial 1/0 because of the split-horizon rule, and as a result R2 will not learn about the 10.1.3.0/24 network. You have to be able to recognize this issue based on the topology and disable the split-horizon rule for RIP on the hub routers' multiaccess interface with the `no ip split-horizon` interface configuration command so that the routing updates can be sent back out the same physical interface they were received on. Your other option is to use point-to-point subinterfaces.

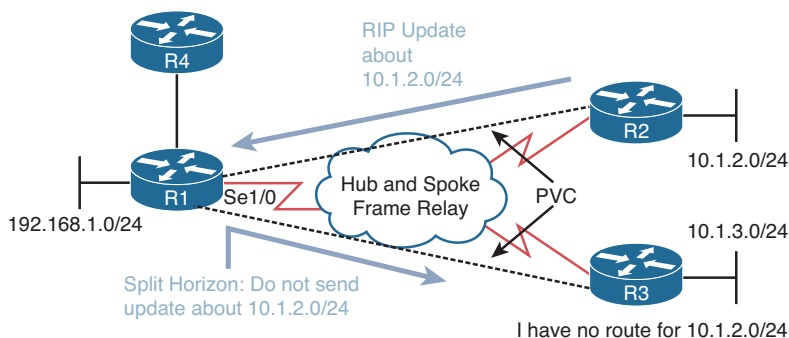


Figure 13-5 Split-Horizon in Hub-and-Spoke Multiaccess Topology

Autosummarization

RIP performs summarization automatically when it sends updates out interfaces that are part of a different classful network than the route it is advertising. This is an issue in RIP domains that have discontinuous networks. Figure 13-6 provides an example of a discontinuous network. The 172.16.0.0/16 Class B classful network is considered discontinuous because its subnets, 172.16.1.0/24 and 172.16.2.0/24, are separated from each other by another network, which is the Class A 10.0.0.0 network in this case. With automatic summarization turned on, when R3 advertises the 172.16.2.0/24 network to R2, it is summarized to 172.16.0.0/16 because it is being sent out an interface in a different network than 172.16.0.0. So, instead of 172.16.2.0/24 being sent, 172.16.0.0/16 is sent, as shown in the **debug ip rip** output of Example 13-27. Likewise, the same thing happens when R1 advertises the 172.16.1.0/24 network to R2; it is advertised as 172.16.0.0/16. If you were to review R2's routing table, it would show an entry for 172.16.0.0 with two next hops (if everything else is equal), one via R3 using Fa0/1 and the other via R1 using Fa0/0.

Now picture a packet arriving at R2 from R4 with a destination IP of 172.16.2.5. Which way does R2 send it? You see the problem? It should send it out Fa0/1, but it could send it out Fa0/0. There is a 50/50 chance that it gets it correct. The moral of this story is this: If you have a discontinuous network, autosummarization has to be off with the **no auto-summary** command in router RIP configuration mode, and you must take care when performing manual summarization. To verify whether automatic summarization is enabled or disabled, use the **show ip protocols** command, as displayed in Example 13-27.

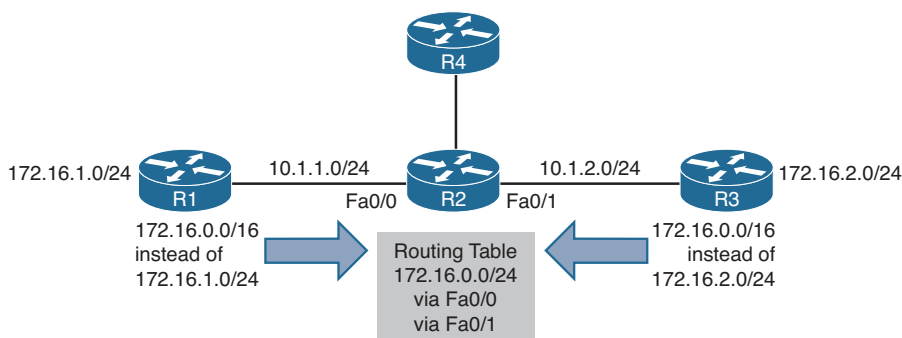


Figure 13-6 Discontinuous Network Example

Example 13-27 *Verifying Automatic Summarization*

```

R3#debug ip rip
RIP protocol debugging is on
R3#
RIP: sending v2 update to 224.0.0.9 via FastEthernet0/1 (10.1.2.3)
RIP: build update entries
    172.16.0.0/16 via 0.0.0.0, metric 1, tag 0
R3#u all
All possible debugging has been turned off
R3#show ip protocols
...output omitted...
    GigabitEthernet0/0    2    2
    GigabitEthernet1/0    2    2
Automatic network summarization is in effect
Maximum path: 4
Routing for Networks:
    10.0.0.0
Routing Information Sources:
    Gateway          Distance    Last Update
    10.1.23.2         120        00:00:12
Distance: (default is 120)

```

Better Source of Information

For a RIP-learned route to be installed in the routing table, it has to be the most believable routing source. Recall that this is based on AD. RIP's default AD is 120. Therefore, if there is another source that is educating the same router about the exact same network and that source has a better AD, the source with the better AD wins, and its information will be installed in the routing table. Review Example 13-28, which is the RIP database on R2, and Example 13-29, which is the routing table of R2 displaying only the RIP installed routes on the router. Notice that there is no entry for 10.1.3.0/24, although there should be according to Figure 13-7. Also, Example 13-30 displays the **debug ip rip** output on R2 that clearly shows R2 is learning it from R3.

Example 13-28 *Sample show ip rip database Command Output*

```

R2#show ip rip database
10.0.0.0/8    auto-summary
10.1.1.0/24
    [1] via 10.1.12.1, 00:00:15, GigabitEthernet0/0
10.1.12.0/24    directly connected, GigabitEthernet0/0
10.1.23.0/24    directly connected, GigabitEthernet1/0

```


Example 13-29 *Sample show ip route rip Command Output*

```
R2#show ip route rip
...output omitted...
    10.0.0.0/8 is variably subnetted, 6 subnets, 2 masks
R       10.1.1.0/24 [120/1] via 10.1.12.1, 00:00:11, GigabitEthernet0/0
```

Example 13-30 *Sample debug ip rip Command Output*

```
R2#debug ip rip
RIP protocol debugging is on
R3#
RIP: received v2 update from 10.1.23.3 on GigabitEthernet1/0
    10.1.3.0/24 via 0.0.0.0 in 1 hops
R3#u all
All possible debugging has been turned off
```

**Figure 13-7** *Sample RIP topology*

On R2, you issue the **show ip route 10.1.3.0** command, as shown in Example 13-31. The output displays that the route is learned via a static route with an AD of 1. Therefore, it is more believable than the RIP-learned route from R3 and the reason why it is not in the table as a RIP learned route.

Example 13-31 *Verifying the Source of 10.1.3.0 Route*

```
R2#show ip route 10.1.3.0
Routing entry for 10.1.3.0/24
  Known via "static", distance 1, metric 0
  Routing Descriptor Blocks:
    * 10.1.23.3
      Route metric is 0, traffic share count is 1
```

So, what is the issue? Because 10.1.3.0 is not being used by R2 as a RIP-learned route, it will not advertise it to R1. Therefore, R1 will not know how to reach 10.1.3.0/24. Examining R1's routing table in Example 13-32 confirms this for us.

Example 13-32 *Verifying R1's Routing Table*

```
R2#show ip route
...output omitted...
    10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
C       10.1.1.0/24 is directly connected, GigabitEthernet0/0
L       10.1.1.1/32 is directly connected, GigabitEthernet0/0
C       10.1.12.0/24 is directly connected, GigabitEthernet1/0
L       10.1.12.1/32 is directly connected, GigabitEthernet1/0
R       10.1.23.0/24 [120/1] via 10.1.12.2, 00:00:19, GigabitEthernet1/0
```

ACLs



RIP uses destination UDP port 520 for communication and the destination multicast address 224.0.0.9. Therefore, if there is an ACL filtering traffic in an interface and it is not permitting User Datagram Protocol (UDP) port 520 traffic or the multicast address 224.0.0.9, RIP routing updates will be denied in the interface. Example 13-33 displays access list 100 applied to interface Gigabit Ethernet 1/0 inbound. Notice that there is no entry permitting UDP port 520 traffic from R2 or all IP traffic (**ip any any**). Therefore, RIP packets will be denied because of the *implicit deny all* rule. The output of **show ip route** confirms that no RIP routes are learned or being used.

Example 13-33 *Verifying R1's Applied Access Lists and RIP Routes*

```
R1#show access-list
Extended IP access list 100
    10 permit ip 10.1.3.0 0.0.0.255 any
    20 permit ip 10.1.23.0 0.0.0.255 any
R1#show ip interface gigabitEthernet 1/0 | include access list
Outgoing access list is not set
Inbound access list is 100
R1#show ip route rip
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default, U - per-user static route
        o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
        + - replicated route, % - next hop override

Gateway of last resort is not set

R1#
```

To solve this issue, you need to add either **permit ip any any** at the end of the ACL or **permit udp any any eq 520** to allow the RIP packets in. However, if you want more control and security, you can specify the source address of the router you only want to receive RIPv2 packets from, which would be R2 in this case.

Load Sharing

By default, RIP will load balance on four equal metric paths. You can verify the maximum number of paths configured for load balancing with **show ip protocols**, as shown in Example 13-34. If you have equal metric paths but they are not being installed in the routing table for a particular destination network, check to make sure that the maximum paths is configured with an appropriate value for the number of paths you have. If it is set to 1, it means that no load balancing will occur.

Example 13-34 *Verifying Maximum Paths for Load Balancing*

```

R1#show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "rip"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Sending updates every 30 seconds, next due in 0 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Redistributing: rip
  Default version control: send version 2, receive version 2
    Interface          Send  Recv  Triggered RIP  Key-chain
  GigabitEthernet1/0    2     2
  Automatic network summarization is not in effect
  Maximum path: 4
  Routing for Networks:
    10.0.0.0
  Passive Interface(s):
    Ethernet0/0
    GigabitEthernet0/0
    GigabitEthernet2/0
  Routing Information Sources:
    Gateway         Distance    Last Update
    10.1.12.2         120        00:00:02
  Distance: (default is 120)

```

Other RIP Issues

In addition to all the reasons why RIP routes might be missing, you may have to troubleshoot issues related to a missing default route or to route summarization.

Missing Default Route

There is a very small chance that a router using RIP will be able to support all 480,000+ summarized Internet routes. Therefore, redistributing the entire BGP routing table into RIP is out of the question. Therefore, for packets sourced in the RIP domain destined to the Internet to be successfully routed, the RIP routers need to know what to do with packets that they do not have a specific match for. This is where the default route enters the picture.

The default route will typically be configured on the edge device with a next-hop address of the Internet service provider's (ISP) router (for example, **ip route 0.0.0.0 0.0.0.0 203.0.13.1**). However, this is a static default route on the edge router and still needs to be injected into the RIP process so that it can be advertised to the other RIP routers in the domain. To inject the default route into the RIP process, use the **default-information**

originate command in router RIP configuration mode. (Note that with RIP you do not need the static route configured to generate a default route. If it is missing, RIP will still generate a default route and advertise it to the other routers after you enter the **default-information originate** command. However, if the static route is missing on the router on which you issued the command, it will not be able to forward the packets and will drop them.)

If you expect a default route on routers in the RIP domain and they are not receiving one, verify the RIP configuration on the router that should be generating the default route (typically an edge router) with the **show run | section router rip** command, as shown in Example 13-35. In this case, you are looking for the **default-information originate** command, which is configured in this example.

Example 13-35 Verifying default-information originate Configuration

```
Edge#show run | section router rip
show run | section router rip
router rip
  version 2
  passive-interface default
  no passive-interface GigabitEthernet1/0
  network 10.0.0.0
  default-information originate
  no auto-summary
```

You may also want to confirm the default route has been inserted into the rip database as shown in Example 13-36 with **show ip rip database**. Notice that it has been inserted into the RIP database and that it says redistributed.

Example 13-36 Verifying the Default Route in the RIP Database

```
Edge#show ip rip database
0.0.0.0/0      auto-summary
0.0.0.0/0      redistributed
                [1] via 0.0.0.0,
10.0.0.0/8     auto-summary
10.1.1.0/24    directly connected, GigabitEthernet0/0
10.1.3.0/24
                [2] via 10.1.12.2, 00:00:19, GigabitEthernet1/0
10.1.12.0/24   directly connected, GigabitEthernet1/0
10.1.23.0/24
                [1] via 10.1.12.2, 00:00:19, GigabitEthernet1/0
```

Route Summarization

With RIP, manual route summarization is enabled on an interface-by-interface basis with the **ip summary address rip address ip_subnet_mask** interface configuration mode

command. Therefore, when troubleshooting route summarization, you want to keep the following in mind:

- Did you enable route summarization on the correct interface?
- Did you associate the summary route with RIP?
- Did you create the appropriate summary route?

You can verify manual route summarization using the **show ip protocols** command, as shown in Example 13-37. In this example, autosummarization is disabled, and manual summarization is enabled for RIP on interface Gigabit Ethernet 1/0 for 10.1.0.0/20.

Example 13-37 *Verifying Route Summarization with show ip protocols*

```
R1#show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "rip"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Sending updates every 30 seconds, next due in 11 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Redistributing: rip
  Default version control: send version 2, receive version 2
    Interface           Send  Recv  Triggered RIP  Key-chain
    GigabitEthernet1/0    2     2
  Automatic network summarization is not in effect
  Address Summarization:
    10.1.0.0/20 for GigabitEthernet1/0
  Maximum path: 4
  Routing for Networks:
    10.0.0.0
  Passive Interface(s):
    Ethernet0/0
    GigabitEthernet0/0
    GigabitEthernet2/0
  Routing Information Sources:
    Gateway         Distance      Last Update
    Gateway         Distance      Last Update
    10.1.12.2         120          00:00:17
  Distance: (default is 120)
```

It is important to remember that a route to null0 is not automatically created with RIP. Therefore, if R1, as shown in Figure 13-8, is configured with a summary route, as shown previously in Example 13-37, and it has a default route in the routing table, as shown in Example 13-38, a routing loop may occur, as shown in the following paragraphs and examples.



Figure 13-8 *Route Summarization Topology*

Example 13-38 *Verifying the Default Route on R1*

```
R1#show ip route
...output omitted...
Gateway of last resort is 10.1.12.2 to network 0.0.0.0

R*    0.0.0.0/0 [120/2] via 10.1.12.2, 00:00:14, GigabitEthernet1/0
      10.0.0.0/8 is variably subnetted, 6 subnets, 2 masks
C      10.1.1.0/24 is directly connected, GigabitEthernet0/0
L      10.1.1.1/32 is directly connected, GigabitEthernet0/0
R      10.1.3.0/24 [120/2] via 10.1.12.2, 00:00:14, GigabitEthernet1/0
C      10.1.12.0/24 is directly connected, GigabitEthernet1/0
L      10.1.12.1/32 is directly connected, GigabitEthernet1/0
R      10.1.23.0/24 [120/1] via 10.1.12.2, 00:00:14, GigabitEthernet1/0
```

Now suppose that R1 receives a packet destined for 10.1.5.2. What will it do with it? Based on the default route, it will send it to R2. However, in Example 13-37, you witnessed in the output of **show ip protocols** that a summary route for 10.1.0.0/20 was configured on R1. Using the command **show ip rip database**, as shown in Example 13-39, indicates 10.1.0.0/20 is installed in the RIP database. Also, reviewing the output of **show ip route** in Example 13-40 on R2 reveals that it is learning the summary route from R1.

Example 13-39 *Verifying the Summary Route in the RIP Database*

```
R1#show ip rip database
0.0.0.0/0      auto-summary
0.0.0.0/0
    [2] via 10.1.12.2, 00:00:27, GigabitEthernet1/0
10.0.0.0/8      auto-summary
10.1.0.0/20     int-summary
10.1.1.0/24     directly connected, GigabitEthernet0/0
10.1.3.0/24
    [2] via 10.1.12.2, 00:00:27, GigabitEthernet1/0
10.1.12.0/24    directly connected, GigabitEthernet1/0
10.1.23.0/24
    [1] via 10.1.12.2, 00:00:27, GigabitEthernet1/0
```

Example 13-40 *Verifying the Summary Route on R2*

```
R2#show ip route
...output omitted...
Gateway of last resort is 10.1.23.3 to network 0.0.0.0
```

```

R*    0.0.0.0/0 [120/1] via 10.1.23.3, 00:00:20, GigabitEthernet1/0
      10.0.0.0/8 is variably subnetted, 7 subnets, 3 masks
R     10.1.0.0/20 [120/1] via 10.1.12.1, 00:00:21, GigabitEthernet0/0
R     10.1.1.0/24 [120/1] via 10.1.12.1, 00:01:16, GigabitEthernet0/0
R     10.1.3.0/24 [120/1] via 10.1.23.3, 00:00:20, GigabitEthernet1/0
C     10.1.12.0/24 is directly connected, GigabitEthernet0/0
L     10.1.12.2/32 is directly connected, GigabitEthernet0/0
C     10.1.23.0/24 is directly connected, GigabitEthernet1/0
L     10.1.23.2/32 is directly connected, GigabitEthernet1/0

```

Because the advertised summary route includes networks that R1 truly does not know how to reach (including 10.1.5.2), a routing loop is created as packets are sent to R1, because of the summary route, and then R1 sends the packet back to R2, because of the default route. For example, R2 sends the packet destined to 10.1.5.2 back to R1, then R1 sends it back to R2, and then R2 back to R1, and so on. This is witnessed in Example 13-41 with a trace from R3 to 10.1.5.2, which loops between R2 and R1.

Example 13-41 *Verifying a Routing Loop with a Trace on R3*

```

R3#trace 10.1.5.2
Type escape sequence to abort.
Tracing the route to 10.1.5.2
VRF info: (vrf in name/id, vrf out name/id)
  1 10.1.23.2 32 msec 44 msec 44 msec
  2 10.1.12.1 72 msec 88 msec 84 msec
  3 10.1.12.2 88 msec 92 msec 88 msec
  4 10.1.12.1 152 msec 120 msec 136 msec
  5 10.1.12.2 132 msec 128 msec 116 msec
  6 10.1.12.1 136 msec 136 msec 124 msec
  7 10.1.12.2 160 msec 136 msec *
  8 10.1.12.1 156 msec 176 msec 180 msec
  9 10.1.12.2 180 msec 168 msec 200 msec
10 10.1.12.1 212 msec 192 msec 196 msec
11 10.1.12.2 232 msec 196 msec 208 msec
12 10.1.12.1 244 msec 260 msec 236 msec
13 10.1.12.2 228 msec 264 msec 228 msec
14 10.1.12.1 276 msec 296 msec 288 msec
15 10.1.12.2 260 msec 276 msec 292 msec
16 10.1.12.1 292 msec 316 msec 276 msec
17 10.1.12.2 288 msec 316 msec 268 msec
18 10.1.12.1 332 msec 312 msec 368 msec
19 10.1.12.2 308 msec 336 msec 324 msec
20 10.1.12.1 364 msec 340 msec 364 msec
21 10.1.12.2 372 msec 340 msec 372 msec
22 10.1.12.1 372 msec 384 msec 372 msec
23 10.1.12.2 416 msec 384 msec 404 msec
24 10.1.12.1 428 msec 416 msec 424 msec

```

```

25 10.1.12.2 420 msec 432 msec 440 msec
26 10.1.12.1 456 msec 460 msec 452 msec
27 10.1.12.2 460 msec 484 msec 460 msec
28 10.1.12.1 468 msec 484 msec 492 msec
29 10.1.12.2 488 msec 508 msec 480 msec
30 10.1.12.1 528 msec 516 msec 548 msec

```

To solve this issue, you would need to create a static route to null0 on R1 for the summary route or create a better summary route. This will ensure that when R1 receives a packet that falls within the summary route but that it does not know how to reach, it will drop the packet instead of sending it back to R2 because of the default route. Example 13-42 displays the static route configuration to null0 for the same network as the summary route and the output of **show ip route** to verify the newly created static route.

Example 13-42 *Configuring a Static Route to Null0*

```

R1#config t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#ip route 10.1.0.0 255.255.240.0 null 0
R1(config)#end
R1#show ip route
...output omitted...
Gateway of last resort is 10.1.12.2 to network 0.0.0.0

R*    0.0.0.0/0 [120/2] via 10.1.12.2, 00:00:01, GigabitEthernet1/0
      10.0.0.0/8 is variably subnetted, 7 subnets, 3 masks
S     10.1.0.0/20 is directly connected, Null0
C     10.1.1.0/24 is directly connected, GigabitEthernet0/0
L     10.1.1.1/32 is directly connected, GigabitEthernet0/0
R     10.1.3.0/24 [120/2] via 10.1.12.2, 00:00:01, GigabitEthernet1/0
C     10.1.12.0/24 is directly connected, GigabitEthernet1/0
L     10.1.12.1/32 is directly connected, GigabitEthernet1/0
R     10.1.23.0/24 [120/1] via 10.1.12.2, 00:00:01, GigabitEthernet1/0

```

Example 13-43 confirms that the loop no longer exists with a trace. Notice the !H that is returned. It means the host is not reachable. In this case it is because, the packet is being dropped by the Null0 route.

Example 13-43 *Confirming That the Loop No Longer Exists with a Trace*

```

R3#trace 10.1.5.2
Type escape sequence to abort.
Tracing the route to 10.1.5.2
VRF info: (vrf in name/id, vrf out name/id)
  1 10.1.23.2 32 msec 44 msec 40 msec
  2 10.1.12.1 68 msec 92 msec 72 msec
  3 10.1.12.1 !H !H !H

```


Troubleshooting RIPng

RIPng is the next generation RIP routing protocol designed for routing IPv6 addresses and prefixes. It functions the same as RIPv2, but had to be modified to support IPv6. Some of the RIPng enhancements include the use of the all-RIP-devices multicast group of FF02::9, the removal of the **network** command, which was replaced by an interface configuration command, and the use of link-local addresses as the next-hop IPv6 address.

In this section, you will learn how to troubleshoot and verify RIPng issues.

Before you even begin troubleshooting RIPng, you need to verify that IPv6 unicast routing is enabled on the router. If it is not, it must be enabled before you proceed any further. As shown in Example 13-44, you can use the **show run | include ipv6 unicast-routing** command to verify if it is enabled.

Example 13-44 *Confirming That IPv6 Unicast Routing Is Enabled*

```
R1#show run | include ipv6 unicast-routing
ipv6 unicast-routing
R1#
```

Figure 13-9 depicts a RIPng routing domain. To verify the RIPng database on R1, you use the **show ipv6 rip database** command, as shown in Example 13-45. In this example, you can see that 2001:DB8:0:3::/64 and 2001:DB8:0:23::/64 are installed in the routing table but that 2001:DB8:0:12::/64 is not. This is because 2001:DB8:0:3::/64 and 2001:DB8:0:23::/64 have been learned from a neighboring router, and 2001:DB8:0:12::/64 is a directly connected network. Therefore, there is a better route based on AD that can be installed in the routing table instead of this RIPng one.

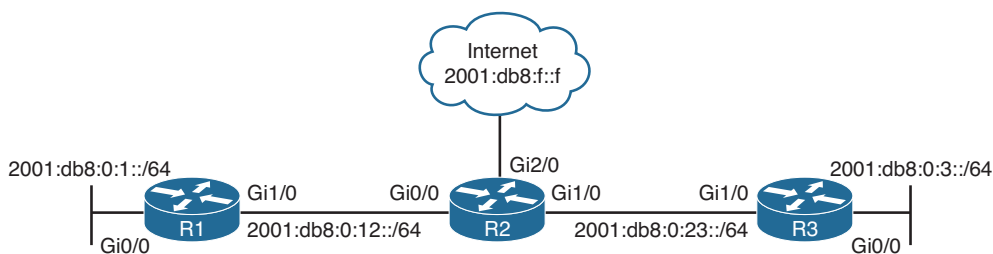


Figure 13-9 RIPng Sample Topology

Example 13-45 *Sample Output of the RIPng Database*

```
R1#show ipv6 rip database
RIP process "TSHOOT_RIP", local RIB
 2001:DB8:0:3::/64, metric 3, installed
   GigabitEthernet1/0/FE80::C801:3FF:FE9C:8, expires in 172 secs
 2001:DB8:0:12::/64, metric 2
   GigabitEthernet1/0/FE80::C801:3FF:FE9C:8, expires in 172 secs
 2001:DB8:0:23::/64, metric 2, installed
   GigabitEthernet1/0/FE80::C801:3FF:FE9C:8, expires in 172 secs
```

Key
Topic

To verify the RIPv6 routes installed in the routing table, you use the **show ipv6 route rip** command, as shown in Example 13-46. Notice that the RIPv6 routes are still represented by an R and that the AD is still 120. In addition, the hop count is still based on the number of routers that have to be traversed to reach the destination network. Note that the next-hop IP address is an FE80:: link-local address.

Example 13-46 Viewing RIPv6 Routes in the Routing Table

```
R1#show ipv6 route rip
IPv6 Routing Table - default - 9 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, R - RIP, H - NHRP, I1 - ISIS L1
       I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary, D - EIGRP
       EX - EIGRP external, ND - ND Default, NDp - ND Prefix, DCE - Destination
       NDr - Redirect, O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1
       OE2 - OSPF ext 2, ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2, l - LISP
R   2001:DB8:0:3::/64 [120/3]
    via FE80::C801:3FF:FE9C:8, GigabitEthernet1/0
R   2001:DB8:0:23::/64 [120/2]
    via FE80::C801:3FF:FE9C:8, GigabitEthernet1/0
```



The **show ipv6 protocols** output, as shown in Example 13-47, is not as verbose as the **show ip protocols** output. Presently, it is only showing the interfaces that are enabled for the RIPv6 process called TSHOOT_RIP and that no redistribution is happening. If you want to verify timers, maximum paths, port number, multicast group, as well as the interfaces that are enabled for the RIPv6 process, you need to use the **show ipv6 rip process_name** command. In Example 13-48, the output of **show ipv6 rip TSHOOT_RIP** is displayed. You can verify that the maximum paths is set to 16, the multicast group is FF02::9, the RIPv6 port number is 521, the AD is 120, split horizon is on, updates are sent every 30 seconds and will expire after 180 seconds, and interfaces Gigabit Ethernet 0/0 and 1/0 are enabled for this RIPv6 process.

Example 13-47 Viewing the Output of show ipv6 protocols

```
R1#show ipv6 protocols
IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "ND"
IPv6 Routing Protocol is "rip TSHOOT_RIP"
  Interfaces:
    GigabitEthernet1/0
    GigabitEthernet0/0
  Redistribution:
    None
```

Example 13-48 Viewing the Output of show ipv6 rip TSHOOT_RIP

```
R1#show ipv6 rip TSHOOT_RIP
RIP process "TSHOOT_RIP", port 521, multicast-group FF02::9, pid 93
  Administrative distance is 120. Maximum paths is 16
```

```

Updates every 30 seconds, expire after 180
Holddown lasts 0 seconds, garbage collect after 120
Split horizon is on; poison reverse is off
Default routes are not generated
Periodic updates 79, trigger updates 8
Full Advertisement 0, Delayed Events 0
Interfaces:
  GigabitEthernet1/0
  GigabitEthernet0/0
Redistribution:
  None

```

If you want to verify the number of routes that a router is learning from a RIPng directly connected router, you can use the **show ipv6 rip next-hops** command, as shown in Example 13-49. In this case, we can reach three different networks (we have learned about three different networks) from the RIPng router at the next-hop IPv6 address of FE80::C801:3FF:FE9C:8.

Example 13-49 Viewing the Number of IPv6 Routes Reachable via a Next-Hop Router

```

R1#show ipv6 rip next-hops
RIP process "TSHOOT_RIP", Next Hops
  FE80::C801:3FF:FE9C:8/GigabitEthernet1/0 [3 paths]

```

If you need to verify RIPng packets in real time, you can use the **debug ipv6 rip** command, as shown in Example 13-50. In this example, you can see the router sending RIPng updates out Gig1/0 for RIPng process TSHOOT_RIP with a link-local source address and a multicast destination of FF02::9. The prefixes are 2001:DB8:0:1::/64 and 2001:DB8:0:12::/64. The router is also receiving RIPng updates on the same interface from the device with a link-local address of FE80::C80F:1FF:FE9C:8. The routes in the update are 2001:DB8:0:12::/64, 2001:DB8:0:23::/64, and 2001:DB8:0:3::/64.

Example 13-50 Sample RIPng debug Output

```

R1#debug ipv6 rip
RIP Routing Protocol debugging is on
R1#
RIPng: Sending multicast update on GigabitEthernet1/0 for TSHOOT_RIP
  src=FE80::C80E:1FF:FE9C:1C
  dst=FF02::9 (GigabitEthernet1/0)
  sport=521, dport=521, length=52
  command=2, version=1, mbz=0, #rte=2
  tag=0, metric=1, prefix=2001:DB8:0:1::/64
  tag=0, metric=1, prefix=2001:DB8:0:12::/64
RIPng: Packet waiting
RIPng: response received from FE80::C80F:1FF:FE9C:8 on GigabitEthernet1/0 for
TSHOOT_RIP
  src=FE80::C80F:1FF:FE9C:8 (GigabitEthernet1/0)

```

```

dst=FF02::9
sport=521, dport=521, length=72
command=2, version=1, mbz=0, #rte=3
tag=0, metric=1, prefix=2001:DB8:0:12::/64
tag=0, metric=1, prefix=2001:DB8:0:23::/64
tag=0, metric=2, prefix=2001:DB8:0:3::/64
R1#u all
All possible debugging has been turned off

```

By default, RIPv6 will load balance on 16 equal metric paths. You can verify the maximum number of paths configured for load balancing with the **show ipv6 rip process_name** command, as shown in Example 13-51. In this case, the maximum paths would have been changed to 11 with the **maximum-paths** command in **ipv6 router rip process_name** configuration mode. If you have equal metric paths but they are not being installed in the routing table for a particular destination network, check to make sure that the maximum paths is configured with an appropriate value for the number of paths you have. If it is set to 1, it means that no load balancing will occur.

Key Topic

Example 13-51 Verifying Maximum Paths for Load Balancing

```

R1#show ipv6 rip TSHOOT_RIP
RIP process "TSHOOT_RIP", port 521, multicast-group FF02::9, pid 276
  Administrative distance is 120. Maximum paths is 11
  Updates every 30 seconds, expire after 180
  Holddown lasts 0 seconds, garbage collect after 120
  Split horizon is on; poison reverse is off
  Default routes are not generated
  Periodic updates 71, trigger updates 2
  Full Advertisement 1, Delayed Events 0
Interfaces:
  GigabitEthernet1/0
  GigabitEthernet0/0
Redistribution:
  None

```

Key Topic

With RIPv6, default routing is enabled on an interface-by-interface basis with the **ipv6 rip process_name default-information [originate | only]** command. The **originate** keyword is used to advertise a default route out the interface along with all the other routes that the router knows. The **only** keyword is used to advertise just a default route, and all other routes that would have been advertised out the interface are suppressed. You can verify whether a default route is being generated by using the **show ipv6 rip process_name** command, as shown in Example 13-52. However, this does not confirm what type of default route is being generated or by which interface. You must review the interface configuration in the running configuration to verify this. Using the command **show run | include interfacedefault**, as shown in Example 13-53, displays that interface Gig1/0 is configured to generate a default route and only advertise that specific route out Gig1/0.

Example 13-52 *Verifying Whether a Default Route Is Being Generated*

```

R1#show ipv6 rip TSHOOT_RIP
RIP process "TSHOOT_RIP", port 521, multicast-group FF02::9, pid 276
    Administrative distance is 120. Maximum paths is 11
    Updates every 30 seconds, expire after 180
    Holddown lasts 0 seconds, garbage collect after 120
    Split horizon is on; poison reverse is off
    Default routes are generated
    Periodic updates 110, trigger updates 2
    Full Advertisement 1, Delayed Events 0
Interfaces:
    GigabitEthernet1/0
    GigabitEthernet0/0
Redistribution:
    None

```

Example 13-53 *Verifying Whether a Default Route Is Configured on an Interface*

```

R1#show run | include interface|default
interface Ethernet0/0
interface GigabitEthernet0/0
interface GigabitEthernet1/0
    ipv6 rip TSHOOT_RIP default-information only
interface GigabitEthernet2/0

```



ACLs can be the cause of many troubleshooting efforts. You implement an ACL on an interface to protect your network from malicious traffic only to break routing in your network because you accidentally denied the routing protocol with the implicit deny all entry in the ACL. To verify whether there are any ACLs denying packets in an interface, you can issue the **debug ipv6 packets** command. The **debug** messages will indicate that the packet is being discarded by a certain ACL. However, be very careful with this command because it debugs every single IPv6 packet and could overload the router's processor and grind it to a halt.

An alternative is to issue the command **show run | include interface|traffic** to find any interface that might have the **ipv6 traffic-filter** command applied, as in Example 13-54. In this example, you see the IPv6 traffic filter called NETWORK is applied inbound to Gig1/0. Now you issue the **show ipv6 access-list NETWORK** command to confirm whether this is in fact the reason why routes are not being learned. In Example 13-55, the only traffic that is permitted is traffic from 2001:DB8:0:3::/64 going to 2001:DB8:0:1::/64. Therefore, all other IPV6 traffic, except neighbor discovery traffic, is denied by the implicit deny all entry. Even the RIPng updates are denied. You will need to add an entry that permits UDP port 521 traffic for the RIPng updates (for example, **permit udp any any eq 521**).

Example 13-54 *Verifying ACLs Applied to Interfaces*

```
R1#show run | include interface|traffic
interface Ethernet0/0
interface GigabitEthernet0/0
interface GigabitEthernet1/0
  ipv6 traffic-filter NETWORK in
interface GigabitEthernet2/0
```

Example 13-55 *Verifying IPv6 ACLs*

```
R1#show ipv6 access-list NETWORK
IPv6 access list NETWORK
permit ipv6 2001:DB8:0:3::/64 2001:DB8:0:1::/64 sequence 10
```



RIPv6 is enabled on an interface-by-interface basis with the **ipv6 rip process_name enable** interface configuration command. If the command is missing from the interface, the RIPv6 process will not be running on the interface, and the network the interface is part of will not be injected into the RIPv6 routing process. To verify which interfaces are participating in a particular RIPv6 process, you can use the **show ipv6 protocols** command or the **show ipv6 rip process_name** command, as shown in Example 13-56.

Example 13-56 *Verifying RIPv6-Enabled Interfaces*

```
R1#show ipv6 protocols
IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "ND"
IPv6 Routing Protocol is "rip TSHOOT_RIP"
  Interfaces:
    GigabitEthernet1/0
    GigabitEthernet0/0
  Redistribution:
    None
R1#show ipv6 rip TSHOOT_RIP
RIP process "TSHOOT_RIP", port 521, multicast-group FF02::9, pid 93
  Administrative distance is 120. Maximum paths is 16
  Updates every 30 seconds, expire after 180
  Holddown lasts 0 seconds, garbage collect after 120
  Split horizon is on; poison reverse is off
  Default routes are not generated
  Periodic updates 79, trigger updates 8
  Full Advertisement 0, Delayed Events 0
  Interfaces:
    GigabitEthernet1/0
    GigabitEthernet0/0
  Redistribution:
    None
```

RIPv2 and RIPvng Trouble Tickets

This section presents various trouble tickets relating to the topics discussed earlier in the chapter. The purpose of these trouble tickets is to give a process that you can follow when troubleshooting in the real world or in an exam environment. All trouble tickets in this section are based on the topology depicted in Figure 13-10.

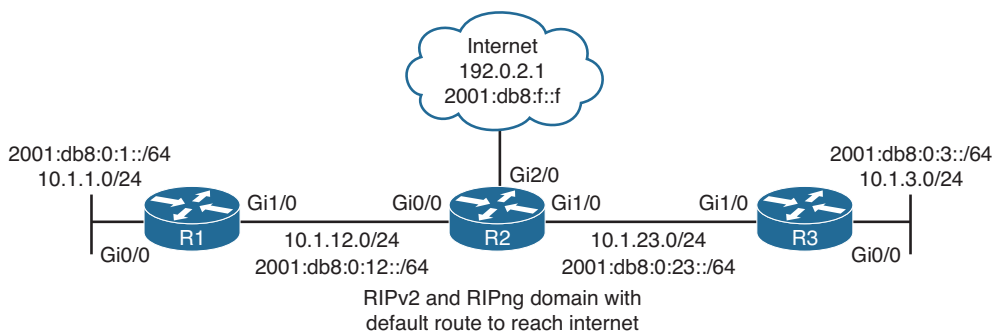


Figure 13-10 RIPv2 and RIPvng Trouble Ticket Topology

Trouble Ticket 13-1

Problem: Users in the 10.1.1.0/24 network indicate that they are not able to access resources in the 10.1.3.0/24 network.

You start your troubleshooting process by confirming the issue. From a PC in the 10.1.1.0/24 network, you ping to the Gig0/0 interface on R3, which has an IP address of 10.1.3.3. In Example 13-57, the ping fails, confirming the issue. A ping to the default gateway of 10.1.1.0/24, which is 10.1.1.1, is successful, indicating that the issue is beyond the LAN and that we can start our troubleshooting efforts on R1.

Example 13-57 Confirming Issue with Ping

```
C:\>ping 10.1.3.3

Pinging 10.1.3.3 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.1.3.3:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\>ping 10.1.1.1

Pinging 10.1.1.1 with 32 bytes of data:
```

```

Reply from 10.1.1.1: bytes=32 time 1ms TTL=128
Reply from 10.1.1.1: bytes=32 time 1ms TTL=128
Reply from 10.1.1.1: bytes=32 time 1ms TTL=128
Reply from 10.1.1.1: bytes=32 time 1ms TTL=128

Ping statistics for 10.1.1.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

```

On R1, you issue the **show ip route 10.1.3.3** command. As shown in Example 13-58, the subnet is not in the table. This is a great indication that we are not learning about the 10.1.3.3 network. However, your network does use a default route; therefore, you issue the **show ip route 0.0.0.0** command and notice that there is a default route, as shown in Example 13-59, that points to a next hop of 10.1.12.2, which is R2. As a result, R1 should be sending the traffic to R2.

Example 13-58 *Confirming Route to 10.1.3.3*

```

R1#show ip route 10.1.3.3
% Subnet not in table

```

Example 13-59 *Reviewing the Default Route in the Routing Table*

```

R1#show ip route 0.0.0.0
Routing entry for 0.0.0.0/0, supernet
    Known via "rip", distance 120, metric 1, candidate default path
    Redistributing via rip
    Last update from 10.1.12.2 on GigabitEthernet1/0, 00:00:21 ago
    Routing Descriptor Blocks:
    * 10.1.12.2, from 10.1.12.2, 00:00:21 ago, via GigabitEthernet1/0
        Route metric is 1, traffic share count is 1

```

You issue a **traceroute** command to 10.1.3.3 on R1, sourcing it from 10.1.1.1, as shown in Example 13-60, and notice that the packet is going to the Internet at R2. It is time to shift your attention to R2.

Example 13-60 *Using the traceroute Command to Verify the Path*

```

R1#traceroute 10.1.3.3 source 10.1.1.1
Type escape sequence to abort.
Tracing the route to 10.1.3.3
VRF info: (vrf in name/id, vrf out name/id)
  1 10.1.12.2 32 msec 44 msec 36 msec
  2 203.0.113.2 64 msec 40 msec 60 msec
  3 * * *
...output omitted...

```


On R2, you issue the command **show ip route 10.1.3.3**, and the result is the same as R1, subnet not in table, as shown in Example 13-61. It appears that R2 might not be learning about the 10.1.3.0/24 network. You issue the command **show ip rip database** on R2, as shown in Example 13-62, and confirm that R2 is not learning about 10.1.3.0/24 from R3.

Example 13-61 *Confirming Route in R2's Routing Table*

```
R2#show ip route 10.1.3.3
% Subnet not in table
```

Example 13-62 *Viewing the RIP Database on R2*

```
R2#show ip rip database
0.0.0.0/0      auto-summary
0.0.0.0/0      redistributed
    [1] via 0.0.0.0,
10.0.0.0/8     auto-summary
10.1.1.0/24
    [1] via 10.1.12.1, 00:00:14, GigabitEthernet0/0
10.1.12.0/24   directly connected, GigabitEthernet0/0
10.1.23.0/24   directly connected, GigabitEthernet1/0
```

You hypothesize that interface Gig0/0 on R3 is not participating in the RIP process. To verify your hypothesis, you issue the **show ip protocols** command on R3, as shown in Example 13-63. According to the output, interface Gig0/0 is participating in the RIP process, and as a result the network associated with the interface should be advertised. You then notice that interface Gig1/0, which is connected to R2, is not participating in the RIP process, because it is not listed in the interface listing. However, you notice that it states lower in the output, Routing for Networks: 10.0.0.0. Interface Gig1/0 has an IP address of 10.1.23.3, as shown in the output of **show ip interface brief** of Example 13-64. Therefore, it is enabled for the RIP process and should be advertising the RIP updates to R2. You then notice that Gig1/0 is configured as a passive interface. When configured as a passive interface, the interface will still receive RIP updates but not send RIP updates. Therefore, R2 is not learning about 10.1.3.0/24, and as a result, R1 does not learn about it.

Example 13-63 *Viewing the Output of show ip protocols on R3*

```
R3#show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "rip"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Sending updates every 30 seconds, next due in 27 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Redistributing: rip
  Default version control: send version 2, receive version 2
    Interface          Send Recv  Triggered RIP  Key-chain
```

```
GigabitEthernet0/0 2 2
Automatic network summarization is in effect
Maximum path: 4
Routing for Networks:
  10.0.0.0
Passive Interface(s):
  GigabitEthernet1/0
Routing Information Sources:
  Gateway          Distance      Last Update
  10.1.23.2         120          00:00:06
Distance: (default is 120)
```

Example 13-64 Viewing the Output of show ip interface brief on R3

```
R3#show ip int brief
```

Interface	IP-Address	OK?	Method	Status	Protocol
Ethernet0/0	unassigned	YES	NVRAM	administratively down	down
GigabitEthernet0/0	10.1.3.3	YES	NVRAM	up	up
GigabitEthernet1/0	10.1.23.3	YES	NVRAM	up	up

You issue the command **show run | section router rip** and confirm that the **passive-interface** command is configured for interface Gig1/0, as shown in Example 13-65. You remove the command with the **no passive-interface GigabitEthernet1/0** command in router RIP configuration mode.

Example 13-65 Verifying the Passive-Interface Configuration on R3

```
R3#show run | section router rip
router rip
version 2
passive-interface GigabitEthernet1/0
network 10.0.0.0
R3#config t
Enter configuration commands, one per line. End with CNTL/Z.
R3(config)#router rip
R3(config-router)#no passive-interface GigabitEthernet1/0
R3(config-router)#end
R3#
```

Now when you go back to R1 and issue the command **show ip route 10.1.3.3**, 10.1.3.0/24 is listed as the entry in the routing table, and the ping to 10.1.3.3 is successful, as shown in Example 13-66.

Example 13-66 Verifying 10.1.3.0/24 Route in R1's Routing Table and a Successful Ping

```
R1#show ip route 10.1.3.3
Routing entry for 10.1.3.0/24
  Known via "rip", distance 120, metric 2
  Redistributing via rip
```

```

Last update from 10.1.12.2 on GigabitEthernet1/0, 00:00:27 ago
Routing Descriptor Blocks:
* 10.1.12.2, from 10.1.12.2, 00:00:27 ago, via GigabitEthernet1/0
    Route metric is 2, traffic share count is 1
R1#ping 10.1.3.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.3.3, timeout is 2 seconds:
!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 36/52/64 ms

```

Trouble Ticket 13-2

Problem: Users in the 2001:db8:0:1::/64 network indicate that they are not able to access any resources on the Internet or the 2001:db8:0:3::/64 network.

You begin troubleshooting by verifying the problem with a ping to the Internet address of 2001:db8:f::f and the router address 2001:db8:0:3::3 with a source of 2001:db8:0:1::1. The results in Example 13-67 confirm the issues.

Example 13-67 *Confirming the Users' Issues with Pings*

```

R1#ping 2001:db8:f::f source 2001:db8:0:1::1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:F::F, timeout is 2 seconds:
Packet sent with a source address of 2001:DB8:0:1::1
.....
Success rate is 0 percent (0/5)
R1#ping 2001:db8:0:3::3 source 2001:db8:0:1::1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:0:3::3, timeout is 2 seconds:
Packet sent with a source address of 2001:DB8:0:1::1
.....
Success rate is 0 percent (0/5)

```

You then issue the **show ipv6 route ip_address** command on R1 to determine whether a route exists. In Example 13-68, the default route is used to reach the Internet address, and 2001:DB8:0:3::/64 is used to reach 2001:DB8:0:3::3. Therefore, R1 knows how to reach the networks. In both cases, the next hop is out Gig1/0 with a link-local address of FE80::C80F:1FF:FE9C:8 (R2).

Example 13-68 *Verifying Routing Table Entries for Destination Networks*

```

R1#show ipv6 route 2001:db8:f::f
Routing entry for ::/0
  Known via "rip TSHOOT_RIP", distance 120, metric 2
  Route count is 1/1, share count 0
  Routing paths:
    FE80::C80F:1FF:FE9C:8, GigabitEthernet1/0
    Last updated 01:16:36 ago

```

```

R1#show ipv6 route 2001:db8:0:3::3
Routing entry for 2001:DB8:0:3::/64
  Known via "rip TSHOOT_RIP", distance 120, metric 3
  Route count is 1/1, share count 0
  Routing paths:
    FE80::C80F:1FF:FE9C:8, GigabitEthernet1/0
    Last updated 12:31:48 ago

```

You issue an extended IPv6 traceroute on R1 to verify the path from the 2001:db8:0:1::/64 network, and it fails immediately, as shown in Example 13-69.

Example 13-69 Issuing a Traceroute to Verify a Path

```

R1#traceroute ipv6
Target IPv6 address: 2001:db8:f::f
Source address: 2001:db8:0:1::1
Insert source routing header? [no]:
Numeric display? [no]:
Timeout in seconds [3]:
Probe count [3]:
Minimum Time to Live [1]:
Maximum Time to Live [30]:
Priority [0]:
Port Number [0]:
Type escape sequence to abort.
Tracing the route to 2001:DB8:F::F

 1  *  *  *
 2  *  *  *
 3  *  *  *
 4  *  *  *
 5  *  *  *
 6  *  *  *
 7  *  *  *
 8  *  *  *
 9  *  *  *
...output omitted...

```

You issue a standard trace from R1 to 2001:db8:f::f so that the trace is generated with a source address of 2001:db8:0:12::1 and notice that it fails at R2, as shown in Example 13-70. However, we can see the !A, which indicates that an ACL is possibly the culprit here. This gives us something to work with now as we shift our attention to R2. However, notice that based on the source address, we had different trace results, ***, or !A !A !A.

Example 13-70 *Issuing a Traceroute to Verify a Path*

```
R1#traceroute 2001:db8:f::f
Type escape sequence to abort.
Tracing the route to 2001:DB8:F::F

 1 2001:DB8:0:12::2 !A !A !A
```

On R2, you issue the **show run | include interface|traffic-filter** command to see whether any interfaces have an IPv6 ACL applied to them. The result of this command, as shown in Example 13-71, indicates that Gig0/0 (which is the interface connected to R1) has an ACL applied named TSHOOT. Your next step is to review the IPv6 ACL named TSHOOT, as shown in Example 13-72, with the **show ipv6 access-list TSHOOT** command. The ACL is permitting all IPv6-related traffic from 2001:DB8:0:1::/64 to anywhere. This explains why the traces produced different results. The trace is allowed when sourced from 2001:DB8:0:1::/64 but not when sourced from 2001:DB8:0:12::/64. This still does not explain why R1 has the routes to reach the networks but connectivity is failing.

Example 13-71 *Verifying IPv6 ACLs Applied to Interfaces*

```
R2#show run | include interface|traffic-filter
interface Ethernet0/0
interface GigabitEthernet0/0
ipv6 traffic-filter TSHOOT in
interface GigabitEthernet1/0
interface GigabitEthernet2/0
```

Example 13-72 *Verifying IPv6 ACLs*

```
R2#show ipv6 access-list TSHOOT
IPv6 access list TSHOOT
    permit ipv6 2001:DB8:0:1::/64 any (74 matches) sequence 10
```

Let's review R2's routing table in Example 13-73. Maybe R2 has routes pointing to an incorrect next hop. They appear fine, so you decide to ping to make sure. As shown in the same example, they are successful.

Example 13-73 *R2's Routing Table and Successful Pings*

```
R2#show ipv6 route
...output omitted...
S   ::/0 [1/0]
    via 2001:DB8:0:A::A
R   2001:DB8:0:3::/64 [120/2]
    via FE80::C811:17FF:FE38:1C, GigabitEthernet1/0
C   2001:DB8:0:A::/64 [0/0]
    via GigabitEthernet2/0, directly connected
...output omitted...
R2#ping 2001:db8:f::f
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:F::F, timeout is 2 seconds:
!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/28/36 ms
R2#ping 2001:db8:0:3::3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:0:3::3, timeout is 2 seconds:
!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/40/88 ms
```

This narrows down the issue to being between R1 and R2. Recall that R1 knows about the routes. R2 knows about the routes. But, does R2 know how to reach R1 at 2001:db8:0:1::/64? You issue the command **show ipv6 route 2001:db8:0:1::/64**, as shown in Example 13-74, and notice that the default route is being used. Therefore, R2 does not know about 2001:db8:0:1::/64. Then you remember the ACL applied inbound on interface Gig0/0 of R2 from Example 13-72. It is permitting traffic from 2001:db8:0:1::/64 to anywhere, but it is denying everything else because of the implicit deny all rule, including RIPv6 updates.

Example 13-74 Verifying a Route to 2001:db8:0:1::/64 on R2

```
R2#show ipv6 route 2001:db8:0:1::/64
Routing entry for ::/0
  Known via "static", distance 1, metric 0
  Route count is 1/1, share count 0
  Routing paths:
    2001:DB8:0:A::A
    Last updated 01:51:27 ago
```

To fix this issue, you need to permit RIPv6 updates in the ACL as well. Example 13-75 displays how this can be accomplished in addition to how it can be verified. Notice that RIPv6 packets are being matched to the **permit** statement now.

Example 13-75 Permitting UDP Port 521 Traffic in an IPv6 ACL

```
R2#config t
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#ipv6 access-list TSHOOT
R2(config-ipv6-acl)#permit udp any any eq 521
R2(config-ipv6-acl)#end
R2#show ipv6 access-list TSHOOT
IPv6 access list TSHOOT
  permit ipv6 2001:DB8:0:1::/64 any (74 matches) sequence 10
  permit udp any any eq 521 (8 matches) sequence 20
```

Now when you reissue the command **show ipv6 route 2001:db8:0:1::/64**, as shown in Example 13-76, the route is correct.

Example 13-76 *Verifying Routing Table Entries*

```
R2#show ipv6 route 2001:db8:0:1::/64
Routing entry for 2001:DB8:0:1::/64
  Known via "rip TSHOOT_RIP", distance 120, metric 2
  Route count is 1/1, share count 0
  Routing paths:
    FE80::C80E:1FF:FE9C:1C, GigabitEthernet0/0
    Last updated 00:01:45 ago
```

Pinging from R1 to the Internet or the 2001:db8:0:3::/64 network is successful now, as shown in Example 13-77.

Example 13-77 *Successful Pings*

```
R1#ping 2001:db8:f::f source 2001:db8:0:1::1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:F::F, timeout is 2 seconds:
Packet sent with a source address of 2001:DB8:0:1::1
!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/33/48 ms
R1#ping 2001:db8:0:3::3 source 2001:db8:0:1::1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:0:3::3, timeout is 2 seconds:
Packet sent with a source address of 2001:DB8:0:1::1
!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/31/52 ms
```

Trouble Ticket 13-3

Problem: Users in the 2001:db8:0:1::/64 network indicate that they are not able to access resources on the Internet.

You begin troubleshooting by verifying the problem with a ping to the Internet address of 2001:db8:f::f with a source of 2001:db8:0:1::1. The results in Example 13-78 confirm the issues.

Example 13-77 *Confirming the Users' Issues with Pings*

```
R1#ping 2001:db8:f::f source 2001:db8:0:1::1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:F::F, timeout is 2 seconds:
Packet sent with a source address of 2001:DB8:0:1::1
.....
Success rate is 0 percent (0/5)
```

You then issue the `show ipv6 route ip_address` command on R1 to determine whether a route exists for 2001:db8:f::f. In Example 13-78, there is no entry for that network.

Based on the network topology in Figure 13-10, a default route should exist in the RIPv6 domain. Example 13-78 also displays the output of **show ipv6 route ::/0**, and the default route is not found either. On R2, you issue the same command, and Example 13-79 displays that it is known via a static route.

Example 13-78 *Verifying Routing Table Entries for Destination Networks*

```
R1#show ipv6 route 2001:db8:f::f
% Route not found
R1#show ipv6 route ::/0
% Route not found
```

Example 13-79 *Verifying Default Route on R2*

```
R2#show ipv6 route ::/0
Routing entry for ::/0
  Known via "static", distance 1, metric 0
  Route count is 1/1, share count 0
  Routing paths:
    2001:DB8:0:A::A
    Last updated 02:49:57 ago
```

Next you issue the **show ipv6 rip** command to verify whether a default route is being generated for RIPv6. In Example 13-80, you can clearly see that a default route is being generated.

Example 13-80 *Verifying a Default Route on R2*

```
R2#show ipv6 rip
RIP process "TSHOOT_RIP", port 521, multicast-group FF02::9, pid 276
  Administrative distance is 120. Maximum paths is 16
  Updates every 30 seconds, expire after 180
  Holddown lasts 0 seconds, garbage collect after 120
  Split horizon is on; poison reverse is off
  Default routes are generated
  Periodic updates 2089, trigger updates 8
  Full Advertisement 1, Delayed Events 0
Interfaces:
  GigabitEthernet0/0
  GigabitEthernet1/0
Redistribution:
  None
```

But wait; remember that default routes for RIPv6 are configured on an interface-by-interface basis. This only tells you that a default route is being generated. It does not tell you where. Therefore, you need to issue the **show run | include interface default** command, as shown in Example 13-81, to determine which RIPv6 interfaces are generating

a default route. In this case, only Gig1/0 is. You need to add the **ipv6 rip TSHOOT_RIP default-information originate** command to Gig0/0 as well. Example 13-82 displays the commands needed to fix this issue.

Example 13-81 *Verifying Which Interfaces Are Generating a Default Route for RIPng*

```
R2#show run | include interface|default
interface Ethernet0/0
interface GigabitEthernet0/0
interface GigabitEthernet1/0
ipv6 rip TSHOOT_RIP default-information originate
interface GigabitEthernet2/0
```

Example 13-82 *Configuring a RIPng Interface to Generate a Default Route*

```
R2#config t
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#interface gigabitEthernet 0/0
R2(config-if)#ipv6 rip TSHOOT_RIP default-information originate
R2(config-if)#end
R2#
```

Back on R1, you issue the **show ipv6 route ::/0** command again and confirm that there is an entry in the routing table. You also issue a ping to the Internet address 2001:db8:f::f, which is successful. Example 13-83 displays the routing table and the successful ping.

Example 13-83 *Verifying the Issue is Solved*

```
R1#show ipv6 route ::/0
Routing entry for ::/0
  Known via "rip TSHOOT_RIP", distance 120, metric 2
  Route count is 1/1, share count 0
  Routing paths:
    FE80::C80F:1FF:FE9C:8, GigabitEthernet1/0
    Last updated 00:03:51 ago

R1#ping 2001:db8:f::f source 2001:db8:0:1::1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:F::F, timeout is 2 seconds:
Packet sent with a source address of 2001:DB8:0:1::1
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 12/28/44 ms
R1#
```

Exam Preparation Tasks

As mentioned in the section “How to Use This Book” in the Introduction, you have a couple of choices for exam preparation: the exercises here; Chapter 22, “Final Preparation;” and the exam simulation questions on the CD-ROM.

Review All Key Topics

Review the most important topics in this chapter, noted with the Key Topic icon in the outer margin of the page. Table 13-2 lists a reference of these key topics and the page numbers on which each is found.



Table 13-2 *Key Topics for Chapter 13*

Key Topic Element	Description	Page Number
List	Outlines reasons why RIPv2 routes may be missing from the RIP database or the IPv4 routing table	466
Paragraph	Identifies how to verify the routes that are received from neighboring RIPv2 enabled routers with show ip rip database	467
Example 13-2	Viewing the RIP installed routes in the routing table with the show ip route rip command	467
Paragraph	Discusses how to verify various RIP parameters with show ip protocols command	468
Paragraph	Describes how to verify which interfaces are participating in the RIP process	470
Example 13-11	Verifying RIP passive interfaces with the show ip protocols command	472
List	Outlines reasons why the maximum hop count may be exceeded	475
List	Identifies what to consider when troubleshooting route filters and RIPv2	479
Section	Describes the split-horizon rule and how it affects RIPv2	480
Paragraph	Discusses how an ACL can affect RIP routing updates	485
Example 13-45	Sample output of the RIPv6 database	492
Paragraph	Describes how to verify various RIPv6 parameters with the show ipv6 rip command.	493
Example 13-51	Verifying maximum paths for load balancing	495

Key Topic Element	Description	Page Number
Paragraph	Describes how to troubleshoot default routing with RIPng	495
Paragraph	Discusses how an IPv6 ACL can affect RIPng routing updates	496
Paragraph	Describes how RIPng is enabled on an interface	497

Define Key Terms

Define the following key terms from this chapter and check your answers in the glossary:

RIPv2, RIPng, network command, passive interface, hop count, split horizon, auto-summarization, maximum paths

Command Reference to Check Your Memory

This section includes the most important **show** and **debug** commands covered in this chapter. It might not be necessary to memorize the complete syntax of every command, but you should be able to remember the basic keywords that are needed.

To test your memory of the commands, cover the right side of Table 13-3 with a piece of paper, read the description on the left side, and then see how much of the command you can remember.

The 300-135 TSHOOT exam focuses on practical, hands-on skills that are used by a networking professional. Therefore, you should be able to identify the commands needed to successfully troubleshoot the topics and concepts covered in this chapter.

Table 13-3 *show and debug commands*

Task	Command Syntax
Displays all RIPv1 and v2 routes learned from neighboring RIP-enabled routers.	show ip rip database
Displays the RIPv1 and v2 routes that have been installed in the routing table.	show ip route rip
Displays various parameters for the routing protocols running on the router. For RIPv1 and v2, it will display the following: outgoing and incoming update filters, timers, interfaces participating in the routing process, any key chains applied for authentication, the status of automatic summarization, the number of paths used for load balancing, passive interfaces, routers routes have been learned from, and AD.	show ip protocols

Task	Command Syntax
Displays the key chains configured on the router along with the key IDs and key strings. It will also identify when the key is valid.	show key chain
Displays various IPv4 parameters of an interface. For RIPv2, it helps identify whether split horizon is enabled, the multicast group the interface joined (224.0.0.9), and whether any ACLs are applied to an interface.	show ip interface <i>interface_type</i> <i>interface_number</i>
Displays all RIPv6 routes learned from neighboring RIPv6 enabled routers.	show ipv6 rip database
Displays the RIPv6 routes that have been installed in the routing table.	show ipv6 route rip
Displays various parameters for the RIPv6 routing processes running on the router. It displays the port used by RIPv6, the multicast group, AD, maximum paths, timers, split horizon, whether a default route is being generated, interfaces participating in the routing process, and redistribution.	show ipv6 rip
Displays the next-hop IPv6 addresses used to reach networks learned through RIPv6.	show ipv6 rip next-hops
Used to debug all RIPv2 and RIPv4 packets that are being sent and received by a router.	debug ip rip
Used to debug all RIPv6 packets that are being sent and received by a router.	debug ipv6 rip



This chapter covers the following topics:

- **Troubleshooting EIGRP for IPv4:** This section covers the reasons why EIGRP for IPv4 neighbor relationships are not being formed and how you can identify them. In addition, you will explore the reasons why EIGRP for IPv4 routes might be missing and how to determine why they are missing.
- **EIGRP for IPv4 Trouble Tickets:** This section provides trouble tickets that demonstrate how you can use a structured troubleshooting process to solve a reported problem.
- **Troubleshooting EIGRP for IPv6:** This section covers the reasons why EIGRP for IPv6 neighbor relationships are not being formed and how you can identify them. In addition, you will explore the reasons why EIGRP for IPv6 routes might be missing and how to determine why they are missing.
- **EIGRP for IPv6 Trouble Tickets:** This section provides trouble tickets that demonstrate how you can use a structured troubleshooting process to solve a reported problem.
- **Troubleshooting Named EIGRP Configurations:** In this section you discover the new **show** commands that you can use to troubleshoot named EIGRP configurations.
- **Named EIGRP Trouble Tickets:** This section provides trouble tickets that demonstrate how you can use a structured troubleshooting process to solve a reported problem.

Troubleshooting EIGRP

The Cisco Enhanced Interior Gateway Routing Protocol (EIGRP) is considered an advanced distance vector routing protocol. Specifically, EIGRP advertises routes to directly attached neighbors, like a distance vector routing protocol, while forming neighbor relationships, similar to a link-state routing protocol.

EIGRP can route for both IPv4 and IPv6 protocols. This chapter focuses on troubleshooting both of these protocols using the classic configurations and the newer named EIGRP configurations.

Before any routes can be exchanged between EIGRP routers on the same LAN or across a WAN, an EIGRP neighbor relationship has to be formed. There are many reasons why a neighbor relationship will not form, and as a troubleshooter, you need to be aware of them. This chapter dives deep into these issues and gives you the tools needed to identify them and successfully solve neighbor issues.

Once neighbor relationships are formed, neighboring routers exchange EIGRP routes. In various cases, routes may end up missing, and you need to be able to determine why the routes are missing. This chapter discusses the various ways that routes could go missing and how you can identify them and solve any route-related issue.

In this chapter, you also learn how to troubleshoot issues related to load balancing, summarization, discontinuous networks, and feasible successors.

“Do I Know This Already?” Quiz

The “Do I Know This Already?” quiz allows you to assess whether you should read this entire chapter thoroughly or jump to the “Exam Preparation Tasks” section. If you are in doubt about your answers to these questions or your own assessment of your knowledge of the topics, read the entire chapter. Table 14-1 lists the major headings in this chapter and their corresponding “Do I Know This Already?” quiz questions. You can find the answers in Appendix A, “Answers to the ‘Do I Know This Already?’ Quizzes.”

Table 14-1 “Do I Know This Already?” Section-to-Question Mapping

Foundation Topics Section	Questions
Troubleshooting EIGRP for IPv4	1–7
Troubleshooting EIGRP for IPv6	8–10
Troubleshooting named EIGRP Configurations	11

Caution The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you do not know the answer to a question or are only partially sure of the answer, you should mark that question as wrong for purposes of the self-assessment. Giving yourself credit for an answer that you correctly guess skews your self-assessment results and might provide you with a false sense of security.

1. Which command enables you to verify the routers that have formed an EIGRP adjacency with the local router, how long they have been neighbors for, and the current sequence number of EIGRP packets?
 - a. `show ip eigrp interfaces`
 - b. `show ip eigrp neighbors`
 - c. `show ip route eigrp`
 - d. `show ip protocols`
2. Which three of the following are reasons EIGRP neighbor relationships might not form?
 - a. Different autonomous system numbers
 - b. Different K values
 - c. Different timers
 - d. Different authentication parameters
3. Which command enables you to verify the configured EIGRP K values?
 - a. `show ip protocols`
 - b. `show ip eigrp interfaces`
 - c. `show ip eigrp neighbor`
 - d. `show ip eigrp topology`
4. Which command enables you to verify EIGRP authentication, split-horizon, and configured EIGRP timers?
 - a. `show ip interfaces`
 - b. `show ip protocols`
 - c. `show ip eigrp interfaces detail`
 - d. `show ip eigrp neighbor`

5. Besides a neighbor relationship not being formed, which three of the following are reasons why routes might be missing in your EIGRP autonomous system?
 - a. Interface not participating in the EIGRP process
 - b. Filters
 - c. Incorrect stub configuration
 - d. Passive interface feature
6. Which command enables you to verify whether any route filters have been applied to an EIGRP enabled interface?
 - a. `show ip interface brief`
 - b. `show ip interface`
 - c. `show ip protocols`
 - d. `show ip eigrp interface`
7. Which command enables you to verify the maximum paths configured for load balancing and whether unequal path load balancing has been enabled?
 - a. `show ip protocols`
 - b. `show ip eigrp interfaces`
 - c. `show ip eigrp neighbors`
 - d. `show ip interfaces`
8. Which EIGRP for IPv6 command is used to verify whether any interfaces have been configured as passive interfaces?
 - a. `show ipv6 protocols`
 - b. `show ipv6 eigrp interface detail`
 - c. `show ipv6 eigrp neighbor detail`
 - d. `show ipv6 eigrp topology`
9. Which EIGRP for IPv6 command enables you to verify whether the local router is a stub router?
 - a. `show ipv6 protocols`
 - b. `show ipv6 eigrp interface detail`
 - c. `show ipv6 eigrp neighbor detail`
 - d. `show ipv6 eigrp topology`

- 10.** Which EIGRP for IPv6 command enables you to verify whether a neighboring router is a stub router?
- a.** `show ipv6 protocols`
 - b.** `show ipv6 eigrp interface detail`
 - c.** `show ipv6 eigrp neighbor detail`
 - d.** `show ipv6 eigrp topology`
- 11.** What are two ways that you can verify which interfaces are participating in the named EIGRP IPv4 address family?
- a.** `show ip eigrp interfaces`
 - b.** `show eigrp address-family ipv4 interfaces`
 - c.** `show ipv6 eigrp interfaces`
 - d.** `show eigrp address-family ipv6 interfaces`

Foundation Topics

Troubleshooting EIGRP for IPv4

EIGRP establishes neighbor relationships by sending hello packets to the multicast address 224.0.0.10 out interfaces participating in the EIGRP process. To enable the EIGRP process on an interface, you use the **network** *ip_address wildcard_mask* command in router EIGRP configuration mode. For example, the following **network** command enables EIGRP on all interfaces with an IP address from 10.1.1.0 through 10.1.1.255: **network 10.1.1.0 0.0.0.255**. The following **network** command enables the EIGRP process on only the interface with the IP address 10.1.1.65: **network 10.1.1.65 0.0.0.0**. It seems rather simple, and it is; however, there are many reasons why a neighbor relationship may not form, and you need to be aware of all of them if you plan on successfully troubleshooting EIGRP-related problems.

After establishing a neighbor relationship, an EIGRP router performs a full exchange of routing information with the newly established neighbor. After the full exchange, only updates to route information are exchanged with that neighbor. Routing information learned from EIGRP neighbors is inserted into the EIGRP topology table. If the EIGRP information for a specific route happens to be the best source of information, it is installed in the routing table. There are various reasons as to why EIGRP routes might be missing from either the topology table or the routing table, and you need to be aware of all of them if you plan on successfully troubleshooting EIGRP route-related problems.

This section focuses on the reasons why an EIGRP neighbor relationship might not form and how you can identify them during the troubleshooting process. In addition, we examine the reasons why EIGRP routes might be missing, and how we can determine the reason why they are missing. To wrap up the section, we troubleshoot EIGRP issues that do not fall into the neighbor relationship or route categories.

Troubleshooting EIGRP for IPv4 Neighbor Adjacencies

To verify EIGRP neighbors, you use the **show ip eigrp neighbors** command. In Example 14-1, you can see sample output of the **show ip eigrp neighbors** command. It lists the IPv4 address of the neighboring device's interface that sent the hello packet, the local interface on the router used to reach that neighbor, how long the local router will consider the neighboring router to be a neighbor, how long the routers have been neighbors for, the amount of time it takes for the neighbors to communicate on average, the number of EIGRP packets in a queue waiting to be sent to a neighbor (which should always be zero since we want up-to-date routing information), and a sequence number to keep track of the EIGRP packets received from the neighbor to ensure that only newer packets are accepted and processed.

Example 14-1 *Verifying EIGRP Neighbors with show ip eigrp neighbors*

```
R2#show ip eigrp neighbors
EIGRP-IPv4 Neighbors for AS(100)
H   Address      Interface          Hold Uptime    SRTT   RTO  Q  Seq
                               (sec)          (ms)          Cnt  Num
1   10.1.23.3     Gi1/0             14 10:01:09    72   432  0   3
0   10.1.12.1     Gi0/0             11 10:32:14    75   450  0   8
```

Here is a listing of reasons why an EIGRP neighbor relationship might not form:



- **Interface is down:** The interface has to be up/up.
- **Mismatched autonomous system numbers:** Both routers need to be in the same autonomous system.
- **Incorrect network statement:** The **network** statement must identify the IP address of the interface you want to include in the EIGRP process.
- **Mismatched K values:** Both routers must be using the exact same K values.
- **Passive interface:** The passive interface feature suppresses the sending and receiving of hello packets while still allowing the interfaces network to be advertised.
- **Different subnets:** The exchanging of hello packets must be done on the same subnet; if not, the hello packets are ignored.
- **Authentication:** If authentication is being used, the key ID and key string must match, in addition to when the key is valid (if configured).
- **ACL:** An access control list (ACL) that is denying packets to the EIGRP multicast address 224.0.0.10.
- **Timers:** Timers do not have to match; however, if they are not configured correctly, your neighbor adjacencies will flap.

When an EIGRP neighbor relationship does not form, the neighbor is not listed in the neighbor table. Therefore, you will need the assistance of an accurate network diagram and the **show cdp neighbors** command to verify who should be the neighbors.

When troubleshooting EIGRP, you need to be aware of how to verify the parameters associated with each reason listed. Let's look at them individually.

Interface Is Down

The interface has to be up if you plan on forming an EIGRP neighbor adjacency. As you have seen already, you can verify the status of an interface with the **show ip interface brief** command.

Mismatched Autonomous System Numbers

For an EIGRP neighbor relationship to be formed, both routers need to be in the same autonomous system. The autonomous system number is specified when you issue the

router eigrp autonomous_system_number command in global configuration mode. If both routers are in different autonomous systems, they will not form an EIGRP neighbor relationship. Most EIGRP **show** commands will display the autonomous system number in the output. However, the best one is **show ip protocols**, which displays an incredible amount of information for troubleshooting, as shown in Example 14-2. In this example, you can verify that R1 is participating in EIGRP autonomous system 100. Using the *spot-the-difference* method, you can compare the autonomous system value listed to the value on a neighboring router to determine whether it differs.



Example 14-2 Verifying the Autonomous System Number with show ip protocols

```
R1#show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "eigrp 100"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP-IPv4 Protocol for AS(100)
    Metric weight K1=1, K2=0, K3=1, K4=0, K5=0
    NSF-aware route hold timer is 240
    Router-ID: 10.1.12.1
    Topology : 0 (base)
      Active Timer: 3 min
      Distance: internal 90 external 170
      Maximum path: 4
      Maximum hopcount 100
      Maximum metric variance 1

Automatic Summarization: disabled
Maximum path: 4
Routing for Networks:
  10.1.1.1/32
  10.1.12.1/32
Routing Information Sources:
  Gateway         Distance      Last Update
  10.1.12.2        90           09:54:36
Distance: internal 90 external 170
```

When using the **debug eigrp packet** command, as shown in Example 14-3, the **debug** output will show that the router is not receiving any hello packets from the neighbors with the mismatched autonomous system number. In this example, R1 is sending hello packets out Gig0/0 and Gig1/0. However, it is not receiving any hello packets. This could be because of an autonomous system mismatch. The local router could have the wrong autonomous system number, or the remote routers could have the wrong autonomous system number.

Example 14-3 *Sample Output of debug eigrp packet When an Autonomous System Mismatch Exists*

```

R1#debug eigrp packets
      (UPDATE, REQUEST, QUERY, REPLY, HELLO, UNKNOWN, PROBE, ACK, STUB, SIAQUERY,
      SIAREPLY)
EIGRP Packet debugging is on
R1#
EIGRP: Sending HELLO on Gi0/0 - paklen 20
      AS 100, Flags 0x0:(NULL), Seq 0/0 interfaceQ 0/0 iidbQ un/rely 0/0
R1#
EIGRP: Sending HELLO on Gi1/0 - paklen 20
      AS 100, Flags 0x0:(NULL), Seq 0/0 interfaceQ 0/0 iidbQ un/rely 0/0
R1#
EIGRP: Sending HELLO on Gi0/0 - paklen 20
      AS 100, Flags 0x0:(NULL), Seq 0/0 interfaceQ 0/0 iidbQ un/rely 0/0
R1#1
EIGRP: Sending HELLO on Gi1/0 - paklen 20
      AS 100, Flags 0x0:(NULL), Seq 0/0 interfaceQ 0/0 iidbQ un/rely 0/0
R1#1
EIGRP: Sending HELLO on Gi0/0 - paklen 20
      AS 100, Flags 0x0:(NULL), Seq 0/0 interfaceQ 0/0 iidbQ un/rely 0/0
R1#u all
All possible debugging has been turned off

```

Incorrect Network Statement

If the **network** command is misconfigured, EIGRP may not be enabled on the proper interfaces, and as a result, hello packets will not be sent and neighbor relationships will not be formed. You can verify the interfaces that are participating in the EIGRP process with the command **show ip eigrp interfaces**, as shown in Example 14-4. In this output, you can see that two interfaces are participating in the EIGRP process for autonomous system 100. Gi0/0 does not have an EIGRP peer, and Gi1/0 does have an EIGRP peer. This is expected because there are no other routers reachable out Gi0/0. However, if you expect an EIGRP peer out the interface based on your documentation, you would need to troubleshoot why the peering/neighbor relationship is not forming. Shift your attention to the Pending Routes column. Notice all interfaces are listed as 0. This is expected. Any other value in this column means that some issue on the network is preventing the interface from sending the necessary updates to the neighbor. For example, it might be congestion.

Note Remember that EIGRP passive interfaces do not show up in this output.



Example 14-4 Verifying EIGRP Interfaces with show ip eigrp interfaces

```
R2#show ip eigrp interfaces
EIGRP-IPv4 Interfaces for AS(100)
```

Interface	Peers	Un/Reliable	Un/Reliable	Mean SRTT	Pacing Time	Multicast Flow Timer	Pending Routes
Gi0/0	0	0/0	0/0	0	0/0	0	0
Gi1/0	1	0/0	0/0	78	0/0	300	0

The output of **show ip protocols** displays the interfaces that are running EIGRP as a result of the **network** commands. It is not obvious at first unless someone tells you, like I just did. The reason it is not obvious is that it is not displayed properly. Focus on the highlighted text in Example 14-5. Notice that it states *Routing for Networks*. Those are *not* the networks we are routing for. We are routing for the networks associated with the interface EIGRP will be enabled on based on the **network** commands. In this case, **10.1.1.1/32** really means **network 10.1.1.1 0.0.0.0** and **10.1.12.1/32** really means **network 10.1.12.1 0.0.0.0**. Therefore, a better option is using the **show run | section router eigrp** command, as displayed in Example 14-6.

Example 14-5 Verifying Network Statements with show ip protocols

```
R1#show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "eigrp 100"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
EIGRP-IPv4 Protocol for AS(100)
  Metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  NSF-aware route hold timer is 240
  Router-ID: 10.1.12.1
  Topology : 0 (base)
    Active Timer: 3 min
    Distance: internal 90 external 170
    Maximum path: 4
    Maximum hopcount 100
    Maximum metric variance 1

Automatic Summarization: disabled
Maximum path: 4
Routing for Networks:
  10.1.1.1/32
  10.1.12.1/32
Routing Information Sources:
  Gateway         Distance      Last Update
  10.1.12.2        90           09:54:36
Distance: internal 90 external 170
```

Example 14-6 *Verifying network Statements with show run | section router eigrp*

```
R1#show run | s router eigrp
router eigrp 100
  network 10.1.1.1 0.0.0.0
  network 10.1.12.1 0.0.0.0
```

As you can see, the **network** statement is extremely important. If it is misconfigured, interfaces that should be participating in the EIGRP process might not be, and interfaces that should not be participating in the EIGRP process might be. So, you should be able to recognize issues related to the **network** statement.

When using the **debug eigrp packet** command on the router with the misconfigured or missing network statement, you will notice that hello packets are not being sent out the interface that they should be. For example, if you expect hello packets to be sent out Gig1/0 but the **debug eigrp packet** command is not indicating so, it is possible that the interface is not participating in the EIGRP process because of a bad **network** statement.

Mismatched K Values

The K values, which are used during the metric calculation, must match between neighbors to form an adjacency. You can verify whether K values match with **show ip protocols**, as shown in Example 14-7. The default K values are highlighted in Example 14-7. Usually there is no need to change the K values. However, if they are changed, make them match on every router in the autonomous system. You can use the *spot-the-difference* method when determining whether K values do not match between routers. In addition, if you are logging syslog messages with a severity level of 5, you will receive a message similar to the following:

```
%DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 10.1.12.2 (GigabitEthernet1/0)
is down: K-value mismatch
```

**Example 14-7** *Verifying K Values with show ip protocols*

```
R1#show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "eigrp 100"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP-IPv4 Protocol for AS(100)
    Metric weight K1=1, K2=0, K3=1, K4=0, K5=0
    NSF-aware route hold timer is 240
    Router-ID: 10.1.12.1
    Topology : 0 (base)
    Active Timer: 3 min
    Distance: internal 90 external 170
```

```

Maximum path: 4
Maximum hopcount 100
Maximum metric variance 1

Automatic Summarization: disabled
Maximum path: 4
Routing for Networks:
  10.1.1.1/32
  10.1.12.1/32
Routing Information Sources:
  Gateway          Distance      Last Update
  10.1.12.2         90           09:54:36
Distance: internal 90 external 170

```

Passive Interface

The passive interface feature is a must have for all organizations. It does two things:

- Reduces the EIGRP related traffic on a network
- Improves EIGRP security

The passive interface feature turns off the sending and receiving of EIGRP packets on an interface while still allowing the interfaces network ID to be injected into the EIGRP process and advertised to other EIGRP neighbors. This ensures that rogue routers attached to the LAN will not be able to form an adjacency with your legitimate router on that interface, because it is not sending or receiving EIGRP packets on the interface. However, if you configure the wrong interface as passive, a legitimate EIGRP neighbor relationship will not be formed. As shown in the **show ip protocols** output of Example 14-8, Gigabit Ethernet 0/0 is a passive interface. If there are no passive interfaces, the passive interface section does not appear in the **show ip protocols** output.



Example 14-8 *Verifying Passive Interfaces with show ip protocols*

```

R1#show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "eigrp 100"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
EIGRP-IPv4 Protocol for AS(100)
  Metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  NSF-aware route hold timer is 240
  Router-ID: 10.1.12.1
  Topology : 0 (base)

```



```

Active Timer: 3 min
Distance: internal 90 external 170
Maximum path: 4
Maximum hopcount 100
Maximum metric variance 1

Automatic Summarization: disabled
Maximum path: 4
Routing for Networks:
  10.1.1.1/32
  10.1.12.1/32
Passive Interface(s):
  GigabitEthernet0/0
Routing Information Sources:
  Gateway          Distance      Last Update
  10.1.12.2         90           11:00:14
Distance: internal 90 external 170

```

Remember, for EIGRP, passive interfaces will not appear in the EIGRP interface table. Therefore, before you jump to the conclusion that the wrong network command was used and the interface has not been enabled for EIGRP, check to see whether the interface is passive.

When using the **debug eigrp packet** command on the router with the passive interface, you will notice that hello packets are not being sent out that interface. For example, if you expect hello packets to be sent out Gig1/0 but the **debug eigrp packet** command is not indicating so, it is possible that interface is participating in the EIGRP process but is configured as a passive interface.

Different Subnets

To form an EIGRP neighbor adjacency, the router interfaces must be on the same subnet. You can confirm this in many ways. The simplest way is to look at the interface configuration in the running configuration with the **show run interface interface_type interface_number** command. Example 14-9 displays the configuration of Gig1/0 on R1 and Gig0/0 on R2. Are they in the same subnet? Yes! Based on the IP address and the subnet mask, they would both be in the 10.1.12.0/24 subnet. However, if they are not in the same subnet and you have syslog setup for a severity level of 6, a message similar to the following will be displayed:

```
%DUAL-6-NBRINFO: EIGRP-IPv4 100: Neighbor 10.1.21.2 (GigabitEthernet1/0) is
blocked: not on common subnet (10.1.12.1/24)
```

Example 14-9 Verifying IPv4 Addresses and Masks on Router Interfaces

```

R1#show running-config interface gigabitEthernet 1/0
Building configuration...

Current configuration : 90 bytes
!

```

```

interface GigabitEthernet1/0
  ip address 10.1.12.1 255.255.255.0
  negotiation auto
end

R2#show running-config interface gigabitEthernet 0/0
Building configuration...

Current configuration : 132 bytes
!
interface GigabitEthernet0/0
  ip address 10.1.12.2 255.255.255.0
  negotiation auto
end

```

Authentication



Authentication is used to ensure that your EIGRP routers only form neighbor relationships with legitimate routers and that they only accept EIGRP packets from legitimate routers. Therefore, if authentication is implemented, both routers must agree on the settings for a neighbor relationship to form. With authentication, you can use the *spot-the-difference* method. Example 14-10 is displaying the output of the commands **show run interface interface_type interface_number** and **show ip eigrp interface detail interface_type interface_number**, which will identify whether EIGRP authentication is enabled on the interface. According to the highlighted text, it is. Note that the authentication must be on the correct interface and that it must be tied to the correct autonomous system number. If you put in the wrong autonomous system number, it will not be enabled for the correct autonomous system. In addition, make sure that you specify the correct key chain that will be used for the message digest 5 (MD5) authentication hash. You can verify the key chain with the command **show key chain**, as shown in Example 14-11. The keys in this example do not expire. However, if you have implemented rotating keys, the keys must be valid for authentication to be successful.

Example 14-10 Verifying EIGRP Authentication on an Interface

```

R1#show run interface gig 1/0
Building configuration...

Current configuration : 178 bytes
!
interface GigabitEthernet1/0
  ip address 10.1.12.1 255.255.255.0
  ip authentication mode eigrp 100 md5
  ip authentication key-chain eigrp 100 EIGRP_AUTH
  negotiation auto
end

```

```
R1#show ip eigrp interfaces detail gigabitEthernet 1/0
EIGRP-IPv4 Interfaces for AS(100)

      Xmit Queue   PeerQ      Mean   Pacing Time   Multicast    Pending
Interface Peers  Un/Reliable Un/Reliable SRTT  Un/Reliable  Flow Timer  Routes
Gi1/0      1      0/0        0/0      87     0/0        376        0

Hello-interval is 5, Hold-time is 15
Split-horizon is enabled
Next xmit serial <none>
Packetized sent/expedited: 2/0
Hello's sent/expedited: 17/2
Un/reliable mcasts: 0/3 Un/reliable ucasts: 2/2
Mcast exceptions: 0 CR packets: 0 ACKs suppressed: 0
Retransmissions sent: 1 Out-of-sequence rcvd: 1
Topology-ids on interface - 0
Authentication mode is md5, key-chain is "EIGRP_AUTH"
```

Example 14-11 Verifying the Key Chain Used for EIGRP Authentication

```
R1#show key chain
Key-chain EIGRP_AUTH:
  key 1 -- text "TSHOOT"
    accept lifetime (always valid) - (always valid) [valid now]
    send lifetime (always valid) - (always valid) [valid now]
```

Inside the key chain you find the key ID (1 in this case) and the key string (TSHOOT in this case). It is mandatory that the key ID in use and the key string in use between neighbors match. Therefore, if you have multiple keys and key strings in a chain, the same key and string must be used at the same time by both routers (meaning they must be valid and in use); otherwise, authentication will fail.

When using the **debug eigrp packets** command for troubleshooting authentication, you will receive different output based on the authentication issue. Example 14-12 displays what message is generated when the neighbor is not configured for authentication. It ignores that packet and states “(missing authentication).” When the key IDs or the key strings do not match between the neighbors, the **debug** output states “(invalid authentication),” as shown in Example 14-13.

Example 14-12 Example debug Output When Authentication Is Missing on Neighbor

```
R1#debug eigrp packets
(UPDATE, REQUEST, QUERY, REPLY, HELLO, UNKNOWN, PROBE, ACK, STUB, SIAQUERY,
SIAREPLY)
EIGRP Packet debugging is on
R1#
EIGRP: Sending HELLO on Gi1/0 - paklen 60
AS 100, Flags 0x0:(NULL), Seq 0/0 interfaceQ 0/0 iidbQ un/rely 0/0
```

```
EIGRP: Gi1/0: ignored packet from 10.1.12.2, opcode = 5 (missing authentication)
EIGRP: Sending HELLO on Gi0/0 - paklen 20
      AS 100, Flags 0x0:(NULL), Seq 0/0 interfaceQ 0/0 iidbQ un/rely 0/0
R1#u all
All possible debugging has been turned off
```

Example 14-13 Example Debug Output When Key IDs or Key Strings Do Not Match

```
R1#debug eigrp packets
      (UPDATE, REQUEST, QUERY, REPLY, HELLO, UNKNOWN, PROBE, ACK, STUB, SIAQUERY,
      SIAREPLY)
EIGRP Packet debugging is on
R1#
EIGRP: pkt authentication key id = 2, key not defined
EIGRP: Gi1/0: ignored packet from 10.1.12.2, opcode = 5 (invalid authentication)
EIGRP: Sending HELLO on Gi0/0 - paklen 20
      AS 100, Flags 0x0:(NULL), Seq 0/0 interfaceQ 0/0 iidbQ un/rely 0/0
EIGRP: Sending HELLO on Gi1/0 - paklen 60
      AS 100, Flags 0x0:(NULL), Seq 0/0 interfaceQ 0/0 iidbQ un/rely 0/0
R1#u all
All possible debugging has been turned off
```

ACLs

Access control lists are extremely powerful. How they are implemented will determine what they are controlling in your network. If there is an ACL applied to an interface and the ACL is denying EIGRP packets, a neighbor relationship will not form. To determine whether an ACL is applied to an interface, use the **show ip interface interface_type interface_number** command, as shown in Example 14-14. Notice that ACL 100 is applied inbound on interface Gig 1/0. To verify the ACL 100 entries, issue the command **show access-list 100**, as shown in Example 14-15. In this case, you can see that ACL 100 is denying EIGRP traffic, which would prevent a neighbor relationship from forming.

Example 14-14 Verifying ACLs Applied to Interfaces

```
R1#show ip interface gig 1/0
GigabitEthernet1/0 is up, line protocol is up
  Internet address is 10.1.12.1/24
  Broadcast address is 255.255.255.255
  Address determined by setup command
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Multicast reserved groups joined: 224.0.0.10
  Outgoing access list is not set
  Inbound access list is 100
  Proxy ARP is enabled
```

```
Local Proxy ARP is disabled
Security level is default
Split horizon is enabled
```

Example 14-15 *Verifying ACL Entries*

```
R1#show access-lists 100
Extended IP access list 100
 10 deny eigrp any any (62 matches)
 20 permit ip any any
```

Timers

Although EIGRP timers do not have to match, if the timers are skewed enough, the adjacency will flap. For example, suppose that R1 is using the default timers of 5 and 15, while R2 is sending hello packets every 20 seconds. R1's hold time will expire before it receives another hello packet from R2 terminating the neighbor relationship. Five seconds later, the hello packet arrives, and the neighbor relationship is formed, only to be terminated 15 seconds later.

Although timers do not have to match, it is important that each router sends hello packets at a rate that is faster than the hold timer. You can verify the configured timers with the `show ip eigrp interfaces detail` command, as shown earlier in Example 14-10.

Troubleshooting EIGRP for IPv4 Routes

EIGRP only learns from directly connected neighbors, making it easy to follow the path of routes when troubleshooting. For example, if R1 does not know about the route but its neighbor does, there is more than likely something wrong between the neighbors. However, if the neighbor does not know about it either, you can focus on the neighbors' neighbor and so on.

As we have discussed already, neighbor relationships are the foundation for EIGRP information sharing. If we have no neighbors, we will not learn any routes. So, besides the lack of a neighbor, what would be reasons for missing routes in an EIGRP network? Following is a listing of some common reasons as to why EIGRP routes might be missing either in the topology table or the routing table:



- **Bad or missing network command:** The `network` command enables the EIGRP process on an interface and injects the network the interface is part of into the EIGRP process.
- **Better source of information:** If the exact same network is learned from a more reliable source, it is used instead of the EIGRP learned information.
- **Route filtering:** A filter might be set up that is preventing a route from being advertised or learned.

- **Stub configuration:** If the wrong setting is chosen during the stub router configuration, or the wrong router is chosen as the stub router, you might prevent a network from being advertised.
- **Interface is shut down:** The EIGRP enabled interface must be up/up for the network associated with the interface to be advertised.
- **Split-horizon:** Loop-prevention feature that prevents a router from advertising routes out the same interface they were learned on.

Let's take a look at each of these individually and identify how to recognize them during the troubleshooting process.

Bad or Missing **Network** Command

When you use the **network** command, the EIGRP process is enabled on the interfaces that fall within the range of IP addresses identified by the command. EIGRP then takes the network/subnet the interface is part of and injects it into the topology table so that it can be advertised to other routers in the autonomous system. Therefore, even interfaces that will not form neighbor relationships with other routers need a valid **network** statement that will enable EIGRP on those interfaces so the networks the interfaces belong to will be injected into the EIGRP process and advertised. If the **network** statement is missing or configured incorrectly, EIGRP will not be enabled on the interface, and the network the interface belongs to will not be advertised.

As discussed in an earlier section, the output of **show ip protocols** displays the **network** statements in a nonintuitive way. Focus on the highlighted text in Example 14-16. Notice that it states *Routing for Networks*. Those are *not* the networks we are routing for. We are routing for the networks associated with the interface EIGRP will be enabled on based on the **network** statement. In this case, **10.1.1.1/32** really means **network 10.1.1.0.0.0.0**, and **10.1.12.1/32** really means **network 10.1.12.1 0.0.0.0**.

Example 14-16 *Verifying network Statements with show ip protocols*

```
R1#show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "eigrp 100"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP-IPv4 Protocol for AS(100)
    Metric weight K1=1, K2=0, K3=1, K4=0, K5=0
    NSF-aware route hold timer is 240
    Router-ID: 10.1.12.1
    Topology : 0 (base)
      Active Timer: 3 min
      Distance: internal 90 external 170
```

```

Maximum path: 4
Maximum hopcount 100
Maximum metric variance 1

Automatic Summarization: disabled
Maximum path: 4
Routing for Networks:
  10.1.1.1/32
  10.1.12.1/32
Routing Information Sources:
  Gateway          Distance      Last Update
  10.1.12.2         90           09:54:36
Distance: internal 90 external 170

```

So what networks are we actually routing for then? The networks associated with the interfaces that are now enabled for EIGRP. In Example 14-17, you can see the output of the **show ip interface** command on R1 for Gig0/0 and Gig1/0, which was piped to only include the **Internet** address. Notice that they are in a /24 network. As a result, the network IDs would be 10.1.1.0/24 and 10.1.12.0/24. *Those* are the networks we are routing for.

Example 14-17 *Verifying Network IDs with show ip interface*

```

R1#show ip interface gi0/0 | i Internet
  Internet address is 10.1.1.1/24
R1#show ip interface gil/0 | i Internet
  Internet address is 10.1.12.1/24

```

Therefore, if you expect to route for the network 10.1.1.0/24 or 10.1.12.0/24, as in this case, you better have a **network** statement that enables the EIGRP process on the router interfaces in those networks.

You can confirm which interfaces are participating in the EIGRP process with the **show ip eigrp interfaces** command, as shown earlier.

Better Source of Information

For an EIGRP-learned route to be installed in the routing table, it has to be the most trusted routing source. Recall that this is based on administrative distance (AD). EIGRP's AD is 90 for internally learned routes (networks inside the autonomous system) and 170 for externally learned routes (networks outside the autonomous system). Therefore, if there is another source that is educating the same router about the exact same network and that source has a better AD, the source with the better AD wins, and its information will be installed in the routing table. Compare Example 14-18, which is an EIGRP topology table, and Example 14-19, which is the routing table displaying only the EIGRP installed routes on the router. Focus on the highlighted networks of the topology table. Do you see them listed as EIGRP routes in the routing table?

Example 14-18 *Sample show ip eigrp topology Command Output*

```

Router#show ip eigrp topology
EIGRP-IPv4 Topology Table for AS(100)/ID(192.4.4.4)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 172.16.33.8/30, 2 successors, FD is 2681856
    via 172.16.33.6 (2681856/2169856), Serial1/0
    via 172.16.33.18 (2681856/2169856), Serial1/2
P 10.1.34.0/24, 1 successors, FD is 2816
    via Connected, GigabitEthernet2/0
P 192.7.7.7/32, 1 successors, FD is 2300416
    via 172.16.33.5 (2300416/156160), Serial1/0
    via 172.16.33.6 (2809856/2297856), Serial1/0
    via 172.16.33.18 (2809856/2297856), Serial1/2
P 192.4.4.4/32, 1 successors, FD is 128256
    via Connected, Loopback0
P 172.16.33.16/30, 1 successors, FD is 2169856
    via Connected, Serial1/2
P 172.16.32.0/25, 2 successors, FD is 2172416
    via 172.16.33.6 (2172416/28160), Serial1/0
    via 172.16.33.18 (2172416/28160), Serial1/2
P 10.1.23.0/24, 1 successors, FD is 3072
    via 10.1.34.3 (3072/2816), GigabitEthernet2/0
P 203.0.113.0/30, 1 successors, FD is 28160
    via Connected, FastEthernet3/0
P 192.5.5.5/32, 1 successors, FD is 2297856
    via 172.16.33.5 (2297856/128256), Serial1/0
P 192.3.3.3/32, 1 successors, FD is 130816
    via 10.1.34.3 (130816/128256), GigabitEthernet2/0
P 192.2.2.2/32, 1 successors, FD is 131072
    via 10.1.34.3 (131072/130816), GigabitEthernet2/0
P 10.1.13.0/24, 1 successors, FD is 3072
    via 10.1.34.3 (3072/2816), GigabitEthernet2/0
P 0.0.0.0/0, 1 successors, FD is 28160
    via Rstatic (28160/0)
P 192.1.1.1/32, 1 successors, FD is 131072
    via 10.1.34.3 (131072/130816), GigabitEthernet2/0
P 172.16.32.192/29, 1 successors, FD is 2174976
    via 172.16.33.5 (2174976/30720), Serial1/0
    via 172.16.33.6 (2684416/2172416), Serial1/0
    via 172.16.33.18 (2684416/2172416), Serial1/2
P 198.51.100.0/30, 1 successors, FD is 28416
    via 10.1.34.3 (28416/28160), GigabitEthernet2/0
P 172.16.33.12/30, 1 successors, FD is 2172416

```



```

        via 172.16.33.5 (2172416/28160), Serial1/0
P 192.6.6.6/32, 2 successors, FD is 2297856
    via 172.16.33.6 (2297856/128256), Serial1/0
    via 172.16.33.18 (2297856/128256), Serial1/2
P 172.16.33.0/29, 1 successors, FD is 2169856
    via Connected, Serial1/0
P 10.1.1.0/26, 1 successors, FD is 3328
    via 10.1.34.3 (3328/3072), GigabitEthernet2/0
P 172.16.32.128/26, 1 successors, FD is 2172416
    via 172.16.33.5 (2172416/28160), Serial1/0

```

Example 14-19 *Sample show ip route eigrp Command Output*

```

Router#show ip route eigrp
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is 203.0.113.1 to network 0.0.0.0

10.0.0.0/8 is variably subnetted, 5 subnets, 3 masks
D       10.1.1.0/26 [90/3328] via 10.1.34.3, 00:49:19, GigabitEthernet2/0
D       10.1.13.0/24 [90/3072] via 10.1.34.3, 00:49:22, GigabitEthernet2/0
D       10.1.23.0/24 [90/3072] via 10.1.34.3, 00:49:22, GigabitEthernet2/0
172.16.0.0/16 is variably subnetted, 9 subnets, 5 masks
D       172.16.32.0/25 [90/2172416] via 172.16.33.18, 00:49:22, Serial1/2
        [90/2172416] via 172.16.33.6, 00:49:22, Serial1/0
D       172.16.32.128/26 [90/2172416] via 172.16.33.5, 00:49:23, Serial1/0
D       172.16.32.192/29 [90/2174976] via 172.16.33.5, 00:49:23, Serial1/0
D       172.16.33.8/30 [90/2681856] via 172.16.33.18, 00:49:22, Serial1/2
        [90/2681856] via 172.16.33.6, 00:49:22, Serial1/0
D       172.16.33.12/30 [90/2172416] via 172.16.33.5, 00:49:23, Serial1/0
192.1.1.0/32 is subnetted, 1 subnets
D       192.1.1.1 [90/131072] via 10.1.34.3, 00:49:19, GigabitEthernet2/0
192.2.2.0/32 is subnetted, 1 subnets
D       192.2.2.2 [90/131072] via 10.1.34.3, 00:49:19, GigabitEthernet2/0
192.3.3.0/32 is subnetted, 1 subnets
D       192.3.3.3 [90/130816] via 10.1.34.3, 00:49:22, GigabitEthernet2/0
192.5.5.0/32 is subnetted, 1 subnets
D       192.5.5.5 [90/2297856] via 172.16.33.5, 00:49:23, Serial1/0
192.6.6.0/32 is subnetted, 1 subnets

```

```

D      192.6.6.6 [90/2297856] via 172.16.33.18, 00:49:22, Serial1/2
      [90/2297856] via 172.16.33.6, 00:49:22, Serial1/0
192.7.7.0/32 is subnetted, 1 subnets
D      192.7.7.7 [90/2300416] via 172.16.33.5, 00:49:23, Serial1/0
198.51.100.0/30 is subnetted, 1 subnets
D      198.51.100.0 [90/28416] via 10.1.34.3, 00:49:22, GigabitEthernet2/0

```

None of the highlighted routes in Example 14-18 appear in the routing table as EIGRP routes. In this case, there is a better source for the same information. Example 14-20, which displays the output of the **show ip route 172.16.33.16 255.255.255.252** command, identifies that this network is directly connected and has an AD of 0. Because a directly connected network has an AD of 0 and an internal EIGRP route has an AD of 90, the directly connected source is installed in the routing table. Refer back to Example 14-18 and focus on the 0.0.0.0/0 route. Notice that it says Rstatic, which means that the route was redistributed from a static route on this router. Therefore, there is a static default route on the local router with a better AD than the EIGRP default route, which would have an AD of 170. As a result, the EIGRP 0.0.0.0/0 route would not be installed in the routing table, the static default route would be.

Example 14-20 Sample show ip route 172.16.33.16 255.255.255.252 Command Output

```

Router#show ip route 172.16.33.16 255.255.255.252
Routing entry for 172.16.33.16/30
  Known via "connected", distance 0, metric 0 (connected, via interface)
  ...output omitted...

```

Key Topic

Having a better source of routing information may not cause users to complain or submit a trouble ticket because they will probably still be able to access the resources they need to. However, it may be causing suboptimal routing in your network. Review Figure 14-1, which shows a network running two different routing protocols. In this case, which path will be used to send traffic from PC1 to 10.1.1.0/24? If you said the longer EIGRP path, you are correct. Even though it is quicker to use the Open Shortest Path First (OSPF) path, EIGRP wins by default because it has the lower AD, and suboptimal routing occurs.

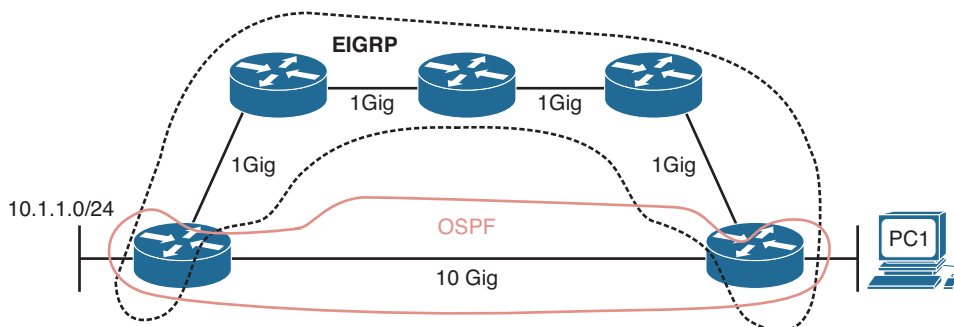


Figure 14-1 Using EIGRP Path Which Is Suboptimal

Being able to recognize when a certain routing source should be used and when it should not be used is key to optimizing your network and reducing the number of troubleshoot-

ing instances related to “*the network is slow.*” In this case, we might want to consider increasing the AD of EIGRP or lowering the AD of OSPF to optimize routing.

Route Filtering

A distribute list applied to an EIGRP process controls which routes are advertised to neighbors or which routes are received from neighbors. The distribute list is applied in EIGRP configuration mode either inbound or outbound, and the routes sent or received are controlled by ACLs, prefix lists, or route maps. So, when troubleshooting route filtering, you need to consider the following:



- Is the distribute list applied in the correct direction?
- Is the distribute list applied to the correct interface?
- If the distribute list is using an ACL, is the ACL correct?
- If the distribute list is using a prefix list, is the prefix list correct?
- If the distribute list is using a route map, is the route map correct?

The **show ip protocols** command will identify whether a distribute list is applied to all interfaces or an individual interface, as shown in Example 14-21. This example indicates that there are no outbound filters and that there is an inbound filter on Gig1/0.

Example 14-21 Verifying Route Filters with show ip protocols

```
R1#show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "eigrp 100"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  GigabitEthernet1/0 filtered by 10 (per-user), default is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP-IPv4 Protocol for AS(100)
    Metric weight K1=1, K2=0, K3=1, K4=0, K5=0
    NSF-aware route hold timer is 240
    Router-ID: 10.1.12.1
  ...output omitted...
```

The inbound filter in Example 14-21 on Gig1/0 is filtering with ACL 10. To verify the entries in the ACL, you must issue the **show access-lists 10** command. If a prefix list was applied, you issue the **show ip prefix-list** command. If a route map was applied you issue the **show route-map** command.

As displayed in Example 14-22, you can verify the command that was used to apply the distribute list in the running configuration by reviewing the EIGRP configuration section.

Example 14-22 *Verifying EIGRP distribute-list Command*

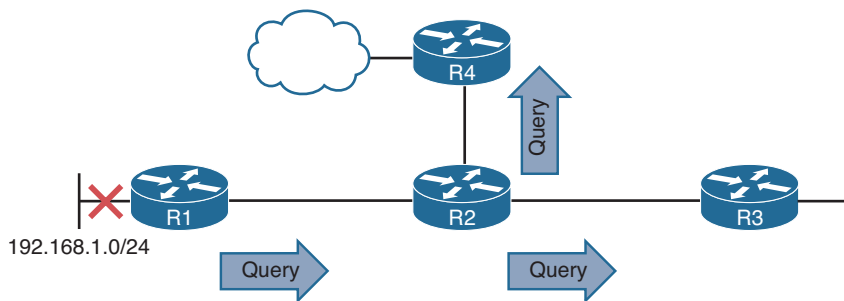
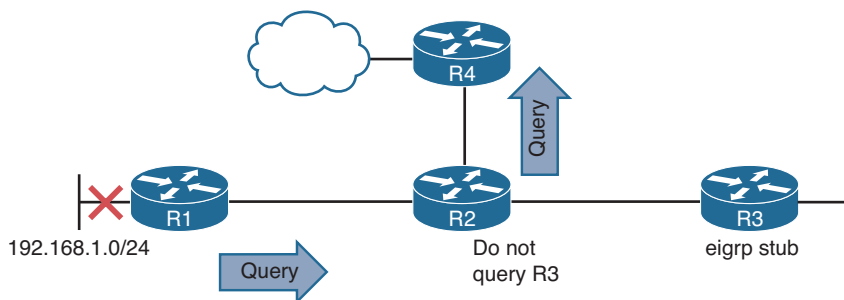
```

R1#show run | section router eigrp
router eigrp 100
  distribute-list 10 in GigabitEthernet1/0
  network 10.1.1.1 0.0.0.0
  network 10.1.12.1 0.0.0.0
  passive-interface GigabitEthernet0/0

```

Stub Configuration

The EIGRP stub feature allows you to control the scope of EIGRP queries in the network. Figure 14-2 shows the failure of network 192.168.1.0/24 on R1 that causes a query to be sent to R2 and then a query from R2 sent to R3 and R4. However, the query to R3 is not needed because R3 will never have alternate information about the 192.168.1.0/24 network. The query wastes resources. Configuring the EIGRP stub feature on R3 with the **eigrp stub** command will ensure that R2 never sends a query to R3, as shown in Figure 14-3.

**Figure 14-2** *Query Scope Without EIGRP Stub Feature***Figure 14-3** *Query Scope with EIGRP Stub Feature*

This feature comes in handy over slow hub-and-spoke WAN links, as shown in Figure 14-4. It prevents the hub from querying the spokes, which reduces the amount of EIGRP traffic sent over the link. In addition, it reduces the chance of a route being stuck-in-active (SIA). SIA happens when a router does not receive a reply to a query that it sent.

Over WANs, this can happen due to congestion and result in the reestablishment of neighbor relationships, which causes convergence, which generates even more EIGRP traffic. Therefore, if we do not query the hubs, we do not have to worry about these issues.

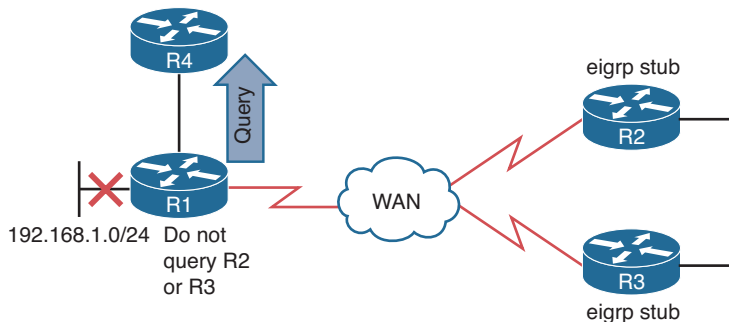


Figure 14-4 EIGRP Stub Feature over WAN Links

When configuring the EIGRP stub feature, you can control the routes that the stub router will advertise to its neighbor. By default, it is connected and summary routes. However, you have the option of just connected, summary, redistributed, static, or a combination of them. The other option is to send no routes (receive-only). If the wrong option is chosen, the stub routers would not be advertising the correct routes to its neighbors, resulting in missing routes on the hub and other routers in the topology. In addition, if you configure the wrong router as the stub router (for example, R1 in Figure 14-4), R1 would never fully share all routes it knows about to R4, R2, and R3, resulting in missing routes in the topology. To verify whether the router is a stub router and the routes it will advertise, issue the **show ip protocols** command, as shown in Example 14-23.

Example 14-23 **show ip protocols** Command Output on R2

```
R2#show ip protocols
...output omitted...
EIGRP-IPv4 Protocol for AS(100)
  Metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  NSF-aware route hold timer is 240
  Router-ID: 192.1.1.1
  Stub, connected, summary
  Topology : 0 (base)
  Active Timer: 3 min
  Distance: internal 90 external 170
  Maximum path: 4
...output omitted...
```

To determine whether a neighbor is a stub router and the types of routes it is advertising, issue the command **show ip eigrp neighbors detail**. Example 14-24 displays the output of **show ip eigrp neighbors detail** on R1 and indicates that the neighbor is a stub router advertising connected and summary routes.

Example 14-24 *Verifying Whether an EIGRP Neighbor Is a Stub Router*

```

R1#show ip eigrp neighbors detail
EIGRP-IPv4 Neighbors for AS(100)
H   Address                Interface                Hold Uptime    SRTT    RTO  Q  Seq
                               (sec)          (ms)          Cnt  Num
0   10.1.13.1              Ser1/0              14 00:00:18    99    594  0  11
Version 11.0/2.0, Retrans: 0, Retries: 0, Prefixes: 2
Topology-ids from peer - 0
Stub Peer Advertising (CONNECTED SUMMARY ) Routes
Suppressing queries
...output omitted...

```

Interface Is Shut Down

As discussed earlier, the **network** command enables the routing process on an interface. Once the EIGRP process is enabled on the interface, the network the interface is part of (the directly connected entry in the routing table) is injected into the EIGRP process. If the interface is shut down, there is no directly connected entry for the network in the routing table. Therefore, the network does not exist, and there is no network that can be injected into the EIGRP process. The interface has to be up/up for routes to be advertised or for neighbor relationships to be formed.

Split-horizon

The EIGRP split-horizon rule states that any routes learned inbound on an interface will not be advertised out the same interface. This rule is designed to prevent routing loops. However, this rule presents an issue in certain topologies. Figure 14-5 shows a nonbroadcast multiaccess (NBMA) Frame Relay hub-and-spoke topology or a dynamic multipoint virtual private network (DMVPN), which both use multipoint interfaces on the hub. The multipoint interface (single physical interface or mGRE tunnel interface) provides connectivity to multiple routers in the same subnet out the single interface, like Ethernet. In this figure, R2 is sending an EIGRP update to R1 on the permanent virtual circuit (PVC) or generic routing encapsulation (GRE) tunnel. Because split-horizon is enabled on the Ser1/0 interface or the multipoint GRE tunnel interface on R1, R1 will not advertise the 10.1.2.0/24 network back out that interface. So, R3 will never learn about 10.1.2.0/24.

To verify whether split-horizon is enabled on an interface, issue the **show ip interface interface_type interface_number** command, as shown in Example 14-25. In this case, you can see that split-horizon is enabled.

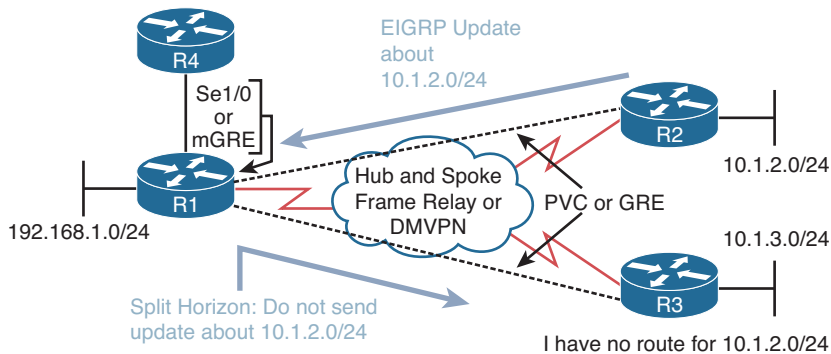


Figure 14-5 EIGRP Split-Horizon Issue

Example 14-25 Verifying Whether Split-horizon Is Enabled on an Interface

```
R1#show ip interface tunnel 0
Tunnel0 is up, line protocol is up
Internet address is 192.168.1.1/24
Broadcast address is 255.255.255.255
Address determined by setup command
MTU is 1476 bytes
Helper address is not set
Directed broadcast forwarding is disabled
Outgoing access list is not set
Inbound access list is not set
Proxy ARP is enabled
Local Proxy ARP is disabled
Security level is default
Split horizon is enabled
ICMP redirects are never sent
...output omitted...
```

To disable split-horizon on an interface completely, issue the **no ip split-horizon** command in interface configuration mode. If you only want to disable it for the EIGRP process running on the interface, issue the command **no ip split-horizon eigrp autonomous_system_number**.

If you disable split-horizon for the EIGRP process, it will still show as enabled in the output of **show ip interface**, as shown in Example 14-25 earlier. To verify whether split-horizon is enabled or disabled for the EIGRP process on an interface, issue the command **show ip eigrp interfaces detail interface_type interface_number**. Example 14-26 shows that it is disabled for EIGRP on interface tunnel 0.

Example 14-26 Verifying Whether Split-horizon Is Enabled for EIGRP on an Interface

```
R1#show ip eigrp interfaces detail tunnel 0
EIGRP-IPv4 Interfaces for AS(100)

Xmit Queue  PeerQ      Mean  Pacing Time  Multicast  Pending
```

Interface	Peers	Un/Reliable	Un/Reliable	SRTT	Un/Reliable	Flow Timer	Routes
Tu0	0	0/0	0/0	0	6/6	0	0

```

Hello-interval is 5, Hold-time is 15
Split-horizon is disabled
Next xmit serial <none>
Packetized sent/expedited: 0/0
Hello's sent/expedited: 17/1
Un/reliable mcasts: 0/0 Un/reliable ucasts: 0/0
Mcast exceptions: 0 CR packets: 0 ACKs suppressed: 0
Retransmissions sent: 0 Out-of-sequence rcvd: 0
Topology-ids on interface - 0
Authentication mode is not set

```

Troubleshooting Miscellaneous EIGRP for IPv4 Issues

So far, your focus has been on troubleshooting EIGRP neighbor relationships and routes. Now your focus will be on troubleshooting issues related to feasible successors, discontinuous networks and autosummarization, route summarization, and equal and unequal metric load balancing.

Feasible Successors

The best route (lowest feasible distance [FD] metric) for a specific network in the EIGRP topology table becomes a candidate to be injected into the router's routing table. (We use the term *candidate* because even though it is the best EIGRP route, there might be a better source of the same information that will be used instead.) If that route is indeed injected into the routing table, that route becomes known as the *successor* (best) route. This is the route that is then advertised to neighboring routers. Example 14-27 displays a sample EIGRP topology table, which you can view by issuing the **show ip eigrp topology** command. Focus on the entry for 172.16.32.192/29. Notice that there are three paths to reach that network. However, based on the fact that it states *1 successors*, only one is being used as the best path. It is the one with the lowest FD of 2174976, which is the path via 172.16.33.5, reachable out interface Serial 1/0.

Example 14-27 Sample show ip eigrp topology Command Output

```

R4#show ip eigrp topology
EIGRP-IPv4 Topology Table for AS(100)/ID(192.4.4.4)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

...output omitted...
P 10.1.1.13.0/24, 1 successors, FD is 3072
    via 10.1.34.3 (3072/2816), GigabitEthernet2/0
P 0.0.0.0/0, 1 successors, FD is 28160
    via Rstatic (28160/0)

```



```

P 192.1.1.1/32, 1 successors, FD is 131072
    via 10.1.34.3 (131072/130816), GigabitEthernet2/0
P 172.16.32.192/29, 1 successors, FD is 2174976
    via 172.16.33.5 (2174976/30720), Serial1/0
    via 172.16.33.6 (2684416/2172416), Serial1/0
    via 172.16.33.18 (2684416/2172416), Serial1/2
P 198.51.100.0/30, 1 successors, FD is 28416
    via 10.1.34.3 (28416/28160), GigabitEthernet2/0
P 172.16.33.12/30, 1 successors, FD is 2172416
    via 172.16.33.5 (2172416/28160), Serial1/0
...output omitted...

```

In the brackets after the next-hop IP address is the FD followed by the reported distance (RD):

- **Reported distance:** The distance from the neighbor at the next-hop address to the destination network
- **Feasible distance:** The RD plus the metric to reach the neighbor at the next-hop address that is advertising the RD

The successor is the path with the lowest FD. However, EIGRP also precalculates paths that could be used if the successor disappeared. These are known as the feasible successors. To be a feasible successor, the RD of the path to become a feasible successor must be less than the FD of the successor. Review Example 14-27 again. The path via 172.16.33.5 is the successor. However, are the paths using 172.16.33.6 and 172.16.33.18 feasible successors (backups)? To determine this, take the RD of these paths (in this case, it is the same [2172416]), and compare it to the FD of the successor (2174976). Is the RD less than the FD? Yes. Therefore, they are feasible successors.

For troubleshooting, it is important to note that the output of **show ip eigrp topology** only displays the successors and feasible successors. If you need to verify the FD or RD of other paths to the same destination that are not feasible successors, you can use the **show ip eigrp topology all-links** command. Example 14-28 displays the output of **show ip eigrp topology** and **show ip eigrp topology all-links**. Focus on the entry for 10.1.34.0/24. Notice how in the output of **show ip eigrp topology** there is only one path listed and in the output of **show ip eigrp topology all-links** there are two. This is because the next hop of 172.16.33.13 has an RD greater than the FD of the successor and therefore cannot be a feasible successor.

Example 14-28 Sample show ip eigrp topology Comparison

```

Router#show ip eigrp topology
EIGRP-IPv4 Topology Table for AS(100)/ID(172.16.33.14)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 172.16.33.8/30, 1 successors, FD is 2169856
    via Connected, Serial1/0

```

```

P 10.1.34.0/24, 1 successors, FD is 2682112
    via 172.16.33.9 (2682112/2170112), Serial1/0
P 203.0.113.0/30, 1 successors, FD is 2684416
    via 172.16.33.9 (2684416/2172416), Serial1/0
P 172.16.32.192/29, 1 successors, FD is 28160
    via Connected, FastEthernet2/0
P 172.16.33.12/30, 1 successors, FD is 5511936
    via Connected, Serial1/1
P 172.16.33.0/29, 1 successors, FD is 2681856
    via 172.16.33.9 (2681856/2169856), Serial1/0

Router#show ip eigrp topology all-links
EIGRP-IPv4 Topology Table for AS(100)/ID(172.16.33.14)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 172.16.33.8/30, 1 successors, FD is 2169856, serno 1
    via Connected, Serial1/0
P 10.1.34.0/24, 1 successors, FD is 2682112, serno 8
    via 172.16.33.9 (2682112/2170112), Serial1/0
    via 172.16.33.13 (6024192/3072256), Serial1/1
P 203.0.113.0/30, 1 successors, FD is 2684416, serno 9
    via 172.16.33.9 (2684416/2172416), Serial1/0
    via 172.16.33.13 (6026496/3074560), Serial1/1
P 172.16.32.192/29, 1 successors, FD is 28160, serno 3
    via Connected, FastEthernet2/0
P 172.16.33.12/30, 1 successors, FD is 5511936, serno 2
    via Connected, Serial1/1
P 172.16.33.0/29, 1 successors, FD is 2681856, serno 5
    via 172.16.33.9 (2681856/2169856), Serial1/0
    via 172.16.33.13 (6023936/3072000), Serial1/1

```

The EIGRP topology table not only contains the routes learned from other routers, but also routes that have been redistributed into the EIGRP process and the local networks whose interfaces are participating in the EIGRP process, as highlighted in Example 14-29.

Example 14-29 Verifying Connected and Redistributed Entries in the Topology Table

```

R4#show ip eigrp topology
EIGRP-IPv4 Topology Table for AS(100)/ID(192.4.4.4)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

...output omitted...
P 192.2.2.2/32, 1 successors, FD is 131072
    via 10.1.34.3 (131072/130816), GigabitEthernet2/0
P 10.1.13.0/24, 1 successors, FD is 3072

```

```

        via 10.1.34.3 (3072/2816), GigabitEthernet2/0
P 0.0.0.0/0, 1 successors, FD is 28160
    via Rstatic (28160/0)
P 192.1.1.1/32, 1 successors, FD is 131072
    via 10.1.34.3 (131072/130816), GigabitEthernet2/0
P 172.16.32.192/29, 1 successors, FD is 2174976
    via 172.16.33.5 (2174976/30720), Serial1/0
    via 172.16.33.6 (2684416/2172416), Serial1/0
    via 172.16.33.18 (2684416/2172416), Serial1/2
P 198.51.100.0/30, 1 successors, FD is 28416
    via 10.1.34.3 (28416/28160), GigabitEthernet2/0
P 172.16.33.12/30, 1 successors, FD is 2172416
    via 172.16.33.5 (2172416/28160), Serial1/0
P 192.6.6.6/32, 2 successors, FD is 2297856
    via 172.16.33.6 (2297856/128256), Serial1/0
    via 172.16.33.18 (2297856/128256), Serial1/2
P 172.16.33.0/29, 1 successors, FD is 2169856
    via Connected, Serial1/0
...output omitted...

```

Discontiguous Networks and Autosummarization

EIGRP supports variable-length subnet masking (VLSM). In earlier releases of the Cisco IOS (pre 15.0), EIGRP automatically performed route summarization at classful network boundaries. This was an issue in networks containing discontiguous networks. As a result, it was necessary when configuring EIGRP to turn off automatic summarization using the **no auto-summary** command in router configuration mode for an EIGRP autonomous system. However, from Cisco IOS 15.0 and onward, automatic summarization is off by default for EIGRP. Therefore, you do not have to worry about issuing the **no auto-summary** command anymore. However, you should be able to recognize a discontiguous network when reviewing a network topology and understand that if someone manually enabled autosummarization in your EIGRP autonomous system, routing would be broken.

Figure 14-6 provides an example of a discontiguous network. The 172.16.0.0/16 Class B classful network is considered discontiguous because it is subnetted as 172.16.1.0/24 and 172.16.2.0/24 and the subnets are separated from each other by a different classful network, which is 10.0.0.0. With automatic summarization turned on, when R3 advertises the 172.16.2.0/24 network to R2, it is summarized to 172.16.0.0/16 because it is being sent out an interface in a different classful network. So, instead of 172.16.2.0/24 being sent, 172.16.0.0/16 is sent. Likewise, the same thing happens when R1 advertises the 172.16.1.0/24 network to R2; it is advertised as 172.16.0.0/16. If you reviewed R2's routing table, it would show an entry for 172.16.0.0 with two next hops (if everything else is equal), one via R3 using Fa0/1 and the other via R1 using Fa0/0.

Now picture a packet arriving at R2 from R4 with a destination IP of 172.16.2.5. Which way does R2 send it? You see the problem? It should send it out Fa0/1, but it could send it out Fa0/0. There is a 50/50 chance it gets it correct. The moral of this story is this: If

you have a discontinuous network, autosummarization has to be off, and you must take care when performing manual summarization. To verify whether automatic summarization is enabled or disabled, use the **show ip protocols** command, as shown in Example 14-30.

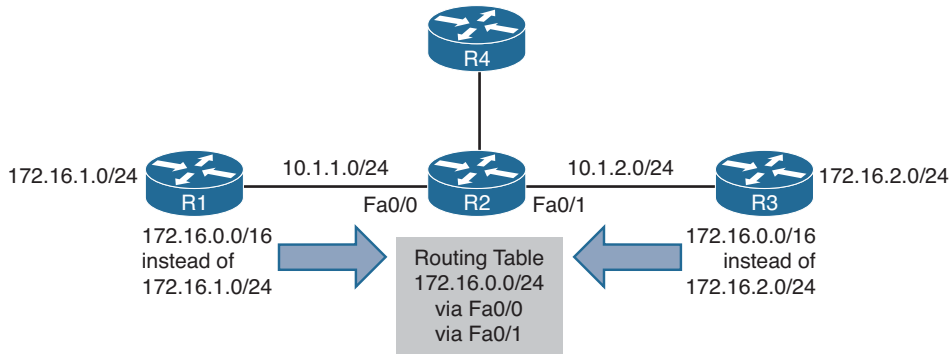


Figure 14-6 *Discontinuous Network Example*

Example 14-30 *Verifying Route Summarization with show ip protocols*

```
Router#show ip protocols
...output omitted...
EIGRP-IPv4 Protocol for AS(100)
  Metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  NSF-aware route hold timer is 240
  Router-ID: 10.1.13.1
  Topology : 0 (base)
  Active Timer: 3 min
  Distance: internal 90 external 170
  Maximum path: 4
  Maximum hopcount 100
  Maximum metric variance 1

Automatic Summarization: disabled
Address Summarization:
  10.1.0.0/20 for Gi2/0
  Summarizing 2 components with metric 2816
  Maximum path: 4
  Routing for Networks:
...output omitted...
```

Route Summarization

By default with IOS 15.0 and later, autosummary is off. Therefore, you can either turn it on (not recommended), or perform manual route summarization (recommended).

With EIGRP, manual route summarization is enabled on an interface-by-interface basis.



Therefore, when troubleshooting route summarization, keep the following in mind:

- Did you enable route summarization on the correct interface?
- Did you associate the summary route with the correct EIGRP autonomous system?
- Did you create the appropriate summary route?

You can verify all of these using the **show ip protocols** command, as shown in Example 14-30. In this example, autosummarization is disabled, and manual summarization is enabled for EIGRP autonomous system 100 on interface Gigabit Ethernet 2/0 for 10.1.0.0/20.

It is important that you create accurate summary routes to ensure that your router is not advertising networks in the summary route that it does not truly know how to reach. If it does, it is possible that it might receive packets to destinations that fall within the summary that it really does not know how to reach. If this is the case, it means that packets will be dropped because of the route to null 0.

When a summary route is created on a router, so is a summary route to null 0, as shown in Example 14-31. This route to null 0 is created to prevent routing loops. It is imperative that this route is in the table to ensure that if a packet is received by this router destined to a network that falls within the summary that the router does not really know how to reach, it will be dropped. If the route to null 0 did not exist, and there was a default route on the router, the router would forward the packet via the default route, and then the next-hop router would end up forwarding it back to this router, because it is using the summary route, then the local router would then forward it based on the default route, and then it would come back. This is a routing loop.

Example 14-31 Verifying Local Summary Route to Null 0

```
Router#show ip route | include Null
D      10.1.0.0/20 is a summary, 00:12:03, Null0
```

The route to null 0 has an AD of 5, as shown in Example 14-32, to ensure that it is more trustworthy than most of the other sources of routing information. Therefore, the only way this route would not be in the routing table is if you had a source with a lower AD (for example, someone creates a static route for the same summary network and points it to a next hop IP address instead of null 0). This would cause a routing loop.

Example 14-32 Verifying the AD of Local Summary Route to Null 0

```
Router#show ip route 10.1.0.0
Routing entry for 10.1.0.0/20
  Known via "eigrp 100", distance 5, metric 2816, type internal
```

Load Balancing

By default, EIGRP will load balance on four equal metric paths. You can change this with the **maximum-paths** command in router configuration mode for EIGRP. However, EIGRP also supports load balancing across unequal metric paths using the *variance* feature. By default, the variance value for an EIGRP routing process is 1, meaning the load balancing will only occur over equal metric paths. You can issue the **variance multiplier** command

in router configuration mode to specify a range of metrics over which load balancing will occur. For example, suppose that a route had a metric of 200000, and you configured the **variance 2** command for the EIGRP routing process. This would cause load balancing to occur over any route with a metric in the range of 200000 through 400000 ($2 * 200000$). As you can see, a route could have a metric as high as 400000 (that is, the variance multiplier multiplied by the best metric) and still be used.

However, even with unequal metric load balancing, you are still governed by the **maximum-paths** command. Therefore, if you have five unequal-metric paths that you want to use and you configured the correct variance multiplier, but maximum paths is set to 2, you will only use two of the five paths. To use all five, you would also need to make sure that the maximum paths were set to 5 as well.

Also, remember that the feasibility condition plays a huge role in unequal path load balancing. If the path is not a feasible successor, it cannot be used for unequal path load balancing. There is no exception to this rule. If you recall, the feasibility condition is this: *To be a feasible successor, your RD must be less than the FD of the successor.*

To verify the configured maximum paths and variance, use the **show ip protocols** command, as shown in Example 14-33.

Key Topic

Example 14-33 Verifying Variance and Maximum Paths

```
Router#show ip protocols
Routing Protocol is "eigrp 100"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP-IPv4 Protocol for AS(100)
    Metric weight K1=1, K2=0, K3=1, K4=0, K5=0
    NSF-aware route hold timer is 240
    Router-ID: 10.1.12.1
    Topology : 0 (base)
      Active Timer: 3 min
      Distance: internal 90 external 170
      Maximum path: 4
      Maximum hopcount 100
      Maximum metric variance 1

  Automatic Summarization: disabled
  Maximum path: 4
  Routing for Networks:
    0.0.0.0
  Routing Information Sources:
    Gateway         Distance      Last Update
    Gateway         Distance      Last Update
    10.1.12.2        90           10:26:36
  Distance: internal 90 external 170
```

EIGRP for IPv4 Trouble Tickets

This section presents various trouble tickets relating to the topics discussed earlier in the chapter. The purpose of these trouble tickets is to give a process that you can follow when troubleshooting in the real world or in an exam environment. All trouble tickets in this section are based on the topology depicted in Figure 14-7.

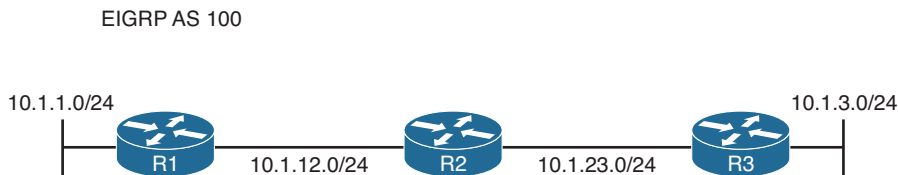


Figure 14-7 EIGRP for IPv4 Trouble Tickets Topology

Trouble Ticket 14-1

Problem: Users in the 10.1.1.0/24 network indicate that they are not able to access resources in the 10.1.3.0/24 network.

As always, the first item on the list for troubleshooting is: verify the problem. You access a PC in the 10.1.1.0/24 network and ping an IP address in the 10.1.3.0/24 network and it is successful (0% loss), as shown in Example 14-34. However, notice that the reply is from the default gateway at 10.1.1.1 and it states: Destination host unreachable. Therefore, it was technically not successful.

Example 14-34 Destination Unreachable Result from ping Command on PC

```

C:\>ping 10.1.3.10

Pinging 10.1.3.10 with 32 bytes of data:

Reply from 10.1.1.1: Destination host unreachable.
Reply from 10.1.1.1: Destination host unreachable.
Reply from 10.1.1.1: Destination host unreachable.
Reply from 10.1.1.1: Destination host unreachable.

Ping statistics for 10.1.3.10:
    Packets: Sent = 4, Received = 4, lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
  
```

The result of this ping tells us two very important things: The PC can reach the default gateway, and the default gateway does not know how to get to the 10.1.3.0/24 network. Therefore, we can focus our attention on R1 and work from there.

On R1, you issue the same ping, but it fails, as shown in Example 14-35.

Example 14-35 *Failed Ping from R1 to 10.1.3.10*

```
R1#ping 10.1.3.10
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.3.10, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

Next, you check R1's routing table with the **show ip route** command and notice that there are only connected routes in the routing table, as shown in Example 14-36. You conclude that R1 is not learning any routes from R2.

Example 14-36 *show ip route Output on R1*

```
R1#show ip route
...output omitted...
Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C       10.1.1.0/24 is directly connected, GigabitEthernet0/0
L       10.1.1.1/32 is directly connected, GigabitEthernet0/0
C       10.1.12.0/24 is directly connected, GigabitEthernet1/0
L       10.1.12.1/32 is directly connected, GigabitEthernet1/0
```

According to Figure 14-7, EIGRP is the routing protocol in use. Therefore, you issue the **show ip protocols** command to verify that EIGRP is running the correct autonomous system. Example 14-37 displays the **show ip protocols** output and confirms that EIGRP 100 is in operation on R1.

Example 14-37 *show ip protocols Output on R1*

```
R1#show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "eigrp 100"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP-IPv4 Protocol for AS(100)
    Metric weight K1=1, K2=0, K3=1, K4=0, K5=0
    NSF-aware route hold timer is 240
    Router-ID: 10.1.12.1
    Topology : 0 (base)
      Active Timer: 3 min
      Distance: internal 90 external 170
      Maximum path: 4
      Maximum hopcount 100
      Maximum metric variance 1
```



```
Automatic Summarization: disabled
Maximum path: 4
Routing for Networks:
  10.1.1.1/32
  10.1.12.1/32
Routing Information Sources:
  Gateway          Distance      Last Update
  10.1.12.2         90           00:45:53
Distance: internal 90 external 170
```

Next you check to see whether R1 has any EIGRP neighbors. According to the topology, R2 should be a neighbor. To verify EIGRP neighbors, you issue the **show ip eigrp neighbors** command on R1, as shown in Example 14-38. According to the output, R1 has no neighbors.

Example 14-38 show ip eigrp neighbors Output on R1

```
R1#show ip eigrp neighbors
EIGRP-IPv4 Neighbors for AS(100)
```

Now you verify whether there are any interfaces participating in the EIGRP process using the **show ip eigrp interfaces** command. Example 14-39 indicates that there are two interfaces participating in the EIGRP process: Gig0/0 and Gig1/0.

Example 14-39 show ip eigrp interfaces Output on R1

```
R1#show ip eigrp interfaces
EIGRP-IPv4 Interfaces for AS(100)
```

Interface	Peers	Xmit Queue Un/Reliable	PeerQ Un/Reliable	Mean SRTT	Pacing Time Un/Reliable	Multicast Flow Timer	Pending Routes
Gig0/0	0	0/0	0/0	0	0/0	0	0
Gig1/0	0	0/0	0/0	0	0/0	304	0

The output of **show cdp neighbors**, as shown in Example 14-40, indicates that R1 is connected to R2 using Gig1/0 and that R2 is using Gig0/0. Therefore, we expect a peering between the two using these interfaces.

Example 14-40 show cdp neighbors Output on R1

```
R1#show cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone,
                  D - Remote, C - CVTA, M - Two-port Mac Relay
```

Device ID	Local Intrfce	Holdtme	Capability	Platform	Port ID
R2	Gig 1/0	172	R	7206VXR	Gig 0/0

Now is a great time to verify whether Gig0/0 on R2 is participating in the EIGRP process. On R2, you issue the **show ip eigrp interfaces** command, as shown in Example 14-41.

Example 14-41 show ip eigrp interfaces *Output on R2*

```
R2#show ip eigrp interfaces
```

EIGRP-IPv4 Interfaces for AS(100)

Interface	Peers	Xmit Queue Un/Reliable	PeerQ Un/Reliable	Mean SRTT	Pacing Time Un/Reliable	Multicast Flow Timer	Pending Routes
Gil/0	0	0/0	0/0	0	0/0	448	0

Example 14-41 confirms that R2's interface Gig0/0 is not participating in the EIGRP process.

You review the output of **show run | section router eigrp** and **show ip interface brief** on R2, as shown in Example 14-42, and confirm that the wrong network statement was issued on R2. The **network** statement **network 10.1.21.2 0.0.0.0** enables the EIGRP process on the interface with that IP address. According to the output of **show ip interface brief**, the **network** statement should be **network 10.1.12.2 0.0.0.0**.

Example 14-42 show run | section router eigrp *Output on R2 and Verifying Interface IP address*

```
R2#show run | section router eigrp
```

```
router eigrp 100
network 10.1.21.2 0.0.0.0
network 10.1.23.2 0.0.0.0
```



```
R2#show ip interface brief
```

Interface	IP-Address	OK?	Method	Status	Protocol
Ethernet0/0	unassigned	YES	unset	administratively down	down
GigabitEthernet0/0	10.1.12.2	YES	manual	up	up
GigabitEthernet1/0	10.1.23.2	YES	manual	up	up

To fix this issue, on R2 you execute the **no network 10.1.21.2 0.0.0.0** command and enter the **network 10.1.12.2 0.0.0.0** command in router EIGRP configuration mode instead. After you have done this, the neighbor relationship forms, as shown with the following syslog messages:

R1#

%DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 10.1.12.2 (GigabitEthernet1/0) is up: new adjacency

R2#

%DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 10.1.12.1 (GigabitEthernet0/0) is up: new adjacency

You confirm the neighbor relationship on R1 with the **show ip eigrp neighbors** command, as shown in Example 14-43.

Example 14-43 *Verifying Neighbors with the show ip eigrp neighbors Command*

```
R1#show ip eigrp neighbors
EIGRP-IPv4 Neighbors for AS(100)
H   Address                  Interface                Hold Uptime   SRTT   RTO   Q   Seq
                               (sec)          (ms)        Cnt  Num
0   10.1.12.2                 Gi1/0                 14  00:02:10   75   450   0   12
```

You go back to the PC and ping the same IP address to confirm the problem is solved, and you receive the same result, as shown in Example 14-44. R1 still does not know about the 10.1.3.0/24 network.

Example 14-44 *Destination Unreachable from ping Command on PC*

```
C:\>ping 10.1.3.10

Pinging 10.1.3.10 with 32 bytes of data:

Reply from 10.1.1.1: Destination host unreachable.
Reply from 10.1.1.1: Destination host unreachable.
Reply from 10.1.1.1: Destination host unreachable.
Reply from 10.1.1.1: Destination host unreachable.

Ping statistics for 10.1.3.10:
    Packets: Sent = 4, Received = 4, lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Back on R1, you issue the **show ip route** command, as shown in Example 14-45. R1 is receiving EIGRP routes because there is now an EIGRP route (D) in the routing table. However, R1 still does not know about the 10.1.3.0/24 network.

Example 14-45 *show ip route Output After Neighbor Relationship with R2 Established*

```
R1#show ip route
...output omitted...
Gateway of last resort is not set

    10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
C       10.1.1.0/24 is directly connected, GigabitEthernet0/0
L       10.1.1.1/32 is directly connected, GigabitEthernet0/0
C       10.1.12.0/24 is directly connected, GigabitEthernet1/0
L       10.1.12.1/32 is directly connected, GigabitEthernet1/0
D       10.1.23.0/24 [90/3072] via 10.1.12.2, 00:07:40, GigabitEthernet1/0
```

Does R2 know about the 10.1.3.0/24 network? Example 14-46 displays R2's routing table, and it is missing 10.1.3.0/24 as well.

Example 14-46 show ip route *Output on R2*

```

R2#show ip route
...output omitted...
Gateway of last resort is not set

    10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
D       10.1.1.0/24 [90/3072] via 10.1.12.1, 00:12:11, GigabitEthernet0/0
C       10.1.12.0/24 is directly connected, GigabitEthernet0/0
L       10.1.12.2/32 is directly connected, GigabitEthernet0/0
C       10.1.23.0/24 is directly connected, GigabitEthernet1/0
L       10.1.23.2/32 is directly connected, GigabitEthernet1/0

```

For R2 to learn about the network, it has to be neighbors with R3. Reviewing the R2 output of **show ip eigrp neighbors** in Example 14-47 indicates that R3 is not a neighbor, only R1.

Example 14-47 show ip eigrp neighbors *on R2*

```

R2#show ip eigrp neighbors
EIGRP-IPv4 Neighbors for AS(100)

```

H	Address	Interface	Hold Uptime (sec)	SRTT (ms)	RTO	Q Cnt	Seq Num
0	10.1.12.1	Gi0/0	11 00:17:28	65	390	0	7

Previously, Example 14-41 indicated that Gig1/0 on R2 was participating in the EIGRP process. Therefore, we should look at the interfaces on R3. According to the output in Example 14-48, both interfaces on R3 are participating in the EIGRP process for autonomous system 10.

Example 14-48 show ip eigrp interfaces *on R3*

```

R3#show ip eigrp interfaces
EIGRP-IPv4 Interfaces for AS(10)

```

Interface	Peers	Xmit Queue Un/Reliable	PeerQ Un/Reliable	Mean SRTT	Pacing Time Un/Reliable	Multicast Flow Timer	Pending Routes
Gi0/0	0	0/0	0/0	0	0/0	0	0
Gi1/0	0	0/0	0/0	0	0/0	0	0

Did you spot the issue? If not, look again at Example 14-48. If you need to compare it to Example 14-47, do so.

The autonomous system numbers do not match, and to form an EIGRP neighbor relationship, the autonomous system numbers must match. To solve this issue, you must enable EIGRP autonomous system 100 on R3, and then provide the correct **network** statements to enable EIGRP on the required interfaces for autonomous system 100. You should also remove any EIGRP configs that are not needed, such as the EIGRP autonomous system 10 configurations. Example 14-49 provides the commands needed to accomplish this.

Example 14-49 *R3 Configurations Required to Solve Issue*

```

R3#config t
Enter configuration commands, one per line. End with CNTL/Z.
R3(config)#no router eigrp 10
R3(config)#router eigrp 100
R3(config-router)#network 10.1.3.3 0.0.0.0
R3(config-router)#network 10.1.23.3 0.0.0.0
%DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 10.1.23.2 (GigabitEthernet1/0) is up:
new adjacency
R3(config-router)#

```

Notice in Example 14-49 that the neighbor relationship with R2 was successful. Now it is time to verify that all our issues are solved. On R2, you issue the **show ip route** command, as shown in Example 14-50, and notice that the 10.1.3.0/24 network is present. You also issue the same command on R1 and notice that 10.1.3.0/24 is present, as shown in Example 14-51. You then ping from the PC again, and the ping is truly successful, as shown in Example 14-52.

Example 14-50 *show ip route Output on R2*

```

R2#show ip route
...output omitted...

Gateway of last resort is not set

    10.0.0.0/8 is variably subnetted, 6 subnets, 2 masks
D       10.1.1.0/24 [90/3072] via 10.1.12.1, 00:37:21, GigabitEthernet0/0
D       10.1.3.0/24 [90/3072] via 10.1.23.3, 00:06:16, GigabitEthernet1/0
C       10.1.12.0/24 is directly connected, GigabitEthernet0/0
L       10.1.12.2/32 is directly connected, GigabitEthernet0/0
C       10.1.23.0/24 is directly connected, GigabitEthernet1/0
L       10.1.23.2/32 is directly connected, GigabitEthernet1/0

```

Example 14-51 *show ip route Output on R1*

```

R1#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

    10.0.0.0/8 is variably subnetted, 6 subnets, 2 masks

```

```

C      10.1.1.0/24 is directly connected, GigabitEthernet0/0
L      10.1.1.1/32 is directly connected, GigabitEthernet0/0
D      10.1.3.0/24 [90/3328] via 10.1.12.2, 00:07:08, GigabitEthernet1/0
C      10.1.12.0/24 is directly connected, GigabitEthernet1/0
L      10.1.12.1/32 is directly connected, GigabitEthernet1/0
D      10.1.23.0/24 [90/3072] via 10.1.12.2, 00:38:12, GigabitEthernet1/0

```

Example 14-52 *A Successful Ping from the 10.1.1.0/24 Network to the 10.1.3.0/24 Network*

```

C:\>ping 10.1.3.10

Pinging 10.1.3.10 with 32 bytes of data:

Reply from 10.1.3.10: bytes=32 time 1ms TTL=128
Reply from 10.1.3.10: bytes=32 time 1ms TTL=128
Reply from 10.1.3.10: bytes=32 time 1ms TTL=128
Reply from 10.1.3.10: bytes=32 time 1ms TTL=128

Ping statistics for 10.1.3.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

```

Trouble Ticket 14-2

Problem: Users in the 10.1.1.0/24 network have indicated that they are not able to access resources in 10.1.3.0/24.

To begin, you verify the problem by pinging from a PC in the 10.1.1.0/24 network to a PC in the 10.1.3.0/24 network, as shown in Example 14-53, and it fails. Notice that the reply is from the default gateway at 10.1.1.1 and it states: Destination host unreachable. Therefore, it was technically not successful.

Example 14-53 *Destination Unreachable Result from ping Command on PC*

```

C:\>ping 10.1.3.10

Pinging 10.1.3.10 with 32 bytes of data:

Reply from 10.1.1.1: Destination host unreachable.
Reply from 10.1.1.1: Destination host unreachable.
Reply from 10.1.1.1: Destination host unreachable.
Reply from 10.1.1.1: Destination host unreachable.

Ping statistics for 10.1.3.10:
    Packets: Sent = 4, Received = 4, lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

```

The result of this ping tells us two very important things: the PC can reach the default gateway, and the default gateway does not know how to get to the 10.1.3.0/24 network. Therefore, we can focus our attention on R1 and work from there.

On R1, you issue the same ping, but it fails, as shown in Example 14-54.

Example 14-54 *Failed Ping from R1 to 10.1.3.10*

```
R1#ping 10.1.3.10
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.3.10, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

Next you check the routing table on R1 with the **show ip route 10.1.3.0 255.255.255.0** command, as shown in Example 14-55, and it states: % Subnet not in table.

Example 14-55 *Determining Whether a Route Is in R1's Routing Table*

```
R1#show ip route 10.1.3.0 255.255.255.0
% Subnet not in table
```

Does R2 know about it? You go to R2 and issue the same command, as shown in Example 14-56. The result is the same: % Subnet not in table.

Example 14-56 *Determining Whether a Route Is in R2's Routing Table*

```
R2#show ip route 10.1.3.0 255.255.255.0
% Subnet not in table
```

Next you go to R3 and issue the same command. Notice that 10.1.3.0/24 is in the routing table as a connected route, as shown in Example 14-57.

Example 14-57 *Determining Whether Route Is in R3's Routing Table*

```
R3#show ip route 10.1.3.0 255.255.255.0
Routing entry for 10.1.3.0/24
  Known via "connected", distance 0, metric 0 (connected, via interface)
  Redistributing via eigrp 100
  Routing Descriptor Blocks:
  * directly connected, via GigabitEthernet0/0
    Route metric is 0, traffic share count is 1
```

What would prevent a connected route from being advertised via EIGRP to a neighbor? Answer: The interface is not participating in the EIGRP process. Now you check the EIGRP interface table on R3 with the **show ip eigrp interfaces** command. Example 14-58 indicates that only Gigabit Ethernet 1/0 is participating in the EIGRP process.

Example 14-58 *Determining Whether an Interface Is Participating in the EIGRP Process*

```
R3#show ip eigrp interfaces
EIGRP-IPv4 Interfaces for AS(100)
```

	Xmit Queue	PeerQ	Mean	Pacing Time	Multicast	Pending	
Interface	Peers	Un/Reliable	Un/Reliable	SRTT	Un/Reliable	Flow Timer	Routes
Gi1/0	1	0/0	0/0	821	0/0	4080	0

However, do not jump to the conclusion that this interface is not participating in the EIGRP process. Remember that EIGRP passive interfaces do not appear in this output. Therefore, check the output of **show ip protocols** for passive interfaces. In Example 14-59, you can see that there are no passive interfaces.

Example 14-59 *Determining Whether an Interface Is Passive*

```
R3#show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "eigrp 100"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
EIGRP-IPv4 Protocol for AS(100)
  Metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  NSF-aware route hold timer is 240
  Router-ID: 10.1.23.3
  Topology : 0 (base)
    Active Timer: 3 min
    Distance: internal 90 external 170
    Maximum path: 4
    Maximum hopcount 100
    Maximum metric variance 1

Automatic Summarization: disabled
Maximum path: 4
Routing for Networks:
  10.1.3.0/32
  10.1.23.3/32
Routing Information Sources:
  Gateway         Distance      Last Update
  10.1.23.2        90           00:19:11
Distance: internal 90 external 170
```


Now you need to make sure that there is a **network** statement that will enable the EIGRP process on the interface connected to the 10.1.3.0/24 network. In Example 14-59, the output of **show ip protocols** indicates that R3 is routing for networks 10.1.3.0/32. Remember from our discussion earlier that this really means **network 10.1.3.0 0.0.0.0**. As a result, EIGRP will be enabled on the interface with the IP address 10.1.3.0. Example 14-60, which displays the output of **show ip interface brief**, shows that there are no interfaces with that IP address. Interface Gig0/0 has an IP address of 10.1.3.3. Therefore, the **network** statement is incorrect, as shown in the output of **show run | section router eigrp** in Example 14-61.

Example 14-60 *Reviewing the Interface IP Addresses*

```
R3#show ip interface brief
```

Interface	IP-Address	OK?	Method	Status	Protocol
Ethernet0/0	unassigned	YES	NVRAM	administratively down	down
GigabitEthernet0/0	10.1.3.3	YES	NVRAM	up	up
GigabitEthernet1/0	10.1.23.3	YES	NVRAM	up	up

Example 14-61 *Reviewing the network Statements in the Running Config*

```
R3#show run | section router eigrp
router eigrp 100
network 10.1.3.0 0.0.0.0
network 10.1.23.3 0.0.0.0
```

After fixing the issue with the **no network 10.1.3.0 0.0.0.0** command and the **network 10.1.3.3 0.0.0.0** command, you check R1's routing table with the command **show ip route 10.1.3.0 255.255.255.0**. As shown in Example 14-62, 10.1.3.0/24 is now in the routing table and can be reached via the next hop 10.1.12.2.

Example 14-62 *Verifying 10.1.3.0/24 Is in R1's Routing Table*

```
R1#show ip route 10.1.3.0 255.255.255.0
Routing entry for 10.1.3.0/24
  Known via "eigrp 100", distance 90, metric 3328, type internal
  Redistributing via eigrp 100
  Last update from 10.1.12.2 on GigabitEthernet1/0, 00:00:06 ago
  Routing Descriptor Blocks:
  * 10.1.12.2, from 10.1.12.2, 00:00:06 ago, via GigabitEthernet1/0
    Route metric is 3328, traffic share count is 1
    Total delay is 30 microseconds, minimum bandwidth is 1000000 Kbit
    Reliability 255/255, minimum MTU 1500 bytes
    Loading 1/255, Hops 2
```

Finally, you ping from the PC again, and the ping is successful, as shown in Example 14-63.

Example 14-63 *A Successful Ping from the 10.1.1.0/24 Network to the 10.1.3.0/24 Network*

```

C:\>ping 10.1.3.10

Pinging 10.1.3.10 with 32 bytes of data:

Reply from 10.1.3.10: bytes=32 time 1ms TTL=128
Reply from 10.1.3.10: bytes=32 time 1ms TTL=128
Reply from 10.1.3.10: bytes=32 time 1ms TTL=128
Reply from 10.1.3.10: bytes=32 time 1ms TTL=128

Ping statistics for 10.1.3.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

```

Trouble Ticket 14-3

Problem: Users in the 10.1.1.0/24 network have indicated that they are not able to access resources in 10.1.3.0/24.

To begin, you verify the problem by pinging from a PC in the 10.1.1.0/24 network to a PC in the 10.1.3.0/24 network, as shown in Example 14-64, and it fails. Notice that the reply is from the default gateway at 10.1.1.1 and it states: Destination host unreachable.

Example 14-64 *Destination Unreachable Result from ping Command on PC*

```

C:\>ping 10.1.3.10

Pinging 10.1.3.10 with 32 bytes of data;

Reply from 10.1.1.1: Destination host unreachable.
Reply from 10.1.1.1: Destination host unreachable.
Reply from 10.1.1.1: Destination host unreachable.
Reply from 10.1.1.1: Destination host unreachable.

Ping statistics for 10.1.3.10:
    Packets: Sent = 4, Received = 4, lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

```

The result of this ping tells us two very important things: The PC can reach the default gateway, and the default gateway does not know how to get to the 10.1.3.0/24 network. Therefore, we can focus our attention on R1 and work from there.

On R1, you issue the same ping, but it fails, as shown in Example 14-65.

Example 14-65 *Failed Ping from R1 to 10.1.3.10*

```
R1#ping 10.1.3.10
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.3.10, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

Next you check the routing table on R1 with the **show ip route 10.1.3.0 255.255.255.0** command, as shown in Example 14-66, and it states: % Subnet not in table.

Example 14-66 *Determining Whether a Route Is in R1's Routing Table*

```
R1#show ip route 10.1.3.0 255.255.255.0
% Subnet not in table
```

Does R2 know about it? You go to R2 and issue the same command, as shown in Example 14-67. R2 does know about it.

Example 14-67 *Determining Whether a Route Is in R2's Routing Table*

```
R2#show ip route 10.1.3.0 255.255.255.0
Routing entry for 10.1.3.0/24
  Known via "eigrp 100", distance 90, metric 3072, type internal
  Redistributing via eigrp 100
  Last update from 10.1.23.3 on GigabitEthernet1/0, 00:44:37 ago
  Routing Descriptor Blocks:
  * 10.1.23.3, from 10.1.23.3, 00:44:37 ago, via GigabitEthernet1/0
    Route metric is 3072, traffic share count is 1
    Total delay is 20 microseconds, minimum bandwidth is 1000000 Kbit
    Reliability 255/255, minimum MTU 1500 bytes
    Loading 1/255, Hops 1
```

Next you go back to R1 and issue the **show ip eigrp topology** command to determine whether R1 is even learning about the 10.1.3.0/24 network. Example 14-68 indicates that it is not.

Example 14-68 *Determining Whether R1 Is Learning About 10.1.3.0/24*

```
R1#show ip eigrp topology
EIGRP-IPv4 Topology Table for AS(100)/ID(10.1.12.1)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 10.1.12.0/24, 1 successors, FD is 2816
    via Connected, GigabitEthernet1/0
P 10.1.23.0/24, 1 successors, FD is 3072
    via 10.1.12.2 (3072/2816), GigabitEthernet1/0
P 10.1.1.0/24, 1 successors, FD is 2816
    via Connected, GigabitEthernet0/0
```

Time to hypothesize! Why would R2 know about 10.1.3.0/24 and R1 not know about it?

- R1 and R2 are not EIGRP neighbors.
- A route filter on R2 prevents it from advertising 10.1.3.0/24 to R1.
- A route filter on R1 prevents it from learning 10.1.3.0/24 in Gig1/0.

On R1, you issue the **show ip eigrp neighbors** command, as shown in Example 14-69, and it shows that R2 is a neighbor. However, if you looked closely at the topology table of R1, you would have noticed that R1 is learning about 10.1.23.0/24 from R2, meaning that they are neighbors and that routes are being learned. Therefore, you hypothesize that there must be a filter in place.

Example 14-69 *Determining Whether R2 Is a Neighbor*

```
R1#show ip eigrp neighbors
EIGRP-IPv4 Neighbors for AS(100)
```

H	Address	Interface	Hold (sec)	Uptime	SRTT (ms)	RTO	Q Cnt	Seq Num
0	10.1.12.2	Gig1/0	12	01:20:27	72	432	0	18

Next you issue the **show ip protocols** command, as shown in Example 14-70, to determine whether there are any route filters on R1. The output indicates that there is an inbound route filter on R1 Gigabit Ethernet 1/0. The route filter is filtering based on a prefix-list called DENY_10.1.3.0/24.

Example 14-70 *Determining Whether There Is a Route Filter on R1*

```
R1#show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "eigrp 100"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  GigabitEthernet1/0 filtered by (prefix-list) DENY_10.1.3.0/24 (per-user),
  default is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP-IPv4 Protocol for AS(100)
  ...output omitted...
```

Now you issue the **show ip prefix-list** command on R1, as shown in Example 14-71, and it indicates that 10.1.3.0/24 is being denied.

Example 14-71 *Reviewing the Prefix List*

```
R1#show ip prefix-list
ip prefix-list DENY_10.1.3.0/24: 2 entries
seq 5 deny 10.1.3.0/24
seq 10 permit 0.0.0.0/0 le 32
```

In this case, you can either modify the prefix-list to allow 10.1.3.0/24 or you can remove the distribute list from the EIGRP process. It would all depend on the requirements of the organization or scenario. In this case, we remove the distribute list from R1 with the **no distribute-list prefix DENY_10.1.3.0/24 in GigabitEthernet1/0** command. Because of this change, the neighbor relationship resets, as the following syslog message indicates:

```
%DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 10.1.12.2 (GigabitEthernet1/0)
is resync: intf route configuration changed
```

After fixing the issue, you check R1's routing table with the command **show ip route 10.1.3.0 255.255.255.0**. As shown in Example 14-72, 10.1.3.0/24 is now in the routing table and can be reached via the next hop 10.1.12.2.

Example 14-72 *Verifying That 10.1.3.0/24 Is in R1's Routing Table*

```
R1#show ip route 10.1.3.0 255.255.255.0
Routing entry for 10.1.3.0/24
  Known via "eigrp 100", distance 90, metric 3328, type internal
  Redistributing via eigrp 100
  Last update from 10.1.12.2 on GigabitEthernet1/0, 00:00:06 ago
  Routing Descriptor Blocks:
  * 10.1.12.2, from 10.1.12.2, 00:00:06 ago, via GigabitEthernet1/0
    Route metric is 3328, traffic share count is 1
    Total delay is 30 microseconds, minimum bandwidth is 1000000 Kbit
    Reliability 255/255, minimum MTU 1500 bytes
    Loading 1/255, Hops 2
```

Finally, you ping from the PC again, and the ping is successful, as shown in Example 14-73.

Example 14-73 *A Successful Ping from the 10.1.1.0/24 Network to the 10.1.3.0/24 Network*

```
C:\>ping 10.1.3.10

Pinging 10.1.3.10 with 32 bytes of data:

Reply from 10.1.3.10: bytes=32 time 1ms TTL=128
Reply from 10.1.3.10: bytes=32 time 1ms TTL=128
Reply from 10.1.3.10: bytes=32 time 1ms TTL=128
Reply from 10.1.3.10: bytes=32 time 1ms TTL=128

Ping statistics for 10.1.3.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Troubleshooting EIGRP for IPv6

Because EIGRP for IPv6 is based on EIGRP for IPv4, you will be dealing with very similar issues when it comes to troubleshooting, with a few minor differences based on IPv6. This should come as a relief, knowing that you do not have to learn a large amount of new information for EIGRP for IPv6. However, you do need to know the **show** commands that will display the information you need to troubleshoot any given EIGRP for IPv6-related issue.

This section explains the same issues presented in the previous section; however, the focus is on the **show** commands that are used when troubleshooting EIGRP for IPv6-related issues.

Troubleshooting EIGRP for IPv6 Neighbor Issues

The neighbor issues are mostly the same except for a few differences based on the way EIGRP for IPv6 is enabled on an interface. To verify EIGRP for IPv6 neighbors, use the **show ipv6 eigrp neighbors** command, as shown in Example 14-74. Notice how EIGRP for IPv6 neighbors are identified by their link-local address. In this case, R2 is neighbors with two routers. One is reachable out Gig1/0, and the other is reachable out Gig0/0.



Example 14-74 Verifying EIGRP for IPv6 Neighbors

```
R2#show ipv6 eigrp neighbors
EIGRP-IPv6 Neighbors for AS(100)
```

H	Address	Interface	Hold (sec)	Uptime	SRTT (ms)	RTO	Q Cnt	Seq Num
1	Link-local address: FE80::C823:17FF:FE0C:1C	Gig1/0	10	00:17:59	320	2880	0	4
0	Link-local address: FE80::C820:17FF:FE04:1C	Gig0/0	12	00:18:01	148	888	0	3

Interface Is Down

To verify that an interface is up, you use the **show ipv6 interface brief** command, as shown in Example 14-75. In this example, Gig0/0 and 1/0 are up/up, and Gig2/0 is administratively down/down. This indicates that Gig2/0 has been configured with the **shutdown** command.

Example 14-75 Verifying the Status of IPv6 Interfaces

```
R1#show ipv6 interface brief
GigabitEthernet0/0    [up/up]
    FE80::C80E:1FF:FE9C:8
    2001:DB8:0:1::1
GigabitEthernet1/0    [up/up]
    FE80::C80E:1FF:FE9C:1C
    2001:DB8:0:12::1
```

```
GigabitEthernet2/0    [administratively down/down]
FE80::C80E:1FF:FE9C:38
2001:DB8:0:13::1
```

Mismatched Autonomous System Numbers

To verify the autonomous system number being used, you can use the **show ipv6 protocols** command as shown in Example 14-76. In this example, the EIGRP autonomous system is 100.

Mismatched K Values

You can verify the EIGRP for IPv6 K values with **show ipv6 protocols**, as shown in Example 14-76. In this example, the K values are 1, 0, 1, 0, 0, which are the defaults.

Passive Interfaces

Router interfaces participating in the EIGRP for IPv6 autonomous system that are passive can be verified with the **show ipv6 protocols** command, as shown in Example 14-76. In this example, Gigabit Ethernet 0/0 is a passive interface.



Example 14-76 Verifying EIGRP for IPv6 Configurations with show ipv6 protocols

```
R1#show ipv6 protocols
...output omitted...
IPv6 Routing Protocol is "eigrp 100"
EIGRP-IPv6 Protocol for AS(100)
Metric weight K1=1, K2=0, K3=1, K4=0, K5=0
NSF-aware route hold timer is 240
Router-ID: 10.1.12.1
Topology : 0 (base)
Active Timer: 3 min
Distance: internal 90 external 170
Maximum path: 16
Maximum hopcount 100
Maximum metric variance 1

Interfaces:
GigabitEthernet1/0
GigabitEthernet0/0 (passive)

Redistribution:
None
```

Mismatched Authentication

If authentication is being used, the key ID and key string must match, in addition to when the key is valid (if configured) between neighbors. Example 14-77 displays how to

verify whether an interface is enabled for EIGRP for IPv6 authentication with the **show ipv6 eigrp interfaces detail** command and how to verify the configuration of the key chain that is being used with the **show key chain** command. In this example, the authentication mode is MD5, and the key chain of TEST is being used.

Example 14-77 Verifying EIGRP for IPv6 Authentication

```
R1#show ipv6 eigrp interfaces detail
EIGRP-IPv6 Interfaces for AS(100)

      Xmit Queue   PeerQ      Mean    Pacing Time   Multicast    Pending
Interface Peers  Un/Reliable  Un/Reliable SRTT    Un/Reliable   Flow Timer   Routes
Gi1/0      1      0/0         0/0       72      0/0          316         0

Hello-interval is 5, Hold-time is 15
Split-horizon is enabled
Next xmit serial <none>
Packetized sent/expedited: 5/0
Hello's sent/expedited: 494/6
Un/reliable mcasts: 0/4  Un/reliable ucasts: 4/59
Mcast exceptions: 0  CR packets: 0  ACKs suppressed: 0
Retransmissions sent: 54  Out-of-sequence rcvd: 3
Topology-ids on interface - 0
Authentication mode is md5, key-chain is "TEST"

R1#show key chain
Key-chain TEST:
  key 1 -- text "TEST"
    accept lifetime (always valid) - (always valid) [valid now]
    send lifetime (always valid) - (always valid) [valid now]
```

Timers

Timers do not have to match; however, if they are not configured appropriately, neighbor relationships might flap. You can verify timers with the **show ipv6 eigrp interfaces detail** command, as shown in Example 14-77. In that example, the hello interval is configured as 5, and the hold interval is 15, which are the defaults.

Interface Not Participating in Routing Process



With EIGRP for IPv6, the interfaces are enabled for the routing process with the **ipv6 eigrp autonomous_system_number** interface configuration command. There are two **show** commands that you can use to verify the interfaces that are participating in the routing process, as shown in Example 14-78: **show ipv6 eigrp interfaces** and **show ipv6 protocols**. As with EIGRP for IPv4, the **show ipv6 eigrp interfaces** command does not show passive interfaces. However, **show ipv6 protocols** does.

Example 14-78 *Verifying EIGRP for IPv6 Interfaces*

```

R1#show ipv6 eigrp interfaces
EIGRP-IPv6 Interfaces for AS(100)

      Xmit Queue   PeerQ      Mean    Pacing Time   Multicast    Pending
Interface  Peers Un/Reliable Un/Reliable SRTT      Un/Reliable   Flow Timer   Routes
-----
Gi1/0      1      0/0        0/0       282       0/0          1348         0

R1#show ipv6 protocols
IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "ND"
IPv6 Routing Protocol is "eigrp 100"
EIGRP-IPv6 Protocol for AS(100)
  Metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  NSF-aware route hold timer is 240
...output omitted...
Interfaces:
  GigabitEthernet1/0
  GigabitEthernet0/0 (passive)
Redistribution:
  None

```

ACLs

EIGRP for IPv6 uses the IPv6 multicast address FF02::A to form neighbor adjacencies. If an IPv6 ACL is denying packets destined to the multicast address FF02::A, neighbor adjacencies will not form. In addition, because neighbor adjacencies are formed with link-local addresses, if the link-local address range is denied based on source or destination IPv6 address in an interface with an IPv6 ACL, neighbor relationships will not form.

Troubleshooting EIGRP for IPv6 Route

The reasons why a route might be missing and the steps used to troubleshoot them with EIGRP for IPv6 is similar to our previous discussions based on EIGRP for IPv4. Therefore, we will identify some of the more common issues here and review the **show** commands that you can use to identify them.

Interface Not Participating in Routing Process

For a network to be advertised by the EIGRP for IPv6 process, the interface associated with that network must be participating in the routing process. As displayed earlier in Example 14-78, you can use the commands **show ipv6 eigrp interfaces** and **show ipv6 protocols** to verify the interfaces participating in the process.

Better Source of Information

If the exact same network is learned from a more reliable source, it is used instead of the EIGRP for IPv6-learned information. To verify the AD associated with the route in the routing table, you can issue the **show ipv6 route ipv6_address/prefix** command. In Example 14-79, the 2001:db8:0:1::/64 network has an AD of 90, and it was learned via EIGRP autonomous system 100.

Example 14-79 Verifying AD of IPv6 Routes

```
R2#show ipv6 route 2001:DB8:0:1::/64
Routing entry for 2001:DB8:0:1::/64
  Known via "eigrp 100", distance 90, metric 3072, type internal
  Route count is 1/1, share count 0
  Routing paths:
    FE80::C820:17FF:FE04:1C, GigabitEthernet0/0
    Last updated 00:25:27 ago
```

Route Filtering

A filter might be set up that is preventing a route from being advertised or learned. With EIGRP for IPv6, the **distribute-list prefix-list** command is used to configure a route filter. To verify the filter applied, use the **show run | section ipv6 router eigrp** command. In Example 14-80, a distribute list is using a prefix list called TSHOOT_EIGRP to filter routes inbound on Gigabit Ethernet 1/0. To successfully troubleshoot route filtering issues, you also need to verify the IPv6 prefix list using the **show ipv6 prefix-list** command.

Example 14-80 Verifying EIGRP for IPv6 Distribute List

```
R1#show run | section ipv6 router eigrp
ipv6 router eigrp 100
  distribute-list prefix-list TSHOOT_EIGRP in GigabitEthernet1/0
  passive-interface default
  no passive-interface GigabitEthernet1/0
```

Stub Configuration



If the wrong router is configured as a stub router, or the wrong setting is chosen during the stub router configuration, you might prevent a network from being advertised that should be advertised. When troubleshooting EIGRP for IPv6 stub configurations, you can use the **show ipv6 protocols** command to verify whether the local router is a stub router and the networks that it is advertising, as shown in Example 14-81. On a remote router, you can issue the **show ipv6 eigrp neighbors detail** command, as shown in Example 14-82. In this case, R1 is a stub router advertising connected and summary routes.

Example 14-81 *Verifying EIGRP Stub Configuration on a Stub Router*

```

R1#show ipv6 protocols
IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "ND"
IPv6 Routing Protocol is "eigrp 100"
EIGRP-IPv6 Protocol for AS(100)
  Metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  NSF-aware route hold timer is 240
  Router-ID: 10.1.12.1
  Stub, connected, summary
  Topology : 0 (base)
    Active Timer: 3 min
    Distance: internal 90 external 170
    Maximum path: 16
    Maximum hopcount 100
    Maximum metric variance 1

  Interfaces:
    GigabitEthernet1/0
    GigabitEthernet0/0 (passive)
  Redistribution:
    None

```

Example 14-82 *Verifying EIGRP Stub Configuration of Neighbor Router*

```

R2#show ipv6 eigrp neighbors detail
EIGRP-IPv6 Neighbors for AS(100)

```

H	Address	Interface	Hold (sec)	Uptime	SRTT (ms)	RTO	Q Cnt	Seq Num
0	Link-local address: FE80::C820:17FF:FE04:1C	Gi0/0	11	00:03:35	68	408	0	10
Version 11.0/2.0, Retrans: 0, Retries: 0, Prefixes: 2								
Topology-ids from peer - 0								
Stub Peer Advertising (CONNECTED SUMMARY) Routes								
Suppressing queries								
1	Link-local address: FE80::C823:17FF:FE0C:1C	Gi1/0	13	00:14:16	252	1512	0	7
Version 11.0/2.0, Retrans: 0, Retries: 0, Prefixes: 2								
Topology-ids from peer - 0								

Split-horizon

Split-horizon is a loop-prevention feature that prevents a router from advertising routes out the same interface they were learned on. As shown in Example 14-83, you can verify whether split-horizon is enabled or disabled by using the `show ipv6 eigrp interfaces detail` command.

**Key
Topic**

Example 14-83 Verifying EIGRP Stub Configuration of Neighbor Router

```
R1#show ipv6 eigrp interfaces detail
EIGRP-IPv6 Interfaces for AS(100)

      Xmit Queue   PeerQ      Mean   Pacing Time   Multicast   Pending
Interface Peers Un/Reliable Un/Reliable SRTT    Un/Reliable Flow Timer Routes
-----
Gi1/0      1      0/0        0/0      50       0/0         208         0

Hello-interval is 5, Hold-time is 15
Split-horizon is enabled
Next xmit serial <none>
Packetized sent/expedited: 8/0
Hello's sent/expedited: 708/3
Un/reliable mcasts: 0/6 Un/reliable ucasts: 11/5
Mcast exceptions: 0 CR packets: 0 ACKs suppressed: 0
Retransmissions sent: 1 Out-of-sequence rcvd: 0
Topology-ids on interface - 0
Authentication mode is md5, key-chain is "TEST"
```

As with EIGRP for IPv4, split-horizon is an issue in EIGRP for IPv6 network designs that need routes to be advertised out interfaces they were learned on: an NBMA Frame Relay hub-and-spoke topology or a DMVPN, which both use multipoint interfaces on the hub. Therefore, it needs to be disabled on the hub in these networks.

EIGRP for IPv6 Trouble Tickets

This section presents various trouble tickets relating to the topics discussed earlier in the chapter. The purpose of these trouble tickets is to give a process that you can follow when troubleshooting in the real world or in an exam environment. All trouble tickets in this section are based on the topology depicted in Figure 14-8.

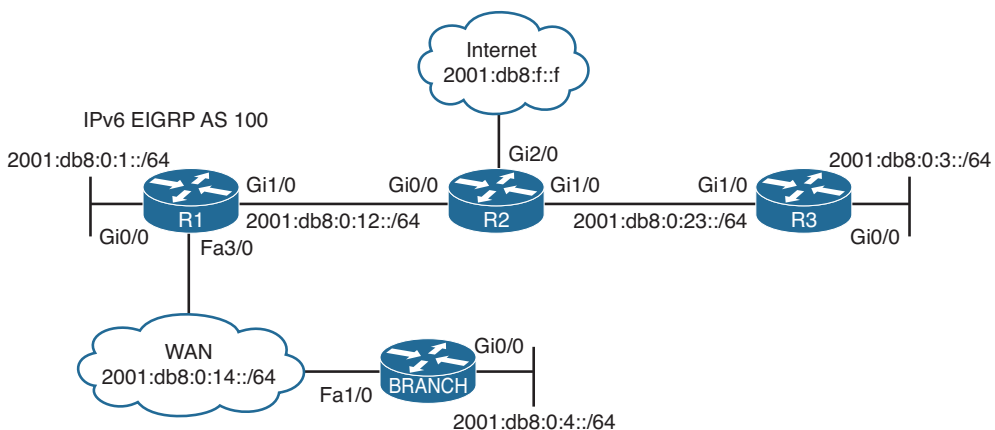


Figure 14-8 EIGRP for IPv6 Trouble Tickets Topology

Trouble Ticket 14-4

Problem: Users in the Branch network of 2001:db8:0:4::/64 have indicated that they are not able to access the Internet.

To verify the problem, you ping 2001:db8:f::f with a source address of 2001:db8:0:4::4, as shown in Example 14-84. The ping fails.

Example 14-84 Verifying the Issue Using an Extended IPv6 Ping

```
Branch#ping
Protocol [ip]: ipv6
Target IPv6 address: 2001:db8:f::f
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands? [no]: y
Source address or interface: 2001:db8:0:4::4
UDP protocol? [no]:
Verbose? [no]:
Precedence [0]:
DSCP [0]:
Include hop by hop option? [no]:
Include destination option? [no]:
Sweep range of sizes? [no]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:F::F, timeout is 2 seconds:
Packet sent with a source address of 2001:DB8:0:4::4
.....
Success rate is 0 percent (0/5)
```

Next you issue the **show ipv6 route 2001:db8:f::f** command on Branch to determine whether there is a route in the IPv6 routing table to reach the address. In Example 14-85, the route is not found.

Example 14-85 Verifying the Route to 2001:db8:f::f in the IPv6 Routing Table on Branch

```
Branch#show ipv6 route 2001:db8:f::f
% Route not found
```

Next you visit R1 to determine whether R1 has a route to reach 2001:db8:f::f by using the command **show ipv6 route 2001:db8:f::f**. In Example 14-86, you can see that the Internet address is reachable via a default route (::/0) that was learned via EIGRP.

Example 14-86 Verifying the Route to 2001:db8:f::f in the IPv6 Routing Table on R1

```
R1#show ipv6 route 2001:db8:f::f
Routing entry for ::/0
  Known via "eigrp 100", distance 170, metric 2816, type external
```

```

Route count is 1/1, share count 0
Routing paths:
  FE80::C821:17FF:FE04:8, GigabitEthernet1/0
    Last updated 00:08:28 ago

```

You conclude from this that Branch is not learning the default route from R1, which would be used to reach the Internet. You believe that it might be due to a neighbor relationship issue. Back on Branch, you issue the **show ipv6 eigrp neighbors** command, as shown in Example 14-87, and the output indicates that there is a neighbor relationship with a device out Fa1/0 that has the link-local address FE80::C820:17FF:FE04:54. You are pretty sure that is R1's link-local address on Fa3/0, but just to be sure, you issue the **show ipv6 interface brief** command on R1, as shown in Example 14-88. The link-local address from Example 14-87 matches the address in Example 14-88.

Example 14-87 Verifying EIGRP for IPv6 Neighbor Adjacencies

```

Branch#show ipv6 eigrp neighbors
EIGRP-IPv6 Neighbors for AS(100)

```

H	Address	Interface	Hold (sec)	Uptime	SRTT (ms)	RTO	Q Cnt	Seq Num
0	Link-local address: FE80::C820:17FF:FE04:54	Fa1/0	12	00:16:01	63	378	0	16

Example 14-88 Verifying an IPv6 Link-Local Address

```

R1#show ipv6 interface brief fastEthernet 3/0
FastEthernet3/0 [up/up]
  FE80::C820:17FF:FE04:54
  2001:DB8:0:14::1

```

You decide to check the EIGRP for IPv6 topology table on Branch to see whether it is learning any IPv6 routes from R1. As shown in Example 14-89, Branch is learning routes from R1. It has learned 2001:DB8:0:1::/64, and 2001:DB8:0:12::/64. You are quick to realize that those are only the connected routes on R1. You visit R1 again and issue the **show ipv6 eigrp topology** command and notice that R1 knows about other IPv6 routes, as shown in Example 14-90. However, it is not advertising them to Branch, as shown in Example 14-89.

Example 14-89 Verifying Learned IPv6 Routes on Branch

```

Branch#show ipv6 eigrp topology
EIGRP-IPv6 Topology Table for AS(100)/ID(4.4.4.4)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 2001:DB8:0:4::/64, 1 successors, FD is 2816
    via Connected, GigabitEthernet0/0
P 2001:DB8:0:1::/64, 1 successors, FD is 28416
    via FE80::C820:17FF:FE04:54 (28416/2816), FastEthernet1/0

```

```
P 2001:DB8:0:14::/64, 1 successors, FD is 28160
    via Connected, FastEthernet1/0
P 2001:DB8:0:12::/64, 1 successors, FD is 28416
    via FE80::C820:17FF:FE04:54 (28416/2816), FastEthernet1/0
```

Example 14-90 *Verifying Learned IPv6 Routes on R1*

```
R1#show ipv6 eigrp topology
EIGRP-IPv6 Topology Table for AS(100)/ID(10.1.12.1)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 2001:DB8:0:4::/64, 1 successors, FD is 28416
    via FE80::C828:DFE4:FEF4:1C (28416/2816), FastEthernet3/0
P 2001:DB8:0:1::/64, 1 successors, FD is 2816
    via Connected, GigabitEthernet0/0
P 2001:DB8:0:3::/64, 1 successors, FD is 3328
    via FE80::C821:17FF:FE04:8 (3328/3072), GigabitEthernet1/0
P ::/0, 1 successors, FD is 2816
    via FE80::C821:17FF:FE04:8 (2816/256), GigabitEthernet1/0
P 2001:DB8:0:14::/64, 1 successors, FD is 28160
    via Connected, FastEthernet3/0
P 2001:DB8:0:12::/64, 1 successors, FD is 2816
    via Connected, GigabitEthernet1/0
P 2001:DB8:0:23::/64, 1 successors, FD is 3072
    via FE80::C821:17FF:FE04:8 (3072/2816), GigabitEthernet1/0
```

You believe that a route filter is applied. Back on Branch, you issue the command **show run | section ipv6 router eigrp**, and as shown in Example 14-91, there is no distribute list (route filter) applied. You jump back to R1 and issue the same **show** command, as shown in Example 14-92, and there is no distribute list (route filter) applied either.

Example 14-91 *Verifying Route Filters on Branch*

```
Branch#show run | section ipv6 router eigrp
ipv6 router eigrp 100
eigrp router-id 4.4.4.4
```

Example 14-92 *Verifying Route Filters on R1*

```
R1#show run | section ipv6 router eigrp
ipv6 router eigrp 100
  passive-interface default
  no passive-interface GigabitEthernet1/0
  no passive-interface FastEthernet3/0
  eigrp stub connected summary
```

However, you notice in the output of Example 14-92 that R1 is configured as an EIGRP stub router that is advertising only connected and summary routes. This is the problem. The wrong router was configured as a stub router. The spoke (Branch) is supposed to be the stub router, not the hub (R1) in HQ. To solve this issue, you remove the stub configuration on R1 with the **no eigrp stub** command in IPv6 router EIGRP 100 configuration mode. You then issue the command **eigrp stub** on Branch in IPv6 router EIGRP 100 configuration mode.

To verify the problem is solved, you issue the **show ipv6 route 2001:db8:f::f** command on Branch to determine whether there is an entry in the routing table now. In Example 14-93, the output shows that the default route will be used.

Example 14-93 *Verifying the Route to 2001:db8:f::f in the IPv6 Routing Table on Branch*

```
Branch#show ipv6 route 2001:db8:f::f
Routing entry for ::/0
  Known via "eigrp 100", distance 170, metric 28416, type external
  Route count is 1/1, share count 0
  Routing paths:
    FE80::C820:17FF:FE04:54, FastEthernet1/0
    Last updated 00:03:09 ago
```

Next you issue the extended IPv6 ping, as shown in Example 14-94, and it is successful.

Example 14-94 *Verifying the Issue Is Solved Using an Extended IPv6 Ping*

```
Branch#ping
Protocol [ip]: ipv6
Target IPv6 address: 2001:db8:f::f
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands? [no]: y
Source address or interface: 2001:db8:0:4::4
UDP protocol? [no]:
Verbose? [no]:
Precedence [0]:
DSCP [0]:
Include hop by hop option? [no]:
Include destination option? [no]:
Sweep range of sizes? [no]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:F::F, timeout is 2 seconds:
Packet sent with a source address of 2001:DB8:0:4::4
!!!!!!
Success rate is 100 percent (5/5)
```


Troubleshooting Named EIGRP Configurations

The purpose of EIGRP named configurations is to provide you with a central location on the local router to perform all EIGRP for IPv4 and IPv6 configurations. Example 14-95 provides a sample named EIGRP configuration called TSHOOT_EIGRP. This named EIGRP configuration includes an IPv4 unicast address family and an IPv6 unicast address family. They are both using autonomous system 100; however, that is not mandatory.

Example 14-95 Sample Named EIGRP Configuration

```
Branch#show run | section router eigrp
router eigrp TSHOOT_EIGRP
!
address-family ipv4 unicast autonomous-system 100
!
af-interface default
  passive-interface
exit-af-interface
!
af-interface FastEthernet1/0
  no passive-interface
exit-af-interface
!
topology base
exit-af-topology
network 10.1.4.4 0.0.0.0
network 10.1.14.4 0.0.0.0
eigrp router-id 4.4.4.4
eigrp stub connected summary
exit-address-family
!
address-family ipv6 unicast autonomous-system 100
!
af-interface default
  passive-interface
exit-af-interface
!
af-interface FastEthernet1/0
  no passive-interface
exit-af-interface
!
topology base
  maximum-paths 2
  variance 3
exit-af-topology
eigrp router-id 44.44.44.44
eigrp stub connected summary
exit-address-family
```

Because the configuration is the only thing that is different, all the issues already discussed thus far for EIGRP for IPv4 and EIGRP for IPv6 will apply here. However, now you need to know which **show** commands will help you successfully troubleshoot named EIGRP deployments.

In this section, you learn the **show** commands that you can use to troubleshoot named EIGRP configurations.

Named EIGRP Verification Commands

With named EIGRP, you can use all the same EIGRP **show** commands that you used for classic EIGRP for IPv4 and classic EIGRP for IPv6 that were covered in this chapter. However, there is also a new set of **show** commands for named EIGRP that you may want to learn.

The command **show eigrp protocols** will display both the EIGRP for IPv4 address family and the EIGRP for IPv6 address family along with the autonomous system number associated with each. It also displays the K values, the router ID, whether the router is a stub router, the AD, the maximum paths, and the variance.

Key Topic

Example 14-96 *Output of show eigrp protocols*

```
Branch#show eigrp protocols
EIGRP-IPv4 VR(TSHOOT_EIGRP) Address-Family Protocol for AS(100)
Metric weight K1=1, K2=0, K3=1, K4=0, K5=0 K6=0
Metric rib-scale 128
Metric version 64bit
NSF-aware route hold timer is 240
Router-ID: 4.4.4.4
Stub, connected, summary
Topology : 0 (base)
  Active Timer: 3 min
  Distance: internal 90 external 170
  Maximum path: 4
  Maximum hopcount 100
  Maximum metric variance 1
  Total Prefix Count: 5
  Total Redist Count: 0

EIGRP-IPv6 VR(TSHOOT_EIGRP) Address-Family Protocol for AS(100)
Metric weight K1=1, K2=0, K3=1, K4=0, K5=0 K6=0
Metric rib-scale 128
Metric version 64bit
NSF-aware route hold timer is 240
Router-ID: 44.44.44.44
Stub, connected, summary
Topology : 0 (base)
  Active Timer: 3 min
```

```
Distance: internal 90 external 170
Maximum path: 2
Maximum hopcount 100
Maximum metric variance 3
Total Prefix Count: 7
Total Redist Count: 0
```

This is similar to the **show ip protocols** and **show ipv6 protocols** output. However, it is missing the interfaces that are participating in the routing process, along with the passive interfaces. Therefore, **show ip protocols** and **show ipv6 protocols** at this time are a preferred option.

To verify the interfaces that are participating in the routing process for each address family, you can issue the **show eigrp address-family ipv4 interfaces** command and the **show eigrp address-family ipv6 interfaces** command, as shown in Example 14-97. Make note that passive interfaces do not show up in this output. Based on the classic **show ip protocols** and **show ipv6 protocols** commands, we would be able to verify the passive interfaces.



Example 14-97 Verifying Interfaces Participating in the Named EIGRP Process

```
Branch#show eigrp address-family ipv4 interfaces
EIGRP-IPv4 VR(TSHOOT_EIGRP) Address-Family Interfaces for AS(100)

      Xmit Queue   PeerQ      Mean    Pacing Time   Multicast   Pending
Interface  Peers Un/Reliable Un/Reliable SRTT      Un/Reliable   Flow Timer   Routes
Fa1/0      1      0/0        0/0       88        0/0          50          0

Branch#show eigrp address-family ipv6 interfaces
EIGRP-IPv6 VR(TSHOOT_EIGRP) Address-Family Interfaces for AS(100)

      Xmit Queue   PeerQ      Mean    Pacing Time   Multicast   Pending
Interface  Peers Un/Reliable Un/Reliable SRTT      Un/Reliable   Flow Timer   Routes
Fa1/0      1      0/0        0/0       73        0/1         304          0
```



As shown in Example 14-98, when you add the **detail** keyword to the **show eigrp address-family ipv4 interfaces** command and the **show eigrp address-family ipv6 interfaces** command, you can verify additional interface parameters (for example, hello interval and hold time, whether split-horizon is enabled, whether authentication is set, and statistics about hellos and packets).

Example 14-98 Verifying Details of Interfaces Participating in the Named EIGRP Process

```
Branch#show eigrp address-family ipv4 interfaces detail
EIGRP-IPv4 VR(TSHOOT_EIGRP) Address-Family Interfaces for AS(100)

      Xmit Queue   PeerQ      Mean    Pacing Time   Multicast   Pending
Interface  Peers Un/Reliable Un/Reliable SRTT      Un/Reliable   Flow Timer   Routes
Fa1/0      1      0/0        0/0       88        0/0          50          0
```

```

Hello-interval is 5, Hold-time is 15
Split-horizon is enabled
Next xmit serial <none>
Packetized sent/expedited: 1/0
Hello's sent/expedited: 333/2
Un/reliable mcasts: 0/1  Un/reliable ucasts: 2/2
Mcast exceptions: 0  CR packets: 0  ACKs suppressed: 0
Retransmissions sent: 1  Out-of-sequence rcvd: 1
Topology-ids on interface - 0
Authentication mode is not set

Branch#show eigrp address-family ipv6 interfaces detail
EIGRP-IPv6 VR(TSHOOT_EIGRP) Address-Family Interfaces for AS(100)

```

Interface	Peers	Xmit Queue Un/Reliable	PeerQ Un/Reliable	Mean SRTT	Pacing Time Un/Reliable	Multicast Flow Timer	Pending Routes
Fa1/0	1	0/0	0/0	73	0/1	304	0

```

Hello-interval is 5, Hold-time is 15
Split-horizon is enabled
Next xmit serial <none>
Packetized sent/expedited: 3/0
Hello's sent/expedited: 595/3
Un/reliable mcasts: 0/2  Un/reliable ucasts: 5/3
Mcast exceptions: 0  CR packets: 0  ACKs suppressed: 0
Retransmissions sent: 1  Out-of-sequence rcvd: 2
Topology-ids on interface - 0
Authentication mode is not set

```

You can verify neighbors with the **show eigrp address-family ipv4 neighbors** and **show eigrp address-family ipv6 neighbors** commands, as shown in Example 14-99. Just like we saw with the classic commands, if you want to verify whether the neighbor is a stub router, you can add the **detail** keyword to the commands.

Example 14-99 Verifying Named EIGRP Neighbors

Key
Topic

```

Branch#show eigrp address-family ipv4 neighbors
EIGRP-IPv4 VR(TSHOOT_EIGRP) Address-Family Neighbors for AS(100)

```

H	Address	Interface	Hold Uptime (sec)	SRTT (ms)	RTO	Q	Seq Cnt Num
0	10.1.14.1	Fa1/0	14 00:31:08	88	528	0	8

```

Branch#show eigrp address-family ipv6 neighbors
EIGRP-IPv6 VR(TSHOOT_EIGRP) Address-Family Neighbors for AS(100)

```

H	Address	Interface	Hold Uptime (sec)	SRTT (ms)	RTO	Q	Seq Cnt Num
0	Link-local address: FE80::C820:17FF:FE04:54	Fa1/0	14 00:50:33	73	438	0	40

To display the topology table, you can use the commands **show eigrp address-family ipv4 topology** and **show eigrp address-family ipv6 topology**, as shown in Example 14-100

Example 14-100 *Verifying Named EIGRP Topology Tables*

```
Branch#show eigrp address-family ipv4 topology
EIGRP-IPv4 VR(TSHOOT_EIGRP) Topology Table for AS(100)/ID(4.4.4.4)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 10.1.12.0/24, 1 successors, FD is 13762560
    via 10.1.14.1 (13762560/1310720), FastEthernet1/0
P 10.1.14.0/24, 1 successors, FD is 13107200
    via Connected, FastEthernet1/0
P 10.1.3.0/24, 1 successors, FD is 15073280
    via 10.1.14.1 (15073280/2621440), FastEthernet1/0
P 10.1.23.0/24, 1 successors, FD is 14417920
    via 10.1.14.1 (14417920/1966080), FastEthernet1/0
P 10.1.4.0/24, 1 successors, FD is 1310720
    via Connected, GigabitEthernet0/0
P 10.1.1.0/24, 1 successors, FD is 13762560
    via 10.1.14.1 (13762560/1310720), FastEthernet1/0

Branch#show eigrp address-family ipv6 topology
EIGRP-IPv6 VR(TSHOOT_EIGRP) Topology Table for AS(100)/ID(44.44.44.44)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 2001:DB8:0:4::/64, 1 successors, FD is 1310720
    via Connected, GigabitEthernet0/0
P 2001:DB8:0:1::/64, 1 successors, FD is 13762560
    via FE80::C820:17FF:FE04:54 (13762560/1310720), FastEthernet1/0
P 2001:DB8:0:3::/64, 1 successors, FD is 15073280
    via FE80::C820:17FF:FE04:54 (15073280/2621440), FastEthernet1/0
P ::/0, 1 successors, FD is 13762560
    via FE80::C820:17FF:FE04:54 (13762560/1310720), FastEthernet1/0
P 2001:DB8:0:14::/64, 1 successors, FD is 13107200
    via Connected, FastEthernet1/0
P 2001:DB8:0:12::/64, 1 successors, FD is 13762560
    via FE80::C820:17FF:FE04:54 (13762560/1310720), FastEthernet1/0
P 2001:DB8:0:23::/64, 1 successors, FD is 14417920
    via FE80::C820:17FF:FE04:54 (14417920/1966080), FastEthernet1/0
```

Named EIGRP Trouble Tickets

This section presents various trouble tickets relating to the topics discussed earlier in the chapter. The purpose of these trouble tickets is to give a process that you can follow when troubleshooting in the real world or in an exam environment. All trouble tickets in this section are based on the topology depicted in Figure 14-9.

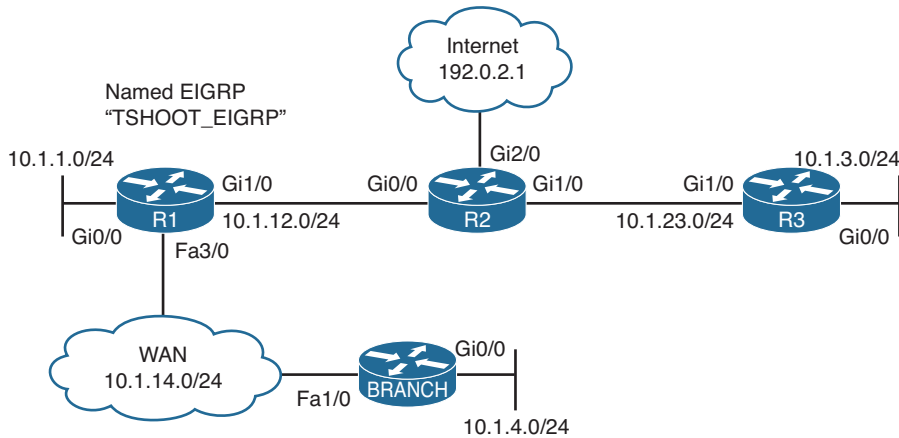


Figure 14-9 *Named EIGRP Trouble Tickets Topology*

Trouble Ticket 14-5

Problem: Users in the 10.1.4.0/24 network indicate that they are not able to access resources outside of their LAN.

On Branch, you verify the problem by pinging a few different IP addresses and source the packets from 10.1.4.4. As shown in Example 14-101, they all fail.

Example 14-101 *Verifying the Problem*

```
Branch#ping 10.1.3.3 source 10.1.4.4
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.3.3, timeout is 2 seconds:
Packet sent with a source address of 10.1.4.4
.....
Success rate is 0 percent (0/5)
Branch#ping 192.0.2.1 source 10.1.4.4
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.0.2.1, timeout is 2 seconds:
Packet sent with a source address of 10.1.4.4
.....
Success rate is 0 percent (0/5)
Branch#ping 10.1.1.1 source 10.1.4.4
Type escape sequence to abort.
```

```

Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:
Packet sent with a source address of 10.1.4.4
.....
Success rate is 0 percent (0/5)

```

Next you issue the **show ip route** command to verify whether any routes are installed in the routing table. As shown in Example 14-102, only local and directly connected routes are in the routing table.

Example 14-102 *Displaying the IPv4 Routing Table on Branch*

```

Branch#show ip route
...output omitted...
Gateway of last resort is not set

    10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C       10.1.4.0/24 is directly connected, GigabitEthernet0/0
L       10.1.4.4/32 is directly connected, GigabitEthernet0/0
C       10.1.14.0/24 is directly connected, FastEthernet1/0
L       10.1.14.4/32 is directly connected, FastEthernet1/0

```

You hypothesize that Branch is not a neighbor with R1 across the WAN. You issue the **show eigrp address-family ipv4 neighbors** command, as shown in Example 14-103, and confirm that R1 is not a neighbor.

Example 14-103 *Displaying the Named EIGRP IPv4 Neighbor Table*

```

Branch#show eigrp address-family ipv4 neighbors
EIGRP-IPv4 VR(TSHOOT_EIGRP) Address-Family Neighbors for AS(100)

```

Next you hypothesize that Fast Ethernet 1/0 (the interface that will form an adjacency with R1) is not participating in the named EIGRP process. You issue the command **show eigrp address-family ipv4 interfaces**, as shown in Example 14-104, and confirm your hypothesis.

Example 14-104 *Displaying the Named EIGRP IPv4 Interface Table*

```

Branch#show eigrp address-family ipv4 interfaces
EIGRP-IPv4 VR(TSHOOT_EIGRP) Address-Family Interfaces for AS(100)

```

	Xmit Queue	PeerQ	Mean	Pacing Time	Multicast	Pending	
Interface	Peers	Un/Reliable	Un/Reliable	SRTT	Un/Reliable	Flow Timer	Routes
Gi0/0	0	0/0	0/0	0	0/0	0	0

As shown in Example 14-105, the output of **show ip interface brief** indicates that Fast Ethernet 1/0 has an IPv4 address of 10.1.14.4. Therefore, a **network** statement is needed that will enable the EIGRP process on that interface.

Example 14-105 *Displaying the IPv4 Addresses of Interfaces*

```
Branch#show ip interface brief
```

Interface	IP-Address	OK?	Method	Status	Protocol
Ethernet0/0	unassigned	YES	unset	administratively down	down
GigabitEthernet0/0	10.1.4.4	YES	manual	up	up
FastEthernet1/0	10.1.14.4	YES	manual	up	up

Armed with the information you have, you issue the **show run | section router eigrp** command on Branch to confirm that the network statement is missing. In Example 14-106, you see that there is a valid **network** statement for 10.1.14.4. It is **network 10.1.14.4 0.0.0.0** and would successfully enable the EIGRP process on the interface. Therefore, your hypothesis was incorrect.

Example 14-106 *Reviewing Named EIGRP Configuration in the Running Configuration*

```
Branch#show running-config | section router eigrp
router eigrp TSHOOT_EIGRP
!
address-family ipv4 unicast autonomous-system 100
!
  af-interface default
    passive-interface
  exit-af-interface
!
  af-interface GigabitEthernet0/0
    no passive-interface
  exit-af-interface
!
  topology base
  exit-af-topology
  network 10.1.4.4 0.0.0.0
  network 10.1.14.4 0.0.0.0
  eigrp router-id 4.4.4.4
  eigrp stub connected summary
exit-address-family
!
address-family ipv6 unicast autonomous-system 100
!
  af-interface default
    passive-interface
  exit-af-interface
!
  af-interface FastEthernet1/0
    no passive-interface
  exit-af-interface
!
```



```

topology base
  maximum-paths 2
  variance 3
exit-af-topology
eigrp router-id 44.44.44.44
eigrp stub connected summary
exit-address-family

```

What could cause a neighbor relationship not to form? You list a few: authentication, passive interface, wrong subnet.

In Example 14-106, you notice that there are no authentication configurations. However, you do spot a passive interface command on Gig0/0. It is the **no passive-interface** command. You also notice that *af-interface default* has the **passive-interface** command and recall that all interfaces inherit configs under *af-interface default*. You also recall that they can be overridden with commands at the interface level. Reviewing the topology in Figure 14-9, you come to the conclusion that the wrong interface was configured with the **no passive-interface** command. It should have been Fast Ethernet 1/0 and not Gig0/0.

Example 14-107 presents the commands that you can use to fix this issue. Notice that once the issue is fixed, the neighbor relationship is formed with R1 at 10.1.14.1.

Example 14-107 *Modifying the Named EIGRP Configuration*

```

Branch#config t
Enter configuration commands, one per line. End with CNTL/Z.
Branch(config)#router eigrp TSHOOT_EIGRP
Branch(config-router)#address-family ipv4 unicast autonomous-system 100
Branch(config-router-af)#af-interface GigabitEthernet0/0
Branch(config-router-af-interface)#passive-interface
Branch(config-router-af-interface)#exit
Branch(config-router-af)#af-interface fastEthernet1/0
Branch(config-router-af-interface)#no passive-interface
%DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 10.1.14.1 (FastEthernet1/0) is up: new
adjacency
Branch(config-router-af-interface)#end
Branch#

```

You then review the IPv4 routing table as shown in Example 14-108 and notice all the EIGRP-learned routes.

Example 14-108 *Verifying the EIGRP-Learned Routes*

```

Branch#show ip route
...output omitted...
Gateway of last resort is 10.1.14.1 to network 0.0.0.0

D*EX 0.0.0.0/0 [170/112640] via 10.1.14.1, 00:00:34, FastEthernet1/0
    10.0.0.0/8 is variably subnetted, 8 subnets, 2 masks

```

```

D    10.1.1.0/24 [90/107520] via 10.1.14.1, 00:05:53, FastEthernet1/0
D    10.1.3.0/24 [90/117760] via 10.1.14.1, 00:05:53, FastEthernet1/0
C    10.1.4.0/24 is directly connected, GigabitEthernet0/0
L    10.1.4.4/32 is directly connected, GigabitEthernet0/0
D    10.1.12.0/24 [90/107520] via 10.1.14.1, 00:05:53, FastEthernet1/0
C    10.1.14.0/24 is directly connected, FastEthernet1/0
L    10.1.14.4/32 is directly connected, FastEthernet1/0
D    10.1.23.0/24 [90/112640] via 10.1.14.1, 00:05:53, FastEthernet1/0

```

Next you reissue the same pings that were used to confirm the problem. In Example 14-109, they are successful.

Example 14-109 *Successful Pings from Branch to Various Network IPs*

```

Branch#ping 10.1.1.1 source 10.1.4.4
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:
Packet sent with a source address of 10.1.4.4
!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 44/55/72 ms
Branch#ping 10.1.3.3 source 10.1.4.4
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.3.3, timeout is 2 seconds:
Packet sent with a source address of 10.1.4.4
!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 52/79/92 ms
Branch#ping 192.0.2.1 source 10.1.4.4
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.0.2.1, timeout is 2 seconds:
Packet sent with a source address of 10.1.4.4
!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 76/84/92 ms

```

Exam Preparation Tasks

As mentioned in the section “How to Use This Book” in the Introduction, you have a couple of choices for exam preparation: the exercises here; Chapter 22, “Final Preparation;” and the exam simulation questions on the CD-ROM.

Review All Key Topics

Review the most important topics in this chapter, noted with the Key Topic icon in the outer margin of the page. Table 14-2 lists a reference of these key topics and the page numbers on which each is found.



Table 14-2 *Key Topics for Chapter 14*

Key Topic Element	Description	Page Number
List	Identifies the possible reasons why an EIGRP neighbor relationship might not form	518
Example 14-2	Verifying the Autonomous System Number with show ip protocols	519
Example 14-4	Verifying EIGRP interfaces with show ip eigrp interfaces	521
Example 14-7	Verifying K values with show ip protocols	522
Example 14-8	Verifying passive interfaces with show ip protocols	523
Section	Authentication	525
List	Identifies the possible reasons why EIGRP for IPv4 routes may be missing from the routing table	528
Paragraph	Describes how a better source of routing information could cause suboptimal routing	533
List	Identifies what should be considered when troubleshooting route filters	534
Section	Stub configuration	535
Section	Split-horizon	537
List	Outlines what to keep in mind while troubleshooting route summarization	544
Example 14-33	Verifying variance and maximum paths	545
Example 14-74	Verifying EIGRP for IPv6 Neighbors	561
Example 14-76	Verifying EIGRP for IPv6 autonomous system numbers with show ipv6 protocols	562
Section	Interface not participating in routing process	563

Key Topic Element	Description	Page Number
Section	Stub configuration	565
Example 14-83	Verifying EIGRP stub configuration of neighbor router	567
Example 14-96	Output of show eigrp protocols	573
Example 14-97	Verifying interfaces participating in the named EIGRP process	574
Paragraph	Explains how to verify EIGRP for IPv4 and EIGRP for IPv6 timers, split-horizon, and authentication settings when using named EIGRP	574
Example 14-99	Verifying named EIGRP neighbors	575

Define Key Terms

Define the following key terms from this chapter and check your answers in the glossary:

hello packet, 224.0.0.10, **network** command, autonomous system number, K values, passive interface, key ID, key string, key chain, stub, split-horizon, successor, feasible successor, reported distance, feasible distance, discontinuous network, autosummari-
zation, classful, classless, maximum paths, variance, named EIGRP, address family

Command Reference to Check Your Memory

This section includes the most important **show** and **debug** commands covered in this chapter. It might not be necessary to memorize the complete syntax of every command, but you should be able to remember the basic keywords that are needed.

To test your memory of the commands, cover the right side of Table 14-3 with a piece of paper, read the description on the left side, and then see how much of the command you can remember.

The 300-135 TSHOOT exam focuses on practical, hands-on skills that are used by a networking professional. Therefore, you should be able to identify the commands needed to successfully identify and troubleshoot the issues presented in this chapter.

Table 14-3 *EIGRP show and debug Commands*

Task	Command Syntax
Displays the IPv4 routing protocols enabled on the router. For EIGRP, it displays autonomous system number, outgoing and incoming filters, K values, router ID, maximum paths, variance, local stub configuration, routing for networks, routing information sources, administrative distance, and passive interfaces.	show ip protocols
Shows a router's EIGRP neighbors.	show ip eigrp neighbors

Task	Command Syntax
Shows detailed information about a router's EIGRP neighbors, including whether the neighbor is a stub router, along with the types of networks it is advertising as a stub.	<code>show ip eigrp neighbors detail</code>
Displays all of a router's interfaces configured to participate in an EIGRP routing process (with the exception of passive interfaces).	<code>show ip eigrp interfaces</code>
Displays the interfaces participating in the EIGRP for IPv4 routing process, along with EIGRP hello and hold timers, whether the split-horizon rule is enabled, and whether authentication is being used.	<code>show ip eigrp interfaces detail</code>
Displays the EIGRP configuration in the running configuration.	<code>show run section router eigrp</code>
Displays the configuration of a specific interface in the running configuration. Valuable when trying to troubleshoot EIGRP interface commands.	<code>show run interface <i>interface_type interface_number</i></code>
Used to display the key chains and associated keys and key strings.	<code>show key chain</code>
Displays IPv4 interface parameters. For EIGRP, you can use it to verify whether the interface has joined the correct multicast group (224.0.0.10), and whether there are any ACLs applied to the interface that might be preventing an EIGRP adjacency from forming.	<code>show ip interface <i>interface_type interface_number</i></code>
Displays routes known to a router's EIGRP routing process. These routes are contained in the EIGRP topology table. The all-links keyword displays all routes learned for each network, and without the all-links keyword, only the successors and feasible successors are displayed for each network.	<code>show ip eigrp topology [all-links]</code>
Shows routes known to a router's IP routing table that were injected by the router's EIGRP routing process.	<code>show ip route eigrp</code>
Shows a router's EIGRP for IPv6 neighbors.	<code>show ipv6 eigrp neighbors</code>
Displays the IPv6 routing protocols enabled on the router. For EIGRP, it displays autonomous system number, outgoing and incoming filters, K values, router ID, maximum paths, variance, local stub configuration, interfaces participating in the routing process, routing information sources, administrative distance, and passive interfaces.	<code>show ipv6 protocols</code>
Displays all of a router's interfaces configured to participate in an EIGRP for IPv6 routing process (with the exception of passive interfaces).	<code>show ipv6 eigrp interfaces</code>

Task	Command Syntax
Displays the interfaces participating in the EIGRP for IPv6 routing process, along with EIGRP hello and hold timers, whether the split-horizon rule is enabled, and whether authentication is being used.	<code>show ipv6 eigrp interfaces detail</code>
Displays the IPv6 EIGRP configuration in the running configuration.	<code>show run section ipv6 router eigrp</code>
Shows detailed information about a router's EIGRP neighbors, including whether the neighbor is a stub router, along with the types of networks it is advertising as a stub.	<code>show ipv6 eigrp neighbors detail</code>
Shows routes known to a router's IP routing table that were injected by the router's EIGRP routing process.	<code>show ipv6 route eigrp</code>
Displays the EIGRP for IPv4 and IPv6 address families that are enabled on the router. It displays autonomous system number, K values, router ID, maximum paths, variance, local stub configuration, and administrative distance.	<code>show eigrp protocols</code>
Displays the interfaces that are participating in the named EIGRP for IPv4 address family.	<code>show eigrp address-family ipv4 interfaces</code>
Displays the interfaces that are participating in the named EIGRP for IPv6 address family.	<code>show eigrp address-family ipv6 interfaces</code>
Displays detailed information about the interfaces participating in the named EIGRP for IPv4 address family, including hello interval and hold time, whether split-horizon is enabled, whether authentication is set, and statistics about hellos and packets.	<code>show eigrp address-family ipv4 interfaces detail</code>
Displays detailed information about the interfaces participating in the named EIGRP for IPv6 address family, including hello interval and hold time, whether split-horizon is enabled, whether authentication is set, and statistics about hellos and packets.	<code>show eigrp address-family ipv6 interfaces detail</code>
Displays the EIGRP for IPv4 neighbor relationships that have formed.	<code>show eigrp address-family ipv4 neighbors</code>
Displays the EIGRP for IPV6 neighbor relationships that have formed.	<code>show eigrp address-family ipv6 neighbors</code>
Displays the EIGRP for IPv4 topology table for the address family.	<code>show eigrp address-family ipv4 topology</code>
Displays the EIGRP for IPv6 topology table for the address family.	<code>show eigrp address-family ipv6 topology</code>
Can be used to display all EIGRP packets exchanged with a router's EIGRP neighbors. However, the focus of the command can be narrowed to only display specific EIGRP packet types (for example, EIGRP hello packets).	<code>debug eigrp packets</code>



This chapter covers the following topics:

- **Troubleshooting OSPFv2:** This section covers the reasons why OSPFv2 neighbor relationships are not being formed and how you can identify them. In addition, you will learn the reasons why OSPFv2 routes might be missing and how to determine why they are missing.
- **OSPFv2 Trouble Tickets:** This section provides trouble tickets that demonstrate how you can use a structured troubleshooting process to solve a reported problem.
- **Troubleshooting OSPFv3 for IPv6:** In this section, you examine the different commands that you can use to troubleshoot OSPFv3 issues.
- **OSPFv3 Trouble Tickets:** This section provides trouble tickets that demonstrate how you can use a structured troubleshooting process to solve a reported problem.
- **Troubleshooting OSPFv3 Address Families:** In this section, you discover the commands that you can use to troubleshoot issues related to OSPFv3 address family configurations.
- **OSPFv3 Address Family Trouble Tickets:** This section provides trouble tickets that demonstrate how you can use a structured troubleshooting process to solve a reported problem.

Troubleshooting OSPF

The Open Shortest Path First (OSPF) dynamic routing protocol is a link-state routing protocol that uses Dijkstra's shortest path first (SPF) algorithm. It is an extremely scalable routing protocol because of its hierarchical design implementation. OSPF can route for both IPv4 and IPv6 protocols. This chapter focuses on troubleshooting both OSPFv2 and OSPFv3 using the classic configurations and the newer OSPF address family configurations.

Before any routes can be exchanged between OSPF routers on the same LAN or across a WAN, an OSPF neighbor relationship has to be formed. There are many reasons why a neighbor relationship will not form, and as a troubleshooter, you need to be aware of them. This chapter delves deeply into these reasons and gives you the tools needed to identify them and successfully solve neighbor issues.

Once neighbor relationships are formed, neighboring routers will exchange OSPF LSAs, which contain information about routes. In various cases, routes may end up missing, and you need to be able to determine why the routes are missing. This chapter discusses the various ways that OSPF routes could go missing, how you can identify the reasons why they are missing, and how you can solve route-related issues.

In this chapter, you will also learn how to troubleshoot issues related to load balancing, summarization, and discontinuous areas.

“Do I Know This Already?” Quiz

The “Do I Know This Already?” quiz allows you to assess whether you should read this entire chapter thoroughly or jump to the “Exam Preparation Tasks” section. If you are in doubt about your answers to these questions or your own assessment of your knowledge of the topics, read the entire chapter. Table 15-1 lists the major headings in this chapter and their corresponding “Do I Know This Already?” quiz questions. You can find the answers in Appendix A, “Answers to the ‘Do I Know This Already?’ Quizzes.”

Table 15-1 “Do I Know This Already?” Section-to-Question Mapping

Foundation Topics Section	Questions
Troubleshooting OSPFv2	1–6
Troubleshooting OSPFv3	7–8
Troubleshooting OSPFv3 Address Families	9–10

Caution The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you do not know the answer to a question or are only partially sure of the answer, you should mark that question as wrong for purposes of the self-assessment. Giving yourself credit for an answer that you correctly guess skews your self-assessment results and might provide you with a false sense of security.

1. Which three of the following are reasons why an OSPF neighbor relationship will not form?
 - a. Mismatched timers
 - b. Mismatched area numbers
 - c. Duplicate router IDs
 - d. Wrong designated router was elected
2. In which two OSPF states are you likely to find routers that have an MTU mismatch?
 - a. Init
 - b. 2Way
 - c. Exstart
 - d. Exchange
3. Which OSPFv2 command enables you to verify the hello interval and the dead interval?
 - a. `show ip protocols`
 - b. `show ip ospf interface`
 - c. `show ip ospf neighbor`
 - d. `show ip ospf database`
4. Which OSPFv2 `debug` command enables you to verify whether area numbers are mismatched?
 - a. `debug ip ospf hello`
 - b. `debug ip ospf adj`
 - c. `debug ip ospf packet`
 - d. `debug ip ospf events`
5. Which OSPF network type is the default on LAN interfaces?
 - a. Broadcast
 - b. NBMA
 - c. Point to point
 - d. Point to multipoint

6. Which LSA type describes routes outside the area but still within the OSPF routing domain (interarea routes)?
 - a. 1
 - b. 2
 - c. 3
 - d. 5
7. Which IPv6 OSPFv3 command enables you to verify whether an area is a stub, totally stubby, NSSA, or totally NSSA area?
 - a. `show ipv6 protocols`
 - b. `show ipv6 ospf`
 - c. `show ipv6 ospf interface`
 - d. `show ipv6 ospf neighbor`
8. Which IPv6 OSPFv3 command enables you to verify which routers the local router has formed neighbor adjacencies with?
 - a. `show ipv6 protocols`
 - b. `show ipv6 ospf`
 - c. `show ipv6 ospf interface`
 - d. `show ipv6 ospf neighbor`
9. Which two OSPFv3 address family commands are used to verify which OSPFv3 address family an interface is participating in?
 - a. `show ospfv3`
 - b. `show ospfv3 interface brief`
 - c. `show ospfv3 neighbors`
 - d. `show ospfv3 database`
10. Which OSPFv3 address family **debug** command will identify whether there is a mismatched stub area configuration?
 - a. `debug ospfv3 hello`
 - b. `debug ospfv3 packet`
 - c. `debug ospfv3 adj`
 - d. `debug ospfv3 events`

Foundation Topics

Troubleshooting OSPFv2

OSPF establishes neighbor relationships by sending hello packets out interfaces participating in the OSPF process. To enable the OSPF process on an interface and place it in an OSPF area, you use the **network *ip_address wildcard_mask area area_id*** command in router OSPF configuration mode or the **ip ospf *process_id area area_id*** command in interface configuration mode. For example, the following **network area** command enables OSPF on all interfaces with an IP address from 10.1.1.0 through 10.1.1.255 and places them in area 0: **network 10.1.1.0 0.0.0.255 area 0**. The following interface configuration command enables the OSPF process on the interface and places it in area 51: **ip ospf 1 area 51**. Because there are two different ways to enable OSPFv2 on an interface, you have to be very careful when troubleshooting neighbor adjacencies so that you are not led down the wrong path thinking the OSPF process was not enabled on an interface when in fact it was. This is your warning to check both places.

OSPF routers will receive LSAs from every router within the same area, meaning they learn about routes directly from the source within the same area. As a result, it is necessary that the LSAs are flooded through the area. This is mandatory because every router in an area must have the exact same LSDB for that area. This makes troubleshooting missing OSPF routes more difficult than distance vector routing protocols because it is harder to follow the path, especially in a multi-area OSPF domain.

This section focuses on the reasons why an OSPF neighbor relationship might not form and how we can identify them during the troubleshooting process. In addition, we will examine the reasons why OSPF routes might be missing, and how we can determine the reason why they are missing. To wrap up the section, we will troubleshoot OSPF issues that do not fall into the neighbor relationship or route categories.

Troubleshooting OSPFv2 Neighbor Adjacencies

To verify OSPFv2 neighbors, you use the **show ip ospf neighbor** command. In Example 15-1, you see a sample output of the **show ip ospf neighbor** command. It lists the neighbor ID, which is the router ID (RID) of the neighbor, the priority of the neighbor for the designated router / backup designated router (DR/BDR) election process, the state of the neighbor (covered shortly), and whether they are a DR, BDR, or DROther. In addition, it displays the dead time, which is how long the local router will wait until it declares the neighbor down if it does not hear another hello packet within that time (default is 40 seconds on a LAN). You can also see the neighbor's interface IP address that they sent the hello packet from and the local router interface that is used to reach that neighbor.



Example 15-1 *Verifying OSPF Neighbors with show ip ospf neighbor*

```
R1#show ip ospf neighbor
```

Neighbor	ID	Pri	State	Dead Time	Address	Interface
10.1.23.2		1	FULL/BDR	00:00:37	10.1.12.2	GigabitEthernet1/0

When an OSPF neighbor adjacency is successfully formed you will receive a syslog message similar to the following:

```
%OSPF-5-ADJCHG: Process 1, Nbr 10.1.23.2 on GigabitEthernet1/0 from LOADING to FULL, Loading Done
```

Here is a listing of reasons why an OSPFv2 neighbor relationship might not form:



- **Interface is down:** The interface has to be up/up.
- **Interface not running the OSPF process:** If the interface is not enabled for OSPF, it will not send hello packets or form an adjacency.
- **Mismatched timers:** Hello and dead timers have to match between neighbors.
- **Mismatched area numbers:** Both ends of a link must be in the same OSPF area.
- **Mismatched area type:** In addition to a normal OSPF area type, an area type could be either stub or not-so-stubby area (NSSA). The routers have to agree on the type of area they are in.
- **Different subnets:** Neighbors have to be in the same subnet.
- **Passive interface:** The passive interface feature suppresses the sending and receiving of hello packets while still allowing the interfaces network to be advertised.
- **Mismatched authentication information:** If one OSPF interface is configured for authentication, the OSPF interface at the other end of the link has to be configured with matching authentication information.
- **ACLs:** An ACL that is denying packets to the OSPF multicast address 224.0.0.5.
- **MTU mismatch:** The maximum transmission unit of neighboring interfaces must match.
- **Duplicate router IDs:** Router IDs must be unique.
- **Mismatched network types:** Based on the OSPF network type characteristics and default values, two neighbors configured with a different OSPF network type might not form an adjacency.

Adjacencies are not established upon the immediate receipt of hello messages. Rather, an adjacency transitions through multiple states, as described in Table 15-2.

**Table 15-2** *Adjacency States*

State	Description
Down	This state indicates that no hellos have been received from a neighbor.
Attempt	This state occurs after a router sends a unicast hello (as opposed to a multicast hello) to a configured neighbor and has not yet received a hello from that neighbor.
Init	This state occurs on a router that has received a hello message from its neighbor; however, the OSPF RID of the receiving router was not contained in the hello message. If a router remains in this state for a long period, something is probably preventing that router from correctly receiving hello packets from the neighboring router.
2Way	This state occurs when two OSPF routers have received hello messages from each other, and each router saw its own OSPF RID in the hello message it received. The 2Way state is an acceptable state to stay in between DROthers on an Ethernet LAN.
Exstart	This state occurs when the routers forming a full neighbor adjacency decide who will send their routing information first. This is accomplished using the RID. The router with the higher RID becomes the master and the other will become the slave. The master will send the routing information first. In a multiaccess network, the DR and BDR have to be determined first before this state starts. However, the DR does not have to be the master because each master/slave election is on a per-neighbor basis. If a router remains in this state for a long period, a maximum transmission unit (MTU) mismatch could exist between the neighboring routers, or a duplicate OSPF RID might exist.
Exchange	This state occurs when the two routers forming an adjacency send one another database descriptor (DBD) packets containing information about a router's link-state database. Each router compares the DBD packets received from the other router to identify missing entries in its own database. If a router remains in this state for a long period, an MTU mismatch could exist between the neighboring routers.
Loading	Based on the missing link-state database entries identified in the Exchange state, the Loading state occurs when each neighboring router requests the other router to send those missing entries. If a router remains in this state for a long period, a packet might have been corrupted, or a router might have a memory issue. Alternatively, it is possible that such a condition could result from the neighboring routers having an MTU mismatch.
Full	This state indicates that the neighboring OSPF routers have successfully exchanged their link-state information with one another, and an adjacency has been formed.

When an OSPF neighbor relationship does not form, the neighbor is not listed in the neighbor table. Therefore, you will need the assistance of an accurate network diagram and the **show cdp neighbors** command to verify who should be the neighbors.

When troubleshooting OSPF adjacencies, you need to be aware of how to verify the parameters associated with each reason we listed earlier. Let's look at them individually.

Interface Is Down

The interface has to be up if you plan on forming an OSPF neighbor adjacency. As we have seen already, we can verify the status of an interface with the **show ip interface brief** command.

Interface Not Running the OSPF Process

If the router OSPF configuration mode **network ip_address wildcard_mask area area_id** command or the **ip ospf process_id area area_id** interface command is misconfigured, OSPF may not be enabled on the proper interfaces. As a result, hello packets will not be sent and neighbor relationships will not be formed. You also have to specify the OSPF area the interface belongs to. Therefore, if the command is correct, except for the area ID, the interface is participating in the OSPF process but in the wrong area. This will prevent a neighbor relationship from forming as well. You can verify which interfaces are participating in the OSPF process with the command **show ip ospf interface brief**, as shown in Example 15-2. In this example, two interfaces are participating in OSPF process 1. They are both in area 1 and are the designated router interfaces for the multiaccess networks. You can also verify the IP address and masks of the interfaces along with the number of full neighbor relationships that have been formed out the interface versus the total number of neighbors out the interface.

Note Remember that OSPF passive interfaces do show up in this output.



Example 15-2 Verifying OSPF Interfaces with **show ip ospf interface brief**

```
R1#show ip ospf interface brief
```

Interface	PID	Area	IP Address/Mask	Cost	State	Nbrs	F/C
Gi0/0	1	1	10.1.1.1/24	1	DR	0/0	
Gi1/0	1	1	10.1.12.1/24	1	DR	1/1	

The output of **show ip protocols** displays the **network ip_address wildcard_mask area area_id** statements as well as those interfaces that were enabled for OSPF with the **ip ospf process_id area area_id** interface command. Focus on the highlighted text in Example 15-3. Notice that it states *Routing for Networks*. Those are *not* the networks we are routing for. We are routing for the networks associated with the interfaces OSPF will be enabled on, based on the **network area** statement. In this case, **10.1.1.1 0.0.0.0 area 1** really means **network 10.1.1.1 0.0.0.0 area 1**. Therefore, the interface with this IP address will be enabled for the OSPF process and placed in area 1. In addition, you can see which interfaces were explicitly configured to participate in the OSPF process with the **ip ospf process_id area area_id** interface configuration mode command. In this

example, it is Gigabit Ethernet 1/0 that was enabled for OSPF with the **ip ospf 1 area 1** command, and Gigabit Ethernet 0/0 was enabled for OSPF with the **network 10.1.1 0.0.0.0 area 1** router OSPF configuration mode command.

Example 15-3 Verifying OSPF-Enabled Interfaces with show ip protocols

```
R1#show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "ospf 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 10.1.12.1
  Number of areas in this router is 1. 1 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    10.1.1.1 0.0.0.0 area 1
  Routing on Interfaces Configured Explicitly (Area 1):
    GigabitEthernet1/0
  Routing Information Sources:
    Gateway         Distance      Last Update
    10.1.23.2        110          00:24:22
  Distance: (default is 110)
```

As you can see, the **network area** statement is extremely important, as is the **ip ospf area** command. If either are misconfigured, interfaces that should be participating in the OSPF process might not be, and interfaces that should not be participating in the OSPF process might be. In addition, it is possible that they might be participating but in the wrong area, causing neighbor relationships not to form. Therefore, you should be able to recognize issues related with both these commands.

Note If an interface is enabled for OSPF with both the **network area** command and the **ip ospf area** command, the **ip ospf area** command takes precedence.

Mismatched Timers

Unlike Enhanced Interior Gateway Routing Protocol (EIGRP), OSPF timers do have to match between neighbors to form a neighbor adjacency. The hello timer defaults to 10 seconds for broadcast and point-to-point network types and 30 seconds for nonbroadcast and point-to-multipoint network types. The dead timer defaults to 40 seconds for broadcast and point-to-point network types and 120 seconds for nonbroadcast and point-to-multipoint network types. To verify the current timers associated with an OSPF interface, issue the **show ip ospf interface *interface_type interface_number*** command,

as shown in Example 15-4. In this example, Gigabit Ethernet 1/0 is using the default timers of 10 and 40. When determining whether timers match, use the spot-the-difference method between the outputs on both routers.

Key Topic

Example 15-4 Displaying OSPF Interface Timers on R1 Gigabit Ethernet 1/0

```
R1#show ip ospf interface gigabitEthernet 1/0
GigabitEthernet1/0 is up, line protocol is up
  Internet Address 10.1.12.1/24, Area 1, Attached via Interface Enable
  Process ID 1, Router ID 10.1.12.1, Network Type BROADCAST, Cost: 1
  Topology-MTID      Cost      Disabled      Shutdown      Topology Name
    0                1         no           no           Base
  Enabled by interface config, including secondary ip addresses
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 10.1.12.1, Interface address 10.1.12.1
  Backup Designated router (ID) 10.1.23.2, Interface address 10.1.12.2
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    oob-resync timeout 40
    Hello due in 00:00:04
  Supports Link-local Signaling (LLS)
  Cisco NSF helper support enabled
  IETF NSF helper support enabled
  Index 1/1, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 1
  Last flood scan time is 0 msec, maximum is 4 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 10.1.23.2 (Backup Designated Router)
  Suppress hello for 0 neighbor(s)
```

Using the **debug ip ospf hello** command when troubleshooting adjacencies will reveal mismatched timers, as shown in Example 15-5. In this example, the packet received (R) has a dead of 44 and a hello of 11. The local device (C) has a dead of 40 and a hello of 10.

Example 15-5 Using debug ip ospf hello to Identify Mismatched Timers

```
R1#debug ip ospf hello
OSPF hello debugging is on
R1#
OSPF-1 HELLO Gi1/0: Rcv hello from 2.2.2.2 area 1 10.1.12.2
OSPF-1 HELLO Gi1/0: Mismatched hello parameters from 10.1.12.2
OSPF-1 HELLO Gi1/0: Dead R 44 C 40, Hello R 11 C 10 Mask R 255.255.255.0 C
255.255.255.0
R1#
```


Mismatched Area Numbers



OSPF uses the concept of areas to make it an extremely scalable dynamic routing protocol. For OSPF routers to form a neighbor adjacency, their neighboring interfaces must be in the same area. You can verify the area an OSPF interface is part of using the **show ip ospf interface interface_type interface_number** command, as shown in Example 15-6, or the **show ip ospf interface brief** command, as shown in Example 15-7. When determining whether area IDs match, use the spot-the-difference method between the outputs on both routers.

Example 15-6 Displaying OSPF Interface Area Using the **show ip ospf interface interface_type interface_number** Command

```
R1#show ip ospf interface gigabitEthernet 1/0
GigabitEthernet1/0 is up, line protocol is up
  Internet Address 10.1.12.1/24, Area 1, Attached via Interface Enable
  Process ID 1, Router ID 10.1.12.1, Network Type BROADCAST, Cost: 1
  Topology-MTID      Cost      Disabled      Shutdown      Topology Name
    0                1          no            no            Base
  Enabled by interface config, including secondary ip addresses
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 10.1.12.1, Interface address 10.1.12.1
  Backup Designated router (ID) 10.1.23.2, Interface address 10.1.12.2
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    oob-resync timeout 40
    Hello due in 00:00:04
  Supports Link-local Signaling (LLS)
  Cisco NSF helper support enabled
  IETF NSF helper support enabled
  Index 1/1, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 1
  Last flood scan time is 0 msec, maximum is 4 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 10.1.23.2 (Backup Designated Router)
  Suppress hello for 0 neighbor(s)
```

Example 15-7 Displaying OSPF Interface Area Using the **show ip ospf interface brief** Command

```
R1#show ip ospf interface brief
Interface  PID  Area  IP Address/Mask  Cost  State Nbrs F/C
Gi1/0     1    1     10.1.12.1/24    1     DR   1/1
```

Using the **debug ip ospf adj** command when troubleshooting adjacencies will reveal mismatched area numbers, as shown in Example 15-8. In this example, the packet received has an area ID of 1 and the local interface is participating in area 2.

Example 15-8 *Using debug ip ospf adj to Identify Mismatched Area Numbers*

```

R1#debug ip ospf adj
OSPF adjacency debugging is on
R1#
OSPF-1 ADJ   Gi1/0: Rcv pkt from 10.1.12.2, area 0.0.0.2, mismatched area 0.0.0.1 in
the header
R1#u all
All possible debugging has been turned off

```

Mismatched Area Type

The default OSPF area type is classified as a normal area. However, you can convert a normal area into a stub area or NSSA area to control the types of LSAs that will be sent into the area from an Area Border Router (ABR). For routers within an area to form adjacencies, they must agree on the area type. Within the hello packet, there is a stub area flag that is designed to indicate the type of area the neighbor is in. You can verify the types of areas connected to the router with the **show ip protocols** command. However, it does not tell you which area is which type. In Example 15-9, which displays the output of **show ip protocols**, there is only one area (area 1); therefore, you can deduce that it is the stub area. However, if there is a router with multiple areas connected to it, you will verify the areas and their type using the **show ip ospf** command, as shown in Example 15-9. In this example, any interface in area 1 is in a stub area.

**Key
Topic****Example 15-9** *Determining the Type of OSPF Areas*

```

R1#show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "ospf 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 10.1.12.1
  Number of areas in this router is 1. 0 normal 1 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    10.1.1.1 0.0.0.0 area 1
  Routing on Interfaces Configured Explicitly (Area 1):
    GigabitEthernet1/0
  Routing Information Sources:
    Gateway         Distance      Last Update
    10.1.23.2         110          00:04:42
  Distance: (default is 110)

R1#show ip ospf
Routing Process "ospf 1" with ID 10.1.12.1
Start time: 02:23:19.824, Time elapsed: 02:08:52.184

```

```

...output omitted...
Reference bandwidth unit is 100 mbps
  Area 1
    Number of interfaces in this area is 2
    It is a stub area
    Area has no authentication
    SPF algorithm last executed 00:05:46.800 ago
...output omitted...

```

Using the **debug ip ospf hello** command when troubleshooting adjacencies will reveal mismatched area types, as shown in Example 15-10. In this example, it states that the packet received has a mismatched Stub/Transit area option bit.

Example 15-10 *Using debug ip ospf hello to Identify Mismatched Area Types*

```

R1#debug ip ospf hello
OSPF hello debugging is on
R1#
OSPF-1 HELLO Gi1/0: Rcv hello from 2.2.2.2 area 1 10.1.12.2
OSPF-1 HELLO Gi1/0: Hello from 10.1.12.2 with mismatched Stub/Transit area option
bit
R1#

```

Different Subnets

To form an OSPF neighbor adjacency, the router interfaces must be on the same subnet. You can verify this in many ways. The simplest is to look at the interface configuration in the running configuration with the **show run interface interface_type interface_number** command. Example 15-11 displays the configuration of Gig1/0 on R1 and Gig0/0 on R2. Are they in the same subnet? Yes! Based on the IP address and the subnet mask, they would both be in the 10.1.12.0/24 subnet.

Example 15-11 *Verifying Neighboring Interfaces Are on the Same Subnet*

```

R1#show running-config interface gigabitEthernet 1/0
Building configuration...

Current configuration : 108 bytes
!
interface GigabitEthernet1/0
 ip address 10.1.12.1 255.255.255.0
 ip ospf 1 area 1
 negotiation auto
end

R2#show running-config interface gigabitEthernet 0/0

```

```
Building configuration...

Current configuration : 132 bytes
!
interface GigabitEthernet0/0
 ip address 10.1.12.2 255.255.255.0
 negotiation auto
end
```

Passive Interface

The passive interface feature is a must have for all organizations. It does two things: 1) reduces the OSPF related traffic on a network; 2) improves OSPF security.



The passive interface feature turns off the sending and receiving of OSPF packets on an interface while still allowing the interfaces network ID to be injected into the OSPF process and advertised to other OSPF neighbors. This ensures that rogue routers that attach to the network will not be able to form an adjacency with your legitimate router on that interface since it is not sending or receiving OSPF packets on the interface. However, if you configure the wrong interface as passive, a legitimate OSPF neighbor relationship will not be formed. As shown in the **show ip protocols** output of Example 15-12, Gigabit Ethernet 0/0 is a passive interface. If there are no passive interfaces, this section will not appear in the output of **show ip protocols**.

Example 15-12 Verifying Passive Interfaces with show ip protocols

```
R1#show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "ospf 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 10.1.12.1
  Number of areas in this router is 1. 0 normal 1 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    10.1.1.1 0.0.0.0 area 1
  Routing on Interfaces Configured Explicitly (Area 1):
    GigabitEthernet1/0
  Passive Interface(s):
    GigabitEthernet0/0
  Routing Information Sources:
    Gateway         Distance      Last Update
    10.1.23.2        110          00:00:03
  Distance: (default is 110)
```

Mismatched Authentication Information

Authentication is used to ensure that your OSPF routers only form neighbor relationships with legitimate routers and they only accept OSPF packets from legitimate routers. Therefore, if authentication is implemented, both routers must agree on the settings for a neighbor relationship to form. With authentication, you can use the spot-the-difference method when troubleshooting. OSPF supports three types of authentication:

- **Null:** Known as type 0 and means no authentication
- **Plain text:** Known as type 1 and sends credentials in clear text
- **MD5:** Known as type 2 and sends a hash

OSPF authentication can be enabled on an interface-by-interface basis or for all interfaces in the area at the same time. Knowing which commands to use to verify these different authentication configuration options is important. To verify whether authentication has been enabled for the entire area on the router, you use the **show ip ospf** command, as shown in Example 15-13. However, with message digest 5 (MD5) authentication, you still have to verify the key ID that is being used on an interface-by-interface basis by using the **show ip ospf interface interface_type interface_number** command, as shown in Example 15-14. In addition, you must verify the case sensitive key string that is being used by using the **show run interface interface_type interface_number** command.



Example 15-13 Verifying OSPF Area Authentication

```
R1#show ip ospf
Routing Process "ospf 1" with ID 10.1.12.1
  Start time: 02:23:19.824, Time elapsed: 02:46:34.488
...output omitted...
Reference bandwidth unit is 100 mbps
  Area 1
    Number of interfaces in this area is 2
    It is a stub area
    Area has message digest authentication
    SPF algorithm last executed 00:25:12.220 ago
...output omitted...
```



Example 15-14 Verifying OSPF Authentication Key

```
R1#show ip ospf interface gigabitEthernet 1/0
GigabitEthernet1/0 is up, line protocol is up
  Internet Address 10.1.12.1/24, Area 1, Attached via Interface Enable
...output omitted...
Neighbor Count is 0, Adjacent neighbor count is 0
Suppress hello for 0 neighbor(s)
Message digest authentication enabled
  Youngest key id is 1
```

Note If you configure authentication on an interface-by-interface basis, the output of `show ip ospf` will state *Area has no authentication*. Therefore, you need to make sure you check the output of `show ip ospf interface` as well.

Using the `debug ip ospf adj` command when troubleshooting adjacencies will reveal mismatched authentication information, as shown in Example 15-15. In this example, the packet received is using null authentication (type 0), and the local router is using plain text authentication (type 1).

Example 15-15 *Using debug ip ospf adj to Identify Mismatched Authentication Information*

```
R1#debug ip ospf adj
OSPF adjacency debugging is on
R1#
OSPF-1 ADJ   Gil/0: Rcv pkt from 10.1.12.2 : Mismatched Authentication type. Input
packet specified type 0, we use type 1
R1#
```

ACLs

Access control lists (ACLs) are extremely powerful. Depending on how they are implemented will determine what they are controlling in your network. If an ACL is applied to an interface and the ACL is not permitting OSPF packets, a neighbor relationship will not form. To determine whether an ACL is applied to an interface, use the `show ip interface interface_type interface_number` command, as shown in Example 15-16. Notice that ACL 100 is applied inbound on interface Gig1/0. To verify the ACL 100 entries, issue the command `show access-list 100`, as shown in Example 15-17. In this case, you can see that ACL 100 is denying OSPF traffic, which would prevent a neighbor relationship from forming.

Example 15-16 *Verifying ACLs Applied to Interfaces*

```
R1#show ip interface gig 1/0
GigabitEthernet1/0 is up, line protocol is up
  Internet address is 10.1.12.1/24
  Broadcast address is 255.255.255.255
  Address determined by setup command
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Multicast reserved groups joined: 224.0.0.10
  Outgoing access list is not set
  Inbound access list is 100
  Proxy ARP is enabled
  Local Proxy ARP is disabled
  Security level is default
  Split horizon is enabled
```

Example 15-17 *Verifying ACLs Entries*

```
R1#show access-lists 100
Extended IP access list 100
 10 deny ospf any any (62 matches)
 20 permit ip any any
```

MTU Mismatch

For OSPF routers to become neighbors and achieve the full adjacency state, each router's interface forming the adjacency must have the exact same MTU. If not, the routers will see each other but get stuck in the exstart/exchange states. In Example 15-18 the output of **show ip ospf neighbor** indicates that R1 is stuck in the exchange state and that R2 is stuck in the exstart state.

Example 15-18 *Symptoms of an MTU Mismatch (Stuck in Exstart/Exchange)*

```
R1#show ip ospf neighbor

Neighbor ID Pri   State           Dead Time   Address      Interface
10.1.23.2    1    EXCHANGE/DR     00:00:38   10.1.12.2   GigabitEthernet1/0

R2#show ip ospf neighbor

Neighbor ID Pri   State           Dead Time   Address      Interface
10.1.12.1    1    EXSTART/BDR     00:00:37   10.1.12.1   GigabitEthernet0/0
```

In the output of **show ip ospf interface brief**, you will see the Nbrs F/C column without expected values. In Example 15-19, you see 0/1 in the Nbrs F/C column, which indicates that there is one neighbor out the interface but that there are zero full adjacencies.

Example 15-19 *Symptoms of an MTU Mismatch (Nbrs Column Values Do Not Match)*

```
R1#show ip ospf interface brief

Interface  PID  Area  IP Address/Mask  Cost  State  Nbrs F/C
Gi1/0     1    1     10.1.12.1/24    1     BDR    0/1
Gi0/0     1    1     10.1.1.1/24     1     DR     0/0
```

To verify the MTU configured on an interface, issue the **show run interface interface_type interface_number** command. As shown in Example 15-20, the MTU of Gigabit Ethernet 1/0 on R1 is 1476, and because nothing is listed in the Gigabit Ethernet 0/0 configuration of R2, it is using the default value of 1500.

Example 15-20 *Verifying the MTU of an Interface*

```

R1#show run interface gigabitEthernet 1/0
Building configuration...

Current configuration : 195 bytes
!
interface GigabitEthernet1/0
 ip address 10.1.12.1 255.255.255.0
 ip mtu 1476
 ip ospf authentication-key CISCO
 ip ospf message-digest-key 1 md5 CISCO
 ip ospf 1 area 1
 negotiation auto
end

R2#show run interface gigabitEthernet 0/0
Building configuration...

Current configuration : 211 bytes
!
interface GigabitEthernet0/0
 ip address 10.1.12.2 255.255.255.0
 ip ospf authentication message-digest
 ip ospf message-digest-key 1 md5 CISCO
 negotiation auto
end

```

To solve this issue, you can either manually modify the MTU values of the interfaces so that they match, or you can use the **ip ospf mtu-ignore** interface configuration command, which will stop OSPF from comparing the MTU when trying to form an adjacency.

Duplicate Router IDs

RIDs must be unique for many reasons. One of the reasons is that a neighbor relationship will not form between two routers if they have the same RID. When a duplicate RID exists, you will receive a syslog message similar to the following:

```
%OSPF-4-DUP_RTRID_NBR: OSPF detected duplicate router-id 10.1.23.2 from
10.1.12.2 on interface GigabitEthernet1/0
```

To verify the RID of an OSPF router use the **show ip protocols** command as shown in Example 15-21. However, almost all OSPF **show** commands display the RID in their output so you can verify it anyway you like. In this case, the RID of R1 is 10.1.23.2, as shown in the output of **show ip protocols**. If you manually change the RID with the **router-id** command in router OSPF configuration mode you must reset the OSPF process with the **clear ip ospf process** command before it takes affect.

Example 15-21 *Verifying OSPF RID*

```

R1#show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "ospf 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 10.1.23.2
  Number of areas in this router is 1. 0 normal 1 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    10.1.1.1 0.0.0.0 area 1
  Routing on Interfaces Configured Explicitly (Area 1):
    GigabitEthernet1/0
  Passive Interface(s):
    Ethernet0/0
    GigabitEthernet0/0
  Routing Information Sources:
    Gateway         Distance      Last Update
    10.1.23.2         110          00:05:31
  Distance: (default is 110)

```

Mismatched Network Types

OSPF supports multiple network types. Different network types have different default values. Therefore, if two OSPF routers that are trying to form a neighbor adjacency are configured with noncompatible network types, a neighbor relationship will not form.

Table 15-3 shows a listing of the OSPF network types and their characteristics.

Table 15-3 *OSPF Network Types and Characteristics*

Type	Default	Neighbors	DR/BDR	Timers
Broadcast	Default on LAN interfaces	Discovered automatically	DR and BDR elected automatically	Hello 10 Dead 40
NBMA (Nonbroadcast)	Default on Frame Relay main and point-to-multipoint interfaces	Statically configured	DR must be manually configured on the hub router	Hello 30 Dead 120
Point-to-Point	Default on point to point serial and point-to-point Frame Relay subinterfaces	Discovered automatically	No DR or BDR	Hello 10 Dead 40



Type	Default	Neighbors	DR/BDR	Timers
Point-to-Multipoint	(Not a default) Optimal for hub-and-spoke topologies (Frame-Relay)	Discovered automatically	No DR or BDR	Hello 30 Dead 120
Point-to-Multipoint Nonbroadcast	(Not a default) Optimal for hub-and-spoke topologies (Frame Relay) that do not support broadcast or multicast traffic	Statically Configured	No DR or BDR	Hello 30 Dead 120

To determine the network type associated with an OSPF-enabled interface, issue the command **show ip ospf interface *interface_type* *interface_number***. In Example 15-22, R1's interface Gig1/0 is using the OSPF network type Broadcast. Use the spot-the-difference troubleshooting method when determining whether the network types do not match.

Example 15-22 Verifying OSPF Network Type

```
R1#show ip ospf interface gigabitEthernet 1/0
GigabitEthernet1/0 is up, line protocol is up
  Internet Address 10.1.12.1/24, Area 1, Attached via Interface Enable
  Process ID 1, Router ID 10.1.12.1, Network Type BROADCAST, Cost: 1
  Topology-MTID      Cost      Disabled      Shutdown      Topology Name
    0                1         no           no           Base
  Enabled by interface config, including secondary ip addresses
  Transmit Delay is 1 sec, State BDR, Priority 1
  Designated Router (ID) 10.1.23.2, Interface address 10.1.12.2
  Backup Designated router (ID) 10.1.12.1, Interface address 10.1.12.1
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    oob-resync timeout 40
    Hello due in 00:00:07
  Supports Link-local Signaling (LLS)
  Cisco NSF helper support enabled
  IETF NSF helper support enabled
  Index 1/1, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 1
  Last flood scan time is 4 msec, maximum is 4 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 10.1.23.2 (Designated Router)
  Suppress hello for 0 neighbor(s)
  Message digest authentication enabled
  Youngest key id is 1
```

Troubleshooting OSPFv2 Routes

As discussed already, neighbor relationships are the foundation for OSPF information sharing. If we have no neighbors, we will not learn any routes. So, besides the lack of a neighbor, what would be reasons for missing routes in an OSPF network?

Following is a listing of some common reasons as to why OSPF routes might be missing either in the LSDB or the routing table:



- **Interface not running the OSPF process:** If the interface is not participating in the OSPF process, the network the interface is part of will not be injected into the OSPF process and therefore will not be advertised to neighbors.
- **Better source of information:** If the exact same network is learned from a more reliable source, it is used instead of the OSPF-learned information.
- **Route filtering:** A filter might be set up that is preventing a route from being installed in the routing table.
- **Stub area configuration:** If the wrong type of stub area is chosen, you might be receiving a default route instead of the actual route.
- **Interface is shut down:** The OSPF-enabled interface must be up/up for the network associated with the interface to be advertised.
- **Wrong designated router was elected:** In a hub-and-spoke environment, if the wrong router is the DR, routes will not be exchanged properly.
- **Duplicate RIDs:** If there are two or more routers with the same RID, routes will be missing in the topology.

Let's take a look at each of these individually and identify how we can recognize them during the troubleshooting process.

Interface Not Running the OSPF Process

As discussed earlier, when you use the **network area** command or the **ip ospf area** interface command, the OSPF process is enabled on interfaces. OSPF then takes the network/subnet the interface is part of and injects it into the link-state database (LSDB) so that it can be advertised to other routers in the autonomous system. Therefore, even interfaces that will not form neighbor relationships with other routers need to be participating in the OSPF process for the interfaces network ID to be advertised.

As discussed in an earlier section, the output of **show ip protocols** displays the **network area** statements in addition to the interfaces that were explicitly configured with the **ip ospf area** interface command. Focus on the highlighted text in Example 15-23. Notice that it states *Routing for Networks*. Those are *not* the networks we are routing for. We are routing for the networks associated with the interface OSPF will be enabled on, based on the network statement. So, **10.1.1.1 0.0.0.0 area 1** means to enable OSPF on the interface with the IP address 10.1.1.1 and place it in area 1. We will then route for the network associated with that interface. Also, you can see that Gig1/0 was explicitly configured to

participate in the OSPF process; therefore, OSPF will route for the network associated with that interface as well.

Example 15-23 Verifying OSPF-Enabled Interfaces with show ip protocols

```
R1#show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "ospf 1"

  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 10.1.12.1
  Number of areas in this router is 1. 1 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    10.1.1.1 0.0.0.0 area 1
  Routing on Interfaces Configured Explicitly (Area 1):
    GigabitEthernet1/0
  Passive Interface(s):
    Ethernet0/0
    GigabitEthernet0/0
  Routing Information Sources:
    Gateway         Distance      Last Update
    10.1.23.2        110          01:00:43
    10.1.23.3        110          01:00:43
  Distance: (default is 110)
```

So, what networks are we actually routing for then? The networks associated with the interfaces that are now enabled for OSPF. In Example 15-24, you can see the output of the **show ip interface** command on R1 for Gig0/0 and Gig1/0, which was piped to only include the **Internet** address. Notice that they are in a /24 network. As a result, the network IDs would be 10.1.1.0/24 and 10.1.12.0/24. *Those are the networks we are routing for.*

Example 15-24 Verifying Network IDs with show ip interface

```
R1#show ip interface gi0/0 | i Internet
  Internet address is 10.1.1.1/24
R1#show ip interface gi1/0 | i Internet
  Internet address is 10.1.12.1/24
```

Better Source of Information

For an OSPF-learned route to be installed in the routing table, it has to be the most believable routing source. Recall that this is based on administrative distance (AD). OSPF's AD is 110 for all learned routes: intra, inter, and external. Therefore, if there is another source that is educating the same router about the exact same network and that

source has a better AD, the source with the better AD wins, and its information will be installed in the routing table. Example 15-25 is displaying only the OSPF-installed routes in the router. Notice that there is no OSPF entry for the network 10.1.1.0/24 and 10.1.12.0/24.

Example 15-25 Sample show ip route ospf Command Output

```
R1#show ip route ospf
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is 10.1.12.2 to network 0.0.0.0

O*E2  0.0.0.0/0 [110/1] via 10.1.12.2, 01:15:29, GigabitEthernet1/0
      10.0.0.0/8 is variably subnetted, 6 subnets, 2 masks
O IA   10.1.3.0/24 [110/3] via 10.1.12.2, 01:15:29, GigabitEthernet1/0
O IA   10.1.23.0/24 [110/2] via 10.1.12.2, 01:15:29, GigabitEthernet1/0
O IA   203.0.113.0/24 [110/3] via 10.1.12.2, 01:15:29, GigabitEthernet1/0
```

In this case, there is a better source for the 10.1.1.0/24 and 10.1.12.0/24 networks. Example 15-26 displays the output of the **show ip route 10.1.1.0 255.255.255.0** command. It identifies that this network is directly connected and has an AD of 0. Because a directly connected network has an AD of 0 and an OSPF route has an AD of 110, the directly connected source is installed in the routing table.

Example 15-26 Sample show ip route 10.1.1.0 255.255.255.0 Command Output

```
R1#show ip route 10.1.1.0 255.255.255.0
Routing entry for 10.1.1.0/24
  Known via "connected", distance 0, metric 0 (connected, via interface)
  Routing Descriptor Blocks:
  * directly connected, via GigabitEthernet0/0
    Route metric is 0, traffic share count is 1
```

But wait, you might be questioning whether 10.1.1.0/24 is in the LSDB, because it is directly connected. Remember, when an interface is participating in the routing process, its network will be injected into the LSDB as a Type 1 (Router) LSA. You can verify this with the **show ip ospf database** command, as shown in Example 15-27. However, there is no listing for 10.1.1.0/24. This is because we are only looking at a summary of the LSAs in the LSDB. If you want to see the specifics of the LSA, you have to open them up. Example 15-28 displays the output of **show ip ospf database router 10.1.12.1**. This command opens the Type 1 Router LSA advertised by the router with the RID 10.1.12.1,

which is R1. It displays that 10.1.1.0/24 is in the LSDB and therefore can be advertised in the OSPF process.

Example 15-27 *Output of show ip ospf database on R1*

```
R1#show ip ospf database

      OSPF Router with ID (10.1.12.1) (Process ID 1)

      Router Link States (Area 1)

Link ID        ADV Router    Age      Seq#           Checksum Link count
10.1.12.1      10.1.12.1      1025     0x80000009    0x006B41  2
10.1.23.2      10.1.23.2      1210     0x8000002D    0x00E7A3  1

      Net Link States (Area 1)

Link ID        ADV Router    Age      Seq#           Checksum
10.1.12.2      10.1.23.2      1210     0x80000007    0x00B307

      Summary Net Link States (Area 1)

Link ID        ADV Router    Age      Seq#           Checksum
10.1.3.0       10.1.23.2      1210     0x80000004    0x00D72E
10.1.23.0      10.1.23.2      1210     0x8000001A    0x00C418
203.0.113.0    10.1.23.2      1210     0x80000004    0x004E88

      Summary ASB Link States (Area 1)

Link ID        ADV Router    Age      Seq#           Checksum
10.1.23.3      10.1.23.2      1210     0x80000003    0x00C629

      Type-5 AS External Link States

Link ID        ADV Router    Age      Seq#           Checksum Tag
0.0.0.0        10.1.23.3      1268     0x80000003    0x00B399  1
```

Example 15-28 *Output of show ip ospf database router 10.1.12.1 on R1*

```
R1#show ip ospf database router 10.1.12.1

      OSPF Router with ID (10.1.12.1) (Process ID 1)

      Router Link States (Area 1)

LS age: 1368
Options: (No TOS-capability, DC)
```

```

LS Type: Router Links
Link State ID: 10.1.12.1
Advertising Router: 10.1.12.1
LS Seq Number: 80000009
Checksum: 0x6B41
Length: 48
Number of Links: 2

Link connected to: a Transit Network
(Link ID) Designated Router address: 10.1.12.2
(Link Data) Router Interface address: 10.1.12.1
Number of MTID metrics: 0
TOS 0 Metrics: 1

Link connected to: a Stub Network
(Link ID) Network/subnet number: 10.1.1.0
(Link Data) Network Mask: 255.255.255.0
Number of MTID metrics: 0
TOS 0 Metrics: 1

```

Having a better source of routing information may not cause users to complain or submit a trouble ticket, because they will probably still be able to access the resources they need to. However, it might be causing suboptimal routing in your network. Review Figure 15-1, which shows a network running two different routing protocols. In this case, which path will be used to send traffic from PC1 to 10.1.1.0/24? If you said the longer EIGRP path, you are correct. Even though it is quicker to use the OSPF path, EIGRP wins by default because it has the lower AD and suboptimal routing occurs.

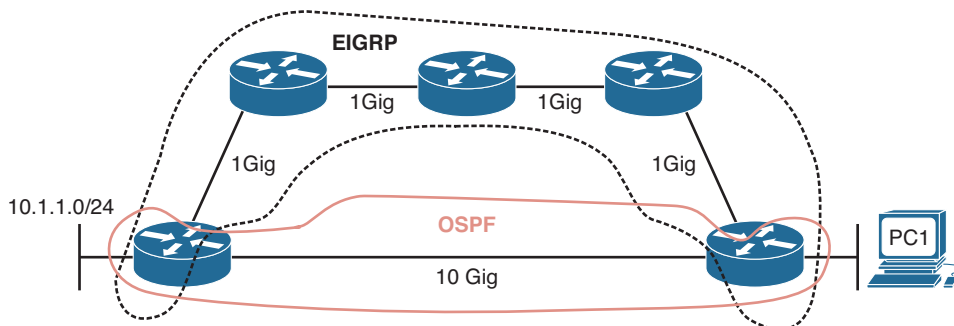


Figure 15-1 Using an EIGRP Path, Which Is Suboptimal

Being able to recognize when a certain routing source should be used and when it should not be used is key to optimizing your network and reducing the number of troubleshooting instances related to “the network is slow.” In this case, we might want to consider increasing the AD of EIGRP or lowering the AD of OSPF to optimize routing.

Route Filtering

A distribute list applied to an Open Shortest Path First (OSPF) process controls which routes are installed into the routing table from the LSDB. Realize that this differs from EIGRP, where it controls routes sent and received between neighbors. The reason this difference exists is that all OSPF routers in an area must have the same LSDB. If you were able to control the routes sent to and received from neighbors, the LSDB would not be the same amongst the routers in the area, which is not permitted.

To apply a route filter to OSPF, the distribute list is applied in OSPF configuration mode inbound (meaning into the routing table), and the routes installed are controlled by ACLs, prefix lists, or route maps. Therefore, when troubleshooting route filtering for OSPF, you need to consider the following:



- Is the distribute list applied in the correct direction?
- If the distribute list is using an ACL, is the ACL correct?
- If the distribute list is using a prefix list, is the prefix list correct?
- If the distribute list is using a route map, is the route map correct?

The **show ip protocols** command will identify whether a distribute list is applied to the OSPF process, as shown in Example 15-29. This example indicates that there are no outbound filters and that there is an inbound filter that is referencing the prefix list called TEST.

Example 15-29 Verifying Route Filters with show ip protocols

```
R1#show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "ospf 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is (prefix-list) TEST
  Router ID 10.1.12.1
  Number of areas in this router is 1. 1 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    10.1.1.1 0.0.0.0 area 1
  Routing on Interfaces Configured Explicitly (Area 1):
    GigabitEthernet1/0
  Passive Interface(s):
    Ethernet0/0
    GigabitEthernet0/0
  Routing Information Sources:
    Gateway         Distance      Last Update
    10.1.23.2         110          00:00:20
    10.1.23.3         110          00:00:20
  Distance: (default is 110)
```


The inbound filter in Example 15-29 is filtered by prefix list TEST. To verify the entries in this prefix list, you issue the **show ip prefix-list TEST** command, as shown in Example 15-30. If an ACL were applied, you would issue the **show access-list** command. If a route map were applied, you would issue the **show route-map** command.

As displayed in Example 15-30, you can verify the command that was used to apply the distribute list in the running configuration.

Example 15-30 *Verifying the OSPF Distribute List and Prefix List*

```
R1#show ip prefix-list TEST
ip prefix-list TEST: 2 entries
  seq 5 deny 10.1.23.0/24
  seq 10 permit 0.0.0.0/0 le 32

R1#show run | section router ospf 1
router ospf 1
  area 1 authentication message-digest
  passive-interface default
  no passive-interface GigabitEthernet1/0
  network 10.1.1.1 0.0.0.0 area 1
  distribute-list prefix TEST in
```

Notice in Example 15-31 that the LSDB still has the 10.1.23.0/24 network listed but that it is not installed in the routing table because of the distribute list that is denying 10.1.23.0/24 from being installed.

Example 15-31 *Verifying OSPF Routes and LSDB After a Distribute List Is Applied*

```
R1#show ip ospf database

        OSPF Router with ID (10.1.12.1) (Process ID 1)

        Router Link States (Area 1)

Link ID        ADV Router    Age         Seq#          Checksum Link count
10.1.12.1      10.1.12.1     16         0x80000011   0x005B49 2
10.1.23.2      10.1.23.2     13         0x80000033   0x00DBA9 1

        Net Link States (Area 1)

Link ID        ADV Router    Age         Seq#          Checksum
10.1.12.2      10.1.23.2     12         0x8000000D   0x00A70D

        Summary Net Link States (Area 1)

Link ID        ADV Router    Age         Seq#          Checksum
10.1.3.0       10.1.23.2     16         0x80000002   0x00DB2C
```

```

10.1.23.0      10.1.23.2      16      0x80000002  0x00F4FF
203.0.113.0    10.1.23.2      16      0x80000002  0x005286

Summary ASB Link States (Area 1)

Link ID      ADV Router    Age      Seq#          Checksum
10.1.23.3    10.1.23.2    18      0x80000001  0x00CA27

Type-5 AS External Link States

Link ID      ADV Router    Age      Seq#          Checksum Tag
0.0.0.0      10.1.23.3    779     0x80000005  0x00AF9B 1

R1#show ip route
...output omitted...

Gateway of last resort is 10.1.12.2 to network 0.0.0.0

O*E2  0.0.0.0/0 [110/1] via 10.1.12.2, 00:00:02, GigabitEthernet1/0
      10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
C      10.1.1.0/24 is directly connected, GigabitEthernet0/0
L      10.1.1.1/32 is directly connected, GigabitEthernet0/0
O IA   10.1.3.0/24 [110/3] via 10.1.12.2, 00:00:02, GigabitEthernet1/0
C      10.1.12.0/24 is directly connected, GigabitEthernet1/0
L      10.1.12.1/32 is directly connected, GigabitEthernet1/0
O IA   203.0.113.0/24 [110/3] via 10.1.12.2, 00:00:02, GigabitEthernet1/0

```

Stub Area Configuration



Because all routers in an area need to have the same LSDB, you cannot manipulate the LSAs within an area; however, you can manipulate LSAs that are flowing between areas by using the stub and NSSA OSPF features.

When you create stub or NSSA areas, you suppress Type 5 LSAs from entering into an area at the ABR. With totally stubby and totally NSSA areas, you suppress Type 5 and Type 3 LSAs from entering into an area at the ABR. The routes that would have been learned via the Type 5 and Type 3 LSAs in the area are now replaced by a default route. Because there is a default route, the router has lost visibility of the overall network, and this could produce suboptimal routing if not implemented correctly in highly redundant environments.

As a result, if you are expecting a Type 5 or Type 3 LSA for a specific route but it is not showing up in the area, verify whether the area is a stub or NSSA area and determine the types of routes that are being suppressed. You can verify whether the area connected to the router is a stub or NSSA area by using the **show ip ospf** command, as shown in Example 15-32.

Example 15-32 *Determining the Type of OSPF Areas*

```

R1#show ip ospf
Routing Process "ospf 1" with ID 10.1.12.1
Start time: 02:23:19.824, Time elapsed: 02:08:52.184
...output omitted...
Reference bandwidth unit is 100 mbps
  Area 1
    Number of interfaces in this area is 2
    It is a stub area
    Area has no authentication
    SPF algorithm last executed 00:05:46.800 ago
...output omitted...

```

However, remember that when implementing totally stub or totally NSSA areas you are only configuring the **no-summary** keyword on the ABR. Therefore, it is best to review the output of **show ip ospf** on the ABR, as shown in Example 15-33. In this example, R2 is configured to suppress Type 3 and Type 5 LSAs from entering into area 1. It will replace them with a default route with a cost of 1.

Example 15-33 *Determining the Type of OSPF Area on the ABR*

```

R2#show ip ospf
Routing Process "ospf 1" with ID 10.1.23.2
Start time: 02:39:09.376, Time elapsed: 15:19:40.352
...output omitted...
  Flood list length 0
  Area 1
    Number of interfaces in this area is 1
    It is a stub area, no summary LSA in this area
    Generates stub default route with cost 1
    Area has no authentication
...output omitted...

```

Interface Is Shut Down

As discussed earlier, once the OSPF process is enabled on the interface, the network the interface is part of (the directly connected entry in the routing table) is injected into the OSPF process. If the interface is shut down, there is no directly connected entry for the network in the routing table. Therefore, the network does not exist, and no network can be injected into the OSPF process. The interface has to be up/up for routes to be advertised or for neighbor relationships to be formed.

Wrong Designated Router Was Elected

A multiaccess network can have multiple routers residing on a common network segment. Rather than having all routers form a full mesh of adjacencies with one another, a designated router (DR) will be elected, and all other routers on the segment form a full adjacency with the DR, as illustrated in Figure 15-2. The rest of the routers will form a 2Way adjacency with each other, and if a BDR exists, they will form a full adjacency with the BDR as well.

A DR is elected based on router priority, with larger priority values being more preferable. If routers have equal priorities, the DR is elected based on the highest OSPF RID. A BDR is also elected based on the same criteria. Routers on the multiaccess network form full adjacencies with the BDR in case the DR becomes unavailable.

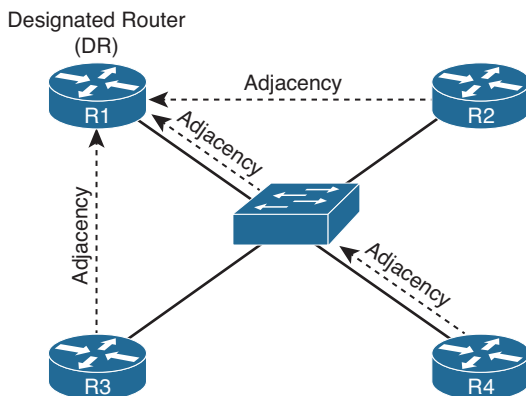


Figure 15-2 DR Election in an Ethernet Network

It does not matter which router is elected as the DR in a multiaccess Ethernet topology or a full-mesh Frame Relay topology, because every router is able to reach the DR since the Layer 2 topology lines up with the Layer 3 addressing. However, over a hub-and-spoke nonbroadcast multiaccess (NBMA) network such as Frame Relay or with a Dynamic Multipoint VPN (DMVPN), it does matter who the DR is because the underlying Layer 2 topology does not line up with the Layer 3 addressing.



Refer to Figure 15-3 which displays a hub-and-spoke Frame Relay or DMVPN network. The multipoint interface (single physical interface or mGRE [multipoint generic routing encapsulation] tunnel interface) provides connectivity to multiple routers in the same subnet out the single interface, like Ethernet. However, in this case, the Layer 2 topology is not the same as the Layer 3 topology. The Layer 3 topology is indicating that all routers are directly reachable out the interfaces (same subnet). But the Layer 2 topology says otherwise. You cannot directly reach R2 from R3 and vice versa. You have to go through R1.

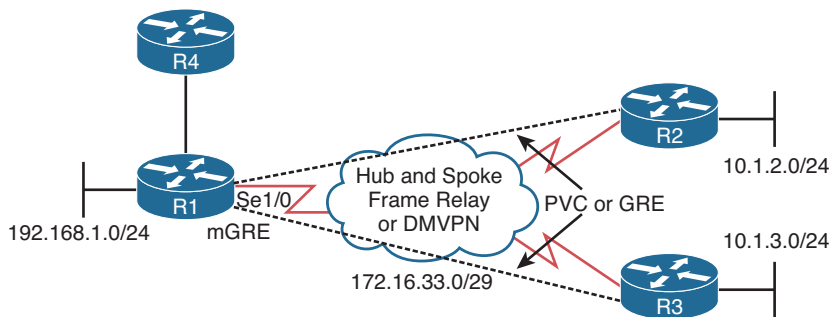


Figure 15-3 *Hub and Spoke*

Figure 15-4 shows the wrong DR placement. The DR router needs to be reachable via a single hop because of how OSPF neighbor relationships are formed and how routers communicate with the DR. Hellos are established with the multicast address 224.0.0.5, and the DR is reachable at the multicast address 224.0.0.6. Packets destined to these two multicast addresses will not be relayed by other routers. Because the DR is responsible for relaying learned routes in a multiaccess network, it needs to be centrally located. Therefore, if R2 were the DR, R3 would not be able to form an adjacency with it because R1 will not relay the hello packet. Therefore, R3 cannot communicate with the DR, meaning that it cannot tell the DR about the 10.1.3.0 network, and as a result, no other router will learn about the 10.1.3.0/24 network.

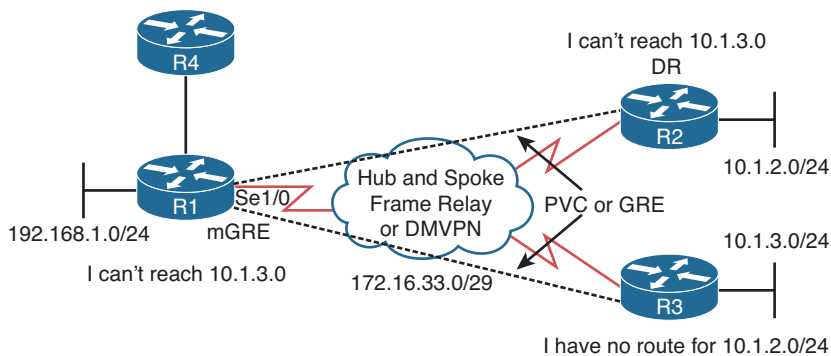


Figure 15-4 *Wrong DR Placement*

In this case, you need to control who the DR is. It has to be R1 to ensure that all routers are able to send LSAs to it and receive LSAs from it, as shown in Figure 15-5.

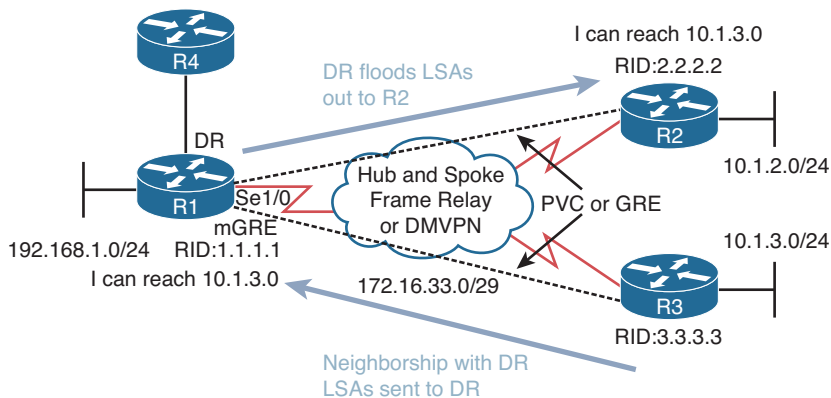


Figure 15-5 Correct DR Placement

To verify the DR placement, issue the command `show ip ospf interface interface_type interface_number` on each of the routers. Example 15-34 indicates that R1 considers the router with the RID 3.3.3.3 as the DR at interface 172.16.33.6. R2 considers itself as the DR and R1 as the BDR. R3 considers itself a DR and R1 as a BDR. Therefore, we have two DRs in this hub-and-spoke environment. As a result, routes will not be successfully learned by all routers in the topology.

Example 15-34 Verifying the DR

```
R1#show ip ospf interface ser 1/0
Serial1/0 is up, line protocol is up
  Internet Address 172.16.33.4/29, Area 0, Attached via Network Statement
  Process ID 1, Router ID 1.1.1.1, Network Type NON_BROADCAST, Cost: 64
  Topology-MTID      Cost      Disabled      Shutdown      Topology Name
    0                 64         no           no           Base
  Transmit Delay is 1 sec, State BDR, Priority 1
  Designated Router (ID) 3.3.3.3, Interface address 172.16.33.6
  Backup Designated router (ID) 1.1.1.1, Interface address 172.16.33.4
  Timer intervals configured, Hello 30, Dead 120, Wait 120, Retransmit 5
  ...output omitted...

R2#show ip ospf interface ser 1/0
Serial1/0 is up, line protocol is up
  Internet Address 172.16.33.5/29, Area 0, Attached via Network Statement
  Process ID 1, Router ID 2.2.2.2, Network Type NON_BROADCAST, Cost: 64
  Topology-MTID      Cost      Disabled      Shutdown      Topology Name
    0                 64         no           no           Base
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 2.2.2.2, Interface address 172.16.33.5
  Backup Designated router (ID) 1.1.1.1, Interface address 172.16.33.4
  Timer intervals configured, Hello 30, Dead 120, Wait 120, Retransmit 5
  ...output omitted...
```

```

R3#show ip ospf interface ser 1/0
Serial1/0 is up, line protocol is up
  Internet Address 172.16.33.6/29, Area 0, Attached via Network Statement
  Process ID 1, Router ID 3.3.3.3, Network Type NON_BROADCAST, Cost: 64
  Topology-MTID      Cost      Disabled      Shutdown      Topology Name
        0             64         no           no           Base
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 3.3.3.3, Interface address 172.16.33.6
  Backup Designated router (ID) 1.1.1.1, Interface address 172.16.33.4
  Timer intervals configured, Hello 30, Dead 120, Wait 120, Retransmit 5
  ...output omitted...

```

To fix this issue, you need to force R1 to be the DR by preventing R2 and R3 from ever wanting to be a DR. On R2 and R3, you go to interface configuration mode and set the OSPF priority to 0, as shown in Example 15-35.

Example 15-35 *Changing OSPF Priority on Spokes*

```

R2#config t
R2(config)#int ser 1/0
R2(config-if)#ip ospf priority 0

R3#config t
R3(config)#int ser 1/0
R3(config-if)#ip ospf priority 0

```

Now the output of `show ip ospf interface ser 1/0` on R1, as shown in Example 15-36, indicates that it is the DR and that there are no BDRs, because we never want a spoke to back up the DR because it would cause the same problem we discussed earlier.

Example 15-36 *Verifying the Hub Router Is the DR*

```

R1#show ip ospf interface ser 1/0
Serial1/0 is up, line protocol is up
  Internet Address 172.16.33.4/29, Area 0, Attached via Network Statement
  Process ID 1, Router ID 1.1.1.1, Network Type NON_BROADCAST, Cost: 64
  Topology-MTID      Cost      Disabled      Shutdown      Topology Name
        0             64         no           no           Base
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 1.1.1.1, Interface address 172.16.33.4
  No backup designated router on this network
  Old designated Router (ID) 3.3.3.3, Interface address 172.16.33.6
  ...output omitted...

```

Duplicate Router IDs

The RID uniquely identifies the routers in the OSPF domain. Because the RID is used during the formation of neighbor relationships and to determine which router is advertising a specific LSA, it is imperative that the RIDs are unique in the domain. If there are duplicate RIDs, the network issues can vary. For example, in the same area, the routers are going to see a Type 1 Router LSA about networks they do not know about from a RID the same as theirs. Therefore, they think they generated the LSA. A router will not use information contained in an LSA they receive that was generated by them because it means there is a loop. However, the LSA is not from itself, it just has the same RID, and as a result we have missing routes on various routers in the domain.

In Figure 15-6, the Type 1 Router LSA from R1 is ignored by R3 because the LSA has the same RID as R3 and so R3 thinks it is its own LSA. Therefore, R3 does not learn about 10.1.1.0/24. The same is true for R1; it does not learn about 10.1.3.0/24 because it is ignoring the LSA that R3 sent because it has the same RID.

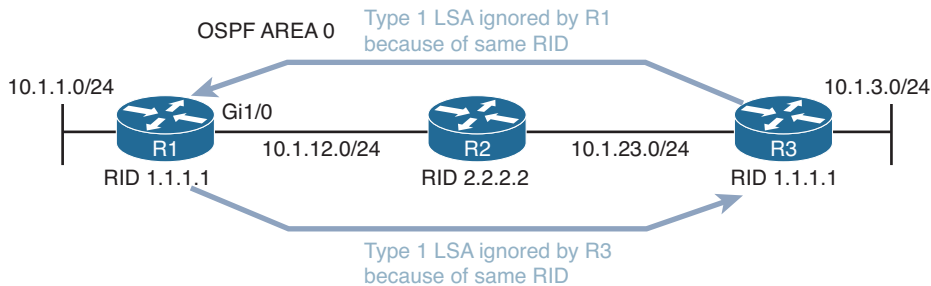


Figure 15-6 Duplicate RIDs in the Same Area

What about duplicate RIDs in different areas? This would cause the physical OSPF topology to be different from what the SPF algorithm sees it as. Refer to Figure 15-7, which displays an OSPF domain with duplicate RIDs in different areas. R1 and R4 both have a RID of 1.1.1.1. As you can see, R2 is going to see the router with the RID in both area 0 and area 1 (which to R2 is technically the same router, but in this case, physically it is not). This can cause routing issues because some routes may not be passed between areas, causing the LSDB and the routing tables to be incomplete.

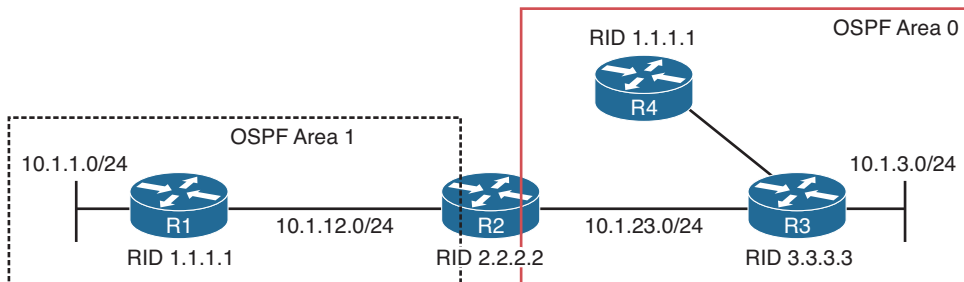


Figure 15-7 Duplicate RIDs in Different Areas

If you have exhausted all possible reasons as to why routes are not appearing in the LSDB or the routing table, take a look at the RIDs of the routers using the **show ip protocols** command, as shown in Example 15-37.

Example 15-37 Verifying OSPF RID

```
R1#show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "ospf 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 1.1.1.1
  Number of areas in this router is 1. 0 normal 1 stub 0 nssa
  Maximum path: 4
  ...output omitted...
```

Troubleshooting Miscellaneous OSPFv2 Issues

So far, your focus has been on troubleshooting issues related to OSPFv2 neighbor relationships and routes. Now your focus will be on tracking LSAs through the network, route summarization, discontinuous areas, load balancing, and default routes.

Tracking OSPF Advertisements Through a Network

When troubleshooting an OSPF issue, tracking the path of OSPF advertisements can be valuable in determining why certain entries are in a router's LSDB.

As an example, notice network 192.168.1.0/24 in the topology provided in Figure 15-8, and consider how this network is entered into the LSDB of the other OSPF routers.

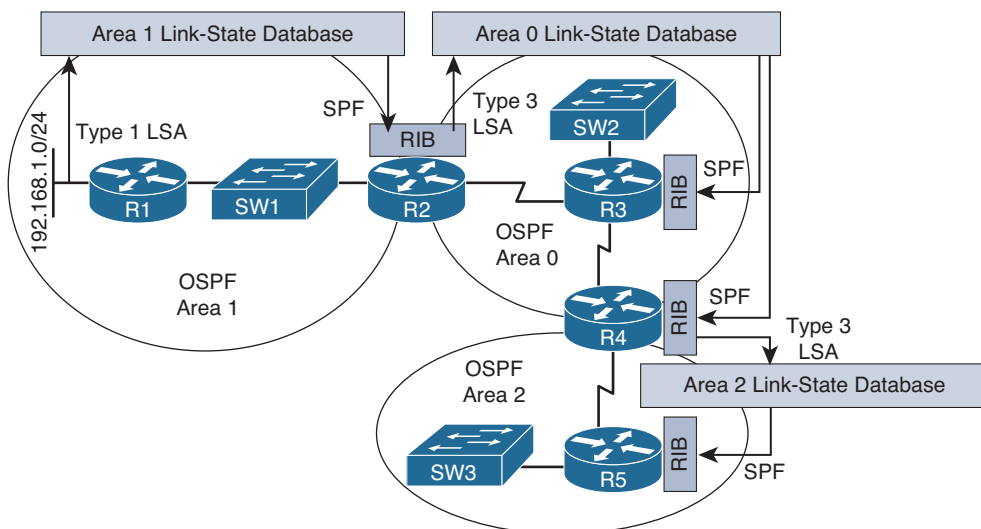


Figure 15-8 Tracking an OSPF Advertisement

The following steps describe how network 192.168.1.0/24, which is directly connected to router R1, is learned by the LSDB of routers R2, R3, R4, and R5:

- Step 1.** Router R1 creates a Type 1 Router LSA for the 192.168.1.0/24 network in the area 1 link-state database and floods it into area 1.
- Step 2.** Router R2 receives the Router LSA for 192.168.1.0/24 and places it in the area 1 link-state database. R2 runs the shortest path first (SPF) algorithm to determine the best path through area 1 to reach the 192.168.1.0/24 network. The best result is placed in R2's routing table (RIB).
- Step 3.** Router R2 informs area 0 routers about the network 192.168.1.0/24 by injecting a Type 3 LSA about the network into the link-state database of area 0 and flooding it into area 0. This LSA includes the cost to reach the 192.168.1.0/24 network, from the perspective of router R2.
- Step 4.** Each of the other area 0 routers (that is, routers R3 and R4) receive the Type 3 LSA and add it to their area 0 LSDB. They run the SPF algorithm to determine the cost to reach router R2. This cost is then added to the cost router R2 advertised in its Type 3 LSA, and the result is stored in the RIB for routers R3 and R4.
- Step 5.** Router R4 informs area 2 routers about the network 192.168.1.0/24 by injecting a Type 3 LSA about the network into the link-state database of area 2 and flooding it into area 2. This LSA includes the cost to reach the 192.168.1.0/24 network, from the perspective of router R4.
- Step 6.** Each of the routers in area 2 receive the Type 3 LSA and add it to their area 2 LSDB. They run the SPF algorithm to determine the cost to reach router R4. This cost is then added to the cost router R4 advertised in its Type 3 LSA, and the result is stored in the RIB of the routers.

To successfully troubleshoot OSPF-related issues, you should have a solid understanding of this process and the different types of OSPF LSAs. Table 15-4 lists the common LSA types you will encounter when troubleshooting a Cisco-based OSPF network.



Table 15-4 *OSPF LSAs*

LSA Type	Description
1	All OSPF routers source Type 1 LSAs. These advertisements list information about directly connected subnets, the OSPF connection types of a router, and the known OSPF adjacencies of a router. A Type 1 LSA is not sent out of its local area.
2	The designated router on a multiaccess network sends a Type 2 LSA for that network if the network contains at least two routers. A Type 2 LSA contains a listing of routers connected to the multiaccess network and, like a Type 1 LSA, is constrained to its local area.

LSA Type	Description
3	A Type 3 LSA is sourced by an ABR. Each Type 3 LSA sent into an area contains information about a network reachable in a different area. Note that network information is exchanged only between the backbone area and a nonbackbone area, as opposed to being exchanged between two nonbackbone areas.
4	Similar to a Type 3 LSA, a Type 4 LSA is sourced by an ABR. However, instead of containing information about OSPF networks, a Type 4 LSA contains information stating how to reach an ASBR.
5	A Type 5 LSA is sourced by an ASBR and contains information about networks reachable outside the OSPF domain. A Type 5 LSA is sent to all OSPF areas, except for stub areas. Note that the ABR for a stub area sends default route information into the stub area, rather than the network-specific Type 5 LSAs.
7	A Type 7 LSA is sourced from an ASBR within a not-so-stubby area (NSSA). Whereas a stub area cannot connect to an external autonomous system, an NSSA can. The Type 7 LSA only exists in the NSSA; therefore, the external routes are announced by the ABR(s) of the NSSA into Area 0 using Type 5 LSAs. In addition, like a stub area, external routes known to another OSPF area are not forwarded into an NSSA since Type 5 LSAs are not permitted in an NSSA.

Route Summarization

OSPF is strict about where route summarization can occur. With OSPF, manual route summarization is enabled on an area by area basis on an ABR to summarize routes as they enter or leave an area or on an ASBR to summarize external routes being injected into an area. Therefore, when troubleshooting route summarization you want to keep the following in mind:



- Did you enable route summarization on the correct router?
- Did you enable route summarization for the correct area?
- Did you create the appropriate summary route?

You can verify all of these using the **show ip ospf** command, as shown in Example 15-38. In this example, R2 is an area border router, and there is a summary address of 10.1.0.0/16 for area 1 that is currently active and being advertised into area 0.

Example 15-38 Verifying Interarea Route Summarization with show ip ospf

```
R2#show ip ospf
Routing Process "ospf 1" with ID 2.2.2.2
...output omitted...
Event-log enabled, Maximum number of events: 1000, Mode: cyclic
It is an area border router
```

```

Router is not originating router-LSAs with maximum metric
...output omitted...
Reference bandwidth unit is 100 mbps
  Area BACKBONE(0)
    Number of interfaces in this area is 1
    Area has no authentication
    SPF algorithm last executed 00:03:27.000 ago
    SPF algorithm executed 14 times
    Area ranges are
    Number of LSA 6. Checksum Sum 0x033162
    Number of opaque link LSA 0. Checksum Sum 0x000000
    Number of DCbitless LSA 0
    Number of indication LSA 0
    Number of DoNotAge LSA 0
    Flood list length 0
  Area 1
    Number of interfaces in this area is 1
    Area has no authentication
    SPF algorithm last executed 00:03:27.024 ago
    SPF algorithm executed 13 times
    Area ranges are
      10.1.0.0/16 Active(1) Advertise
    Number of LSA 9. Checksum Sum 0x0555F1
...output omitted...

```

Remember that interarea summaries are created on ABRs with the **area range** command and that external summaries are created on ASBRs with the **summary-address** command.

When a summary route is created on a router, so is a summary route to Null0, as shown in Example 15-39. This route to Null0 is created to prevent routing loops. It is imperative that this route is in the table to ensure that if a packet is received by this router destined to a network that falls within the summary that the router does not really know how to reach (longer match), it will be dropped. If the route to Null0 did not exist, and there were a default route on the router, the router would forward the packet via the default route, and then the next-hop router would end up forwarding it back to this router, because it is using the summary route, then the local router would then forward it based on the default route, and then it would come back. This is a routing loop.

It is important that you create accurate summary routes to ensure that your router is not advertising networks in the summary route that it does not truly know how to reach. If it does, it is possible that it might receive packets to destinations that fall within the summary that it really does not know how to reach. If this is the case, it means that packets will be dropped because of the route to Null0.

Example 15-39 *Verifying Local Summary Route to Null0*

```
R2#show ip route | include Null
O          10.1.0.0/16 is a summary, 00:16:07, Null0
```

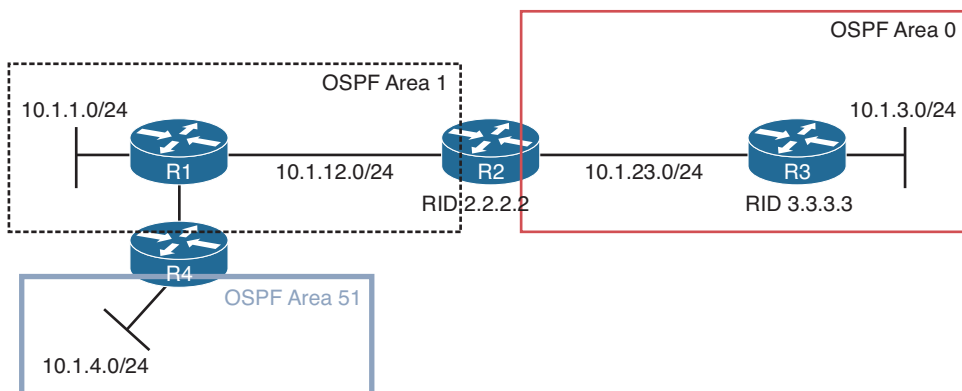
Unlike EIGRP, which gives the route to Null0 an AD of 5, the route to Null0 gets an AD of 110 with OSPF, as shown in Example 15-40. This does not ensure that it is more believable than most of the other sources of routing information. Therefore, it is possible that another better routing source could end up forwarding the traffic for networks that are included in the summary route to Null0.

Example 15-40 *Verifying the AD of a Local Summary Route to Null 0*

```
R2#show ip route 10.1.0.0 255.255.0.0
Routing entry for 10.1.0.0/16
  Known via "ospf 1", distance 110, metric 1, type intra area
  Routing Descriptor Blocks:
    * directly connected, via Null0
      Route metric is 1, traffic share count is 1
```

Discontiguous Areas

In a multiarea OSPF network, a backbone area (numbered *area 0*) must exist, and all other areas must connect to area 0. If an area is not physically adjacent to area 0, routes will not be successfully learned by all routers in the OSPF domain. To solve this issue, a *virtual link* can be configured to logically connect the nonadjacent area with area 0. Figure 15-9 shows area 51 not physically connected to area 0. This results in the 10.1.4.0 network not being learned by any other router in the OSPF domain, because an ABR is needed to send Type 3 LSAs into area 0. R4 is not an ABR in this case because the requirement for an ABR is that one interface must be in area 0 and one or more interfaces in any other area(s). In this case, R4 has no interfaces in area 0.

**Figure 15-9** *Area 51 Not Directly Connected to Area 0*

Now refer to Figure 15-10, which is showing a similar topology; however, area 0 is discontinuous. This will result in LSAs not being successfully flooded though the OSPF domain and, as a result, incomplete routing tables.

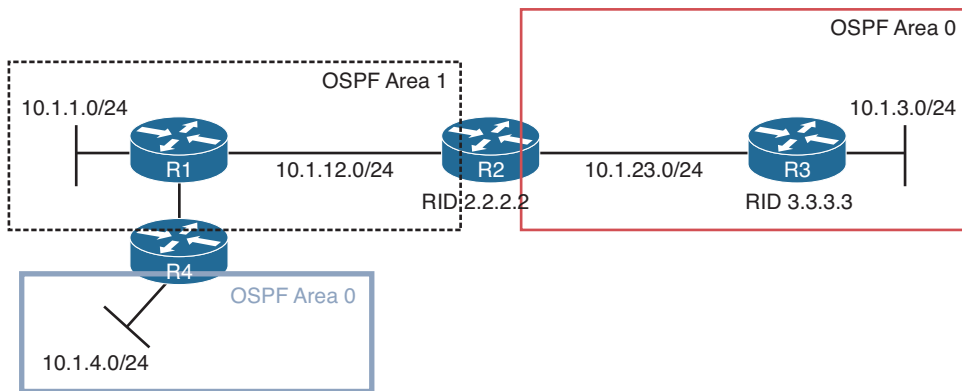


Figure 15-10 *Discontiguous Area 0*

You need to be able to recognize these OSPF design issues, understand how to troubleshoot them and implement a solution. The solution is virtual links. A virtual link in both these examples is created through area 1, which will be known as the transit area because it will transit LSAs from area 51 to area 0 or from area 0 to area 0. Note that virtual links are a temporary solution for these issues. A permanent redesign/fix should be performed as soon as possible.

The virtual link is created between the routers connected to the transit area using their RIDs and the transit area number as shown in Figure 15-11. The router OSPF configuration mode command on R2 is **area 1 virtual-link 4.4.4.4**, and the command on R4 is **area 1 virtual-link 2.2.2.2**. Once the virtual link is established, R4 becomes an ABR since it has an interface (virtual interface in this case) in area 0. Common issues related to failed virtual links include a misconfigured area number or RID. If you type in the area number you are trying to connect to area 0 instead of the transit area number, the virtual link will fail to form. If you use the interface IP address rather than the RID, the virtual link will fail to form.

Example 15-41 displays the output of **show ip ospf neighbor** on R2. Notice how there is a new neighbor relationship with 4.4.4.4 but that the local interface is OSPF_VL0, which is referring to the virtual link interface.

Example 15-41 *Verifying a Neighbor Relationship over a Virtual Link*

```
R2#show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
4.4.4.4	0	FULL/-	-	10.1.14.4	OSPF_VL0
3.3.3.3	1	FULL/BDR	00:00:34	10.1.23.3	GigabitEthernet1/0
1.1.1.1	1	FULL/BDR	00:00:35	10.1.12.1	GigabitEthernet0/0

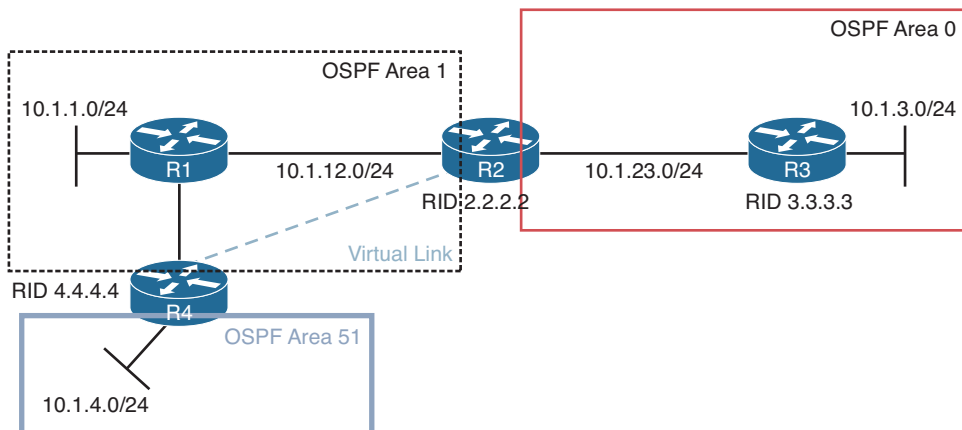


Figure 15-11 LSA Flooding with Virtual Links

Example 15-42 displays the output of `show ip ospf virtual-links`, which provides more details about the virtual link. It is not only important to verify that the virtual link is up but that the state is full, which verifies that LSAs have been successfully exchanged.

Key Topic

Example 15-42 Verifying the Virtual Link

```
R2#show ip ospf virtual-links
Virtual Link OSPF_VL0 to router 4.4.4.4 is up
  Run as demand circuit
  DoNotAge LSA allowed.
  Transit area 1, via interface GigabitEthernet0/0
Topology-MTID      Cost      Disabled      Shutdown      Topology Name
  0                2         no            no            Base
Transmit Delay is 1 sec, State POINT_TO_POINT,
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
Hello due in 00:00:09
Adjacency State FULL (Hello suppressed)
Index 2/3, retransmission queue length 0, number of retransmission 0
First 0x0(0)/0x0(0) Next 0x0(0)/0x0(0)
Last retransmission scan length is 0, maximum is 0
Last retransmission scan time is 0 msec, maximum is 0 msec
```

Load Balancing

OSPF only supports equal-cost load balancing. Therefore, when troubleshooting load balancing for OSPF, your two primary points of concern are the overall end-to-end cost and the maximum number of paths permitted for load balancing. To verify the maximum

number of equal-cost paths an OSPF router is currently configured to support, use the **show ip protocols** command, as shown in Example 15-43. In this example, R1 is currently using the default value of 4.

If your topology is showing multiple paths to reach certain networks in your organization but they are not all showing up in the routing table, it is more than likely because 1) they are not equal-cost paths or 2) the maximum paths value is configured too low.

Example 15-43 *Verifying the Maximum Number of Paths for Load Balancing*

```
R1#show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "ospf 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is (prefix-list) TEST
  Router ID 1.1.1.1
  Number of areas in this router is 2. 2 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    10.1.1.1 0.0.0.0 area 1
  Routing on Interfaces Configured Explicitly (Area 1):
    GigabitEthernet1/0
  ...output omitted...
```

Default Route

With OSPF, a static default route is injected into the routing process using the **default-information originate** command, not the **redistribute static** command. Therefore, if you are troubleshooting why a static default route is not being advertised in the OSPF process, use the **show run | section router ospf** command to verify that the **default-information originate** command is being used.

OSPFv2 Trouble Tickets

This section presents various trouble tickets relating to the topics discussed earlier in the chapter. The purpose of these trouble tickets is to give a process that you can follow when troubleshooting in the real world or in an exam environment. All trouble tickets in this section are based on the topology depicted in Figure 15-12.

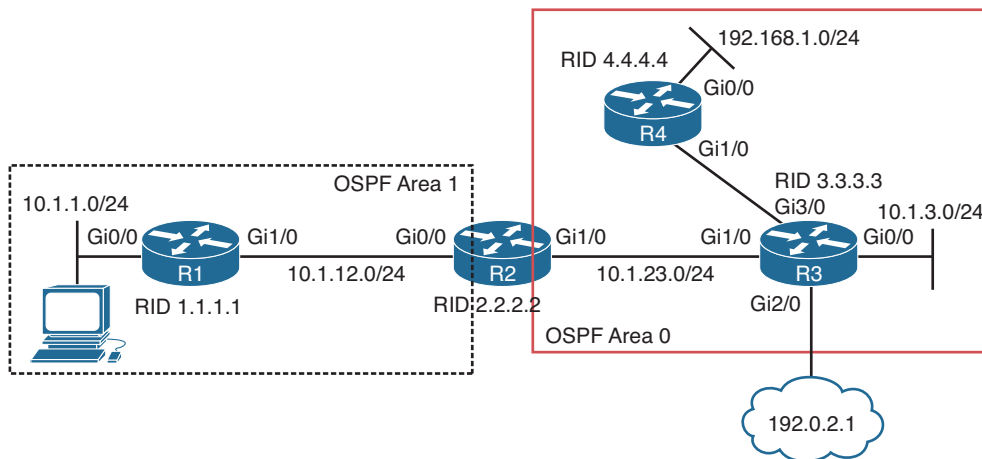


Figure 15-12 OSPFv2 Trouble Tickets Topology

Trouble Ticket 15-1

Problem: Users in the 10.1.1.0/24 network indicate that they are not able to access resources in the 192.168.1.0/24 network.

As always, the first item on the list for troubleshooting is to verify the problem. You access a PC in the 10.1.1.0/24 network and ping an IP address in the 192.168.1.0/24 network and it is successful (0% loss), as shown in Example 15-44. However, notice that the reply is from the default gateway at 10.1.1.1, and it states *Destination host unreachable*. Therefore, it was technically not successful.

Example 15-44 Destination Unreachable Result from a ping Command on a PC

```
C:\>ping 192.168.1.10

Pinging 192.168.1.10 with 32 bytes of data:

Reply from 10.1.1.1: Destination host unreachable.
Reply from 10.1.1.1: Destination host unreachable.
Reply from 10.1.1.1: Destination host unreachable.
Reply from 10.1.1.1: Destination host unreachable.

Ping statistics for 192.168.1.10:
    Packets: Sent = 4, Received = 4, lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

The result of this ping tells us two very important things: 1) The PC can reach the default gateway; 2) The default gateway does not know how to get to the 192.168.1.0/24 network. Therefore, we can focus our attention on R1 and work from there.

On R1, you issue the same ping, but it fails, as shown in Example 15-45.

Example 15-45 *Failed Ping from R1 to 192.168.1.10*

```

R1#ping 192.168.1.10
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.10, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)

```

Next, you check R1's routing table with the **show ip route** command and notice that there are only connected routes in the routing table, as shown in Example 15-46. R1 is not learning any routes from R2.

Example 15-46 *show ip route Output on R1*

```

R1#show ip route
...output omitted...
Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C       10.1.1.0/24 is directly connected, GigabitEthernet0/0
L       10.1.1.1/32 is directly connected, GigabitEthernet0/0
C       10.1.12.0/24 is directly connected, GigabitEthernet1/0
L       10.1.12.1/32 is directly connected, GigabitEthernet1/0

```

According to Figure 15-12, OSPF is the routing protocol in use. Therefore, you issue the **show ip protocols** command to verify that OSPF is running on R1. Example 15-47 displays the **show ip protocols** output and confirms that OSPF process 1 is in operation on R1.

Example 15-47 *show ip protocols Output on R1*

```

R1#show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "ospf 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 1.1.1.1
  Number of areas in this router is 2. 2 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    10.1.1.1 0.0.0.0 area 1
  Routing on Interfaces Configured Explicitly (Area 1):
    GigabitEthernet1/0
  Passive Interface(s):
    Ethernet0/0
    GigabitEthernet0/0
  Routing Information Sources:

```

Gateway	Distance	Last Update
4.4.4.4	110	01:20:29
2.2.2.2	110	00:48:38
3.3.3.3	110	01:20:29
10.1.23.2	110	16:56:39
203.0.113.3	110	17:10:26
Distance: (default is 110)		

Next you check to see whether R1 has any OSPF neighbors. According to the topology R2 should be a neighbor. To verify OSPF neighbors, you issue the **show ip ospf neighbor** command on R1, as shown in Example 15-48. According to the output, R1 is a neighbor with R2.

Example 15-48 show ip ospf neighbor Output on R1

```
R1#show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
2.2.2.2	1	FULL/DR	00:00:36	10.1.12.2	GigabitEthernet1/0

Now is the time to be wise. What is the next best step? Some would consider troubleshooting why the routes are missing on R1 by looking at various features and parameters associated with R1. However, the 192.168.1.0/24 network is in a different area. Who is responsible for telling R1 about 192.168.1.0/24? Is it R4? No. Is it R2? Yes. R2 sends a Type 3 Summary LSA into area 1 which tells area 1 about the 192.168.1.0/24 network. Therefore, if R2 does not know about 192.168.1.0/24 then we can stop troubleshooting on R1. This is a great example of how understanding the flow of different LSAs can save you time while troubleshooting.

On R2, you issue the **show ip route** command, as shown in Example 15-49, and confirm that R2 does not know about the 192.168.1.0/24 network either. In fact, it has not learned about any networks in area 0.

Example 15-49 show ip route Output on R2

```
R2#show ip route
...output omitted...
Gateway of last resort is not set
```

	10.0.0.0/8 is variably subnetted, 6 subnets, 3 masks
O	10.1.0.0/16 is a summary, 15:15:33, Null0
O	10.1.1.0/24 [110/2] via 10.1.12.1, 01:33:14, GigabitEthernet0/0
C	10.1.12.0/24 is directly connected, GigabitEthernet0/0
L	10.1.12.2/32 is directly connected, GigabitEthernet0/0
C	10.1.23.0/24 is directly connected, GigabitEthernet1/0
L	10.1.23.2/32 is directly connected, GigabitEthernet1/0

Wait! Be careful with the previous statement. Remember, with OSPF, distribute lists are used to permit or deny routes from being installed in the routing table from the LSDB. Therefore, you may be learning about them just not installing them.

Example 15-50 shows the output of the LSDB on R2, and as you can see, there are no area 0 Type 1 Router LSAs from R3 (3.3.3.3) or R4 (4.4.4.4). Therefore, we can now officially say that R2 has not been educated about the networks that are missing.

Example 15-50 *show ip ospf database Output on R2 Confirming that Routes are Missing*

```
R2#show ip ospf database
```

```
OSPF Router with ID (2.2.2.2) (Process ID 1)
```

```
Router Link States (Area 0)
```

Link ID	ADV Router	Age	Seq#	Checksum	Link count
2.2.2.2	2.2.2.2	316	0x80000025	0x003B9F	1

```
Summary Net Link States (Area 0)
```

Link ID	ADV Router	Age	Seq#	Checksum
10.1.0.0	2.2.2.2	1339	0x8000001C	0x00927B

```
Router Link States (Area 1)
```

Link ID	ADV Router	Age	Seq#	Checksum	Link count
1.1.1.1	1.1.1.1	1988	0x80000022	0x007843	2
2.2.2.2	2.2.2.2	316	0x80000024	0x0012BA	1

```
Net Link States (Area 1)
```

Link ID	ADV Router	Age	Seq#	Checksum
10.1.12.2	2.2.2.2	1589	0x8000001C	0x007C75

```
Summary Net Link States (Area 1)
```

Link ID	ADV Router	Age	Seq#	Checksum
10.1.23.0	2.2.2.2	61	0x80000020	0x008C66

To receive LSAs, we must have interfaces participating in the OSPF process, and we must have neighbor relationships. The output of **show cdp neighbors** indicates that R3 is a neighbor and that it is reachable out R2's local Gig1/0 interface, as shown in Example 15-51.

Example 15-51 *Using show cdp neighbors to Verify Router Interfaces*

```
R2#show cdp neighbors
```

```
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone,
                  D - Remote, C - CVTA, M - Two-port Mac Relay
```

Device ID	Local Intrfce	Holdtme	Capability	Platform	Port ID
R3	Gig 1/0	178	R	7206VXR	Gig 1/0
R1	Gig 0/0	179	R	7206VXR	Gig 1/0

Issuing the commands **show ip ospf interface brief** and **show ip ospf neighbor**, as shown in Example 15-52, shows that R2's local Gig1/0 interface is participating in the OSPF process but does not have a neighbor on the interface.

Example 15-52 *Verifying OSPF-Enabled Interfaces and Neighbors*

```
R2#show ip ospf interface brief
```

Interface	PID	Area	IP Address/Mask	Cost	State	Nbrs F/C
Gil/0	1	0	10.1.23.2/24	1	DR	0/0
Gi0/0	1	1	10.1.12.2/24	1	DR	1/1

```
R2#show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
1.1.1.1	1	FULL/BDR	00:00:37	10.1.12.1	GigabitEthernet0/0

So, you can now hypothesize that the issue is related to R2 and R3 not having a neighbor adjacency. What would cause this? As our earlier discussion in this chapter indicated, many different issues could cause this. However, if you recall, the majority of them were interface related, and we stated that using the spot-the-difference troubleshooting method would come in handy. Let's do that by examining the output of **show ip ospf interface gigabitethernet 1/0** on R2 and R3, as shown in Example 15-53.

Example 15-53 *Comparing the OSPF Interface Parameters of R2 and R3*

```
R2#show ip ospf interface gigabitEthernet 1/0
```

```
GigabitEthernet1/0 is up, line protocol is up
```

```
Internet Address 10.1.23.2/24, Area 0, Attached via Network Statement
```

```
Process ID 1, Router ID 2.2.2.2, Network Type BROADCAST, Cost: 1
```

Topology-MTID	Cost	Disabled	Shutdown	Topology Name
0	1	no	no	Base

```
Transmit Delay is 1 sec, State DR, Priority 1
```

```
Designated Router (ID) 2.2.2.2, Interface address 10.1.23.2
```

```
No backup designated router on this network
```

```
Timer intervals configured, Hello 11, Dead 44, Wait 44, Retransmit 5
```

```
oob-resync timeout 44
```

```
Hello due in 00:00:08
```

```
Supports Link-local Signaling (LLS)
```

```

Cisco NSF helper support enabled
IETF NSF helper support enabled
Index 1/2, flood queue length 0
Next 0x0(0)/0x0(0)
Last flood scan length is 0, maximum is 3
Last flood scan time is 0 msec, maximum is 4 msec
Neighbor Count is 0, Adjacent neighbor count is 0
Suppress hello for 0 neighbor(s)
Message digest authentication enabled
  Youngest key id is 1

```

R3#show ip ospf interface gigabitEthernet 1/0

```

GigabitEthernet1/0 is up, line protocol is up
  Internet Address 10.1.23.3/24, Area 0, Attached via Network Statement
  Process ID 1, Router ID 3.3.3.3, Network Type BROADCAST, Cost: 1
  Topology-MTID      Cost      Disabled      Shutdown      Topology Name
    0                1         no            no            Base
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 3.3.3.3, Interface address 10.1.23.3
  No backup designated router on this network
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    oob-resync timeout 40
    Hello due in 00:00:04
  Supports Link-local Signaling (LLS)
  Cisco NSF helper support enabled
  IETF NSF helper support enabled
  Index 2/2, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 2
  Last flood scan time is 0 msec, maximum is 4 msec
  Neighbor Count is 0, Adjacent neighbor count is 0
  Suppress hello for 0 neighbor(s)
  Message digest authentication enabled
    Youngest key id is 1

```

- Are the interfaces up? Yes
- Are they in the same subnet? Yes
- Are they in the same area? Yes
- Do the routers have unique RIDs? Yes
- Are they using compatible Network Types? Yes
- Do hello and dead timers match? No (possible reason)
- Do authentication parameters match? Enabled and key matches, but not sure about key string unless we check the running configuration (possible reason)

As you can see in Example 15-53, the hello and dead timers do not match, but they must. Reviewing the output of **show run interface gig 1/0** on R2, as shown in Example 15-54, shows that the command **ip ospf hello-interval 11** was configured.

Example 15-54 *Verifying Interface Configuration on R2*

```
R2#show run interface gigabitEthernet 1/0
Building configuration...

Current configuration : 196 bytes
!
interface GigabitEthernet1/0
 ip address 10.1.23.2 255.255.255.0
 ip ospf authentication message-digest
 ip ospf message-digest-key 1 md5 CISCO
 ip ospf hello-interval 11
 negotiation auto
end
```

Once you remove this command with the **no ip ospf hello-interval 11** command, you receive the following syslog message on R2:

```
%OSPF-5-ADJCHG: Process 1, Nbr 3.3.3.3 on GigabitEthernet1/0 from LOADING to FULL, Loading Done
```

This confirms the adjacency was formed, and reviewing the output of the routing table on R2 using the **show ip route** command confirms that the routes are learned, as shown in Example 15-55.

Example 15-55 *Verifying Routes in the Routing Table on R2*

```
R2#show ip route
...output omitted...
Gateway of last resort is 10.1.23.3 to network 0.0.0.0

O*E2  0.0.0.0/0 [110/1] via 10.1.23.3, 00:01:00, GigabitEthernet1/0
      10.0.0.0/8 is variably subnetted, 8 subnets, 3 masks
O      10.1.0.0/16 is a summary, 00:01:49, Null0
O      10.1.1.0/24 [110/2] via 10.1.12.1, 00:01:00, GigabitEthernet0/0
O      10.1.3.0/24 [110/2] via 10.1.23.3, 00:01:00, GigabitEthernet1/0
C      10.1.12.0/24 is directly connected, GigabitEthernet0/0
L      10.1.12.2/32 is directly connected, GigabitEthernet0/0
C      10.1.23.0/24 is directly connected, GigabitEthernet1/0
L      10.1.23.2/32 is directly connected, GigabitEthernet1/0
O      10.1.34.0/24 [110/2] via 10.1.23.3, 00:01:00, GigabitEthernet1/0
O      192.168.1.0/24 [110/3] via 10.1.23.3, 00:01:00, GigabitEthernet1/0
O      203.0.113.0/24 [110/2] via 10.1.23.3, 00:01:00, GigabitEthernet1/0
```

R1 also knows about the routes now, as shown in Example 15-56, which displays the output of `show ip route` on R1.

Example 15-56 *Verifying Routes in the Routing Table on R1*

```
R1#show ip route
...output omitted...
Gateway of last resort is 10.1.12.2 to network 0.0.0.0

O*E2 0.0.0.0/0 [110/1] via 10.1.12.2, 00:00:13, GigabitEthernet1/0
      10.0.0.0/8 is variably subnetted, 7 subnets, 2 masks
C      10.1.1.0/24 is directly connected, GigabitEthernet0/0
L      10.1.1.1/32 is directly connected, GigabitEthernet0/0
O IA   10.1.3.0/24 [110/3] via 10.1.12.2, 00:00:19, GigabitEthernet1/0
C      10.1.12.0/24 is directly connected, GigabitEthernet1/0
L      10.1.12.1/32 is directly connected, GigabitEthernet1/0
O IA   10.1.23.0/24 [110/2] via 10.1.12.2, 00:00:19, GigabitEthernet1/0
O IA   10.1.34.0/24 [110/3] via 10.1.12.2, 00:00:19, GigabitEthernet1/0
O IA   192.168.1.0/24 [110/4] via 10.1.12.2, 00:00:19, GigabitEthernet1/0
O IA   203.0.113.0/24 [110/3] via 10.1.12.2, 00:00:19, GigabitEthernet1/0
```

Finally, you ping from the PC again, and the ping is successful, as shown in Example 15-57.

Example 15-57 *A Successful Ping from the 10.1.1.0/24 network to the 192.168.1.0/24 network*

```
C:\>ping 192.168.1.10

Pinging 192.168.1.10 with 32 bytes of data:

Reply from 192.168.1.10: bytes=32 time 1ms TTL=128
Reply from 192.168.1.10: bytes=32 time 1ms TTL=128
Reply from 192.168.1.10: bytes=32 time 1ms TTL=128
Reply from 192.168.1.10: bytes=32 time 1ms TTL=128

Ping statistics for 192.168.1.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Trouble Ticket 15-2

Problem: Users in the 10.1.1.0/24 network indicate that they are not able to access resources in the 192.168.1.0/24 network.

As always, the first item on the list for troubleshooting is to verify the problem. You access a PC in the 10.1.1.0/24 network and ping an IP address in the 192.168.1.0/24 network, and it is successful (0% loss), as shown in Example 15-58. However, notice that the reply is from 10.1.23.2 and it states *TTL expired in transit*. Therefore, it was technically not successful.

Example 15-58 *TTL Expired in Transit Result from ping Command on PC*

```
C:\>ping 192.168.1.10

Pinging 192.168.1.10 with 32 bytes of data:

Reply from 10.1.23.2: TTL expired in transit.
Reply from 10.1.23.2: TTL expired in transit.
Reply from 10.1.23.2: TTL expired in transit.
Reply from 10.1.23.2: TTL expired in transit.

Ping statistics for 192.168.1.10:
    Packets: sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

The result of this ping tells us two very important things: 1) The PC can reach the default gateway at 10.1.1.1; 2) the device at 10.1.23.2 expired the packet because the TTL reached 0 and the device sent an ICMP time exceeded message back to the PC.

Pause for a moment and think about this! If the TTL expired in transit, it means that the packet did not reach the destination before the TTL decremented to 0. Each time a router touches the packet, it decrements the TTL by 1. Normally the TTL is set to 255 by default. Unless it was modified, which we did not do, the packet bounced around the network and went through approximately 255 routers before the device at IP 10.1.23.2 decremented the TTL to 0 and sent the ICMP TTL expired message. Because Figure 15-12 clearly shows that there are only four routers from 10.1.1.0/24 to 192.168.1.0/24, the packet is bouncing around the network somewhere. Running a traceroute from the PC will help us identify this as shown in Example 15-59. This example shows that R3 (10.1.23.3) and R2 (10.1.23.2) are bouncing the packet back and forth.

Example 15-59 *Traceroute Showing that R2 and R3 Are Bouncing Packet Back and Forth*

```
C:\>tracert 192.168.1.10

Tracing route to 192.168.1.10 over a maximum of 30 hops

  1    23 ms    15 ms    10 ms    10.1.1.1
  2    36 ms    30 ms    29 ms    10.1.12.2
  3    53 ms    50 ms    39 ms    10.1.23.3
  4    61 ms    39 ms    40 ms    10.1.23.2
```

```

5      61 ms      69 ms      59 ms      10.1.23.3
6      68 ms      50 ms      69 ms      10.1.23.2
7      * ms       78 ms      89 ms      10.1.23.3
8      87 ms      69 ms      * ms      10.1.23.2
...output omitted...
29     175 ms     169 ms     179 ms     10.1.23.3
30     204 ms     189 ms     189 ms     10.1.23.2

Trace complete.

```

We can deduce from this that R3 is not routing the packet correctly. It is sending the packet to R2 instead of R4. Accessing R3 and issuing the **show ip ospf database router 4.4.4.4** command, as shown in Example 15-60, clearly indicates that R3 is learning about the network 192.168.1.0/24 from R4. However, instead of using R4 as a next hop, it is using R2 because it is sending the packets to R2, as shown in the earlier trace.

Example 15-60 *Verifying Whether a Route Is in an OSPF Database*

```

R3#show ip ospf database router 4.4.4.4

      OSPF Router with ID (3.3.3.3) (Process ID 1)

      Router Link States (Area 0)

LS age: 894
Options: (No TOS-capability, DC)
LS Type: Router Links
Link State ID: 4.4.4.4
Advertising Router: 4.4.4.4
LS Seq Number: 80000004
Checksum: 0xEA47
Length: 48
Number of Links: 2

Link connected to: a Transit Network
(Link ID) Designated Router address: 10.1.34.4
(Link Data) Router Interface address: 10.1.34.4
Number of MTID metrics: 0
TOS 0 Metrics: 1

Link connected to: a Stub Network
(Link ID) Network/subnet number: 192.168.1.0
(Link Data) Network Mask: 255.255.255.0
Number of MTID metrics: 0
TOS 0 Metrics: 1

```

Let's look at the routing table to see whether we are installing this network in the routing table. Issuing the command **show ip route ospf** on R3, as shown in Example 15-61, indicates that this OSPF-learned route is not being installed in the routing table.

Example 15-61 *Output of show ip route ospf on R3*

```
R3#show ip route ospf
...output omitted...

Gateway of last resort is 203.0.113.1 to network 0.0.0.0

    10.0.0.0/8 is variably subnetted, 7 subnets, 3 masks
O IA    10.1.0.0/16 [110/2] via 10.1.23.2, 01:25:02, GigabitEthernet1/0
```

Time to hypothesize! What would cause R3 to learn about the route but not install it in the routing table: route filtering, better source, to name a few. However, harness your knowledge and really focus on what is happening.

R3 is routing packets destined to 192.168.1.0/24, which means that there must be some entry in the routing table or policy based routing is enforced.

Issuing the command **show ip route 192.168.1.0 255.255.255.0** on R3 confirms that there is an entry in the routing table on R3, as shown in Example 15-62. However, it is a static entry with an AD of 1 pointing to 10.1.23.2. It looks like we found the problem. There is a better source of routing information according to AD.

Example 15-62 *Output of show ip route 192.168.1.0 255.255.255.0 on R3*

```
R3#show ip route 192.168.1.0 255.255.255.0
Routing entry for 192.168.1.0/24
  Known via "static", distance 1, metric 0
  Routing Descriptor Blocks:
    * 10.1.23.2
      Route metric is 0, traffic share count is 1
```

The command **show run | include ip route**, as shown in Example 15-63, confirms that a static route exists.

Example 15-63 *Output of show run | include ip route*

```
R3#show run | include ip route
ip route 0.0.0.0 0.0.0.0 203.0.113.1
ip route 192.168.1.0 255.255.255.0 10.1.23.2
```

After you remove this command from R3 with the **no ip route 192.168.1.0 255.255.255.0 10.1.23.2** command, pinging from the PC is successful, as shown in Example 15-64.

Example 15-64 *A Successful Ping to the 192.168.1.0/24 Network*

```
C:\>ping 192.168.1.10

Pinging 192.168.1.10 with 32 bytes of data:

Reply from 192.168.1.10: bytes=32 time 1ms TTL=128
Reply from 192.168.1.10: bytes=32 time 1ms TTL=128
Reply from 192.168.1.10: bytes=32 time 1ms TTL=128
Reply from 192.168.1.10: bytes=32 time 1ms TTL=128

Ping statistics for 192.168.1.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Trouble Ticket 15-3

Problem: Routers R1 and R2 are not forming a neighbor adjacency.

The first item on the list for troubleshooting is to verify the problem. You access R1 and issue the **show ip ospf neighbor** command, as shown in Example 15-65, and it confirms that there is no neighbor relationship with R2.

Example 15-65 *Verifying R1's OSPF Neighbors*

```
R1#show ip ospf neighbor
R1#
```

We know that to have a neighbor relationship we need interfaces participating in the OSPF process. Using **show cdp neighbors** confirms that R2 is connected to R1's local Gig1/0 interface, as shown in Example 15-66. Therefore, we need to enable OSPF on that interface.

Example 15-66 *Verifying R1's CDP Neighbors*

```
R1#show cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone,
                  D - Remote, C - CVTA, M - Two-port Mac Relay

Device ID    Local Intrfce    Holdtme    Capability    Platform    Port ID
R2           Gig 1/0         142        R             7206VXR     Gig 0/0
```

The output of **show ip ospf interface brief** confirms that Gig1/0 is participating in the OSPF process as shown in Example 15-67. However, based on Figure 15-12, it is not in the correct area. It should be in area 1.

Example 15-67 *Verifying R1's OSPF-Enabled Interfaces*

```
R1#show ip ospf interface brief
```

Interface	PID	Area	IP Address/Mask	Cost	State	Nbrs	F/C
Gi0/0	1	1	10.1.1.1/24	1	DR	0/0	
Gi1/0	1	51	10.1.12.1/24	1	DR	0/0	

Based on Example 15-67, Gig1/0 has an IP address of 10.1.12.1/24. Therefore, we need a network command that includes that IP address and places the interface in area 1. The output of **show run | section router ospf** indicates that there is a **network** command that will enable the routing process on Gig1/0 and put it in area 1, as shown in Example 15-68.

Example 15-68 *Verifying R1's OSPF Configuration*

```
R1#show run | section router ospf
router ospf 1
  router-id 1.1.1.1
  area 1 authentication message-digest
  passive-interface default
  no passive-interface GigabitEthernet1/0
  network 10.1.1.1 0.0.0.0 area 1
  network 10.1.12.1 0.0.0.0 area 1
```

If you are scratching your head, you're not the only one at this point. The running configuration clearly shows a command that puts Gig1/0 in area 1 yet the output of **show ip interface brief** clearly shows that it is in area 51. If you have not figured out why this happened, keep reading.

Recall that there are two ways to enable OSPF on an interface: 1) with the **network area** command in router OSPF configuration mode; and 2) with the **ip ospf area** interface configuration mode command.

The **ip ospf area** command overrides the **network area** command if both are configured. Let's look at the Gig1/0 interface configuration on R1 using the **show run interface gig 1/0** command, as shown in Example 15-69.

Example 15-69 *Verifying R1's Gig1/0 Configuration*

```
R1#show run interface gigabitEthernet 1/0
Building configuration...

Current configuration : 183 bytes
!
interface GigabitEthernet1/0
  ip address 10.1.12.1 255.255.255.0
  ip ospf authentication-key CISCO
  ip ospf message-digest-key 1 md5 CISCO
  ip ospf 1 area 51
  negotiation auto
end
```

There is the issue. The **ip ospf 1 area 51** command overrides the **network 10.1.12.1 0.0.0.0 area 1** command. You will either need to change the **ip ospf 1 area 51** command so that it states area 1 or remove it completely so that the **network** command can be used.

Troubleshooting OSPFv3 for IPv6

Because OSPFv3 is based on OSPFv2, you will be dealing with similar issues when it comes to troubleshooting, with a few minor differences based on IPv6. This should come as a relief, knowing that you do not have to learn a large amount of new information for OSPFv3. However, you do need to know the **show** commands that will display the information you need to troubleshoot any given OSPFv3-related issue.

This section describes **show** commands that you can use to troubleshoot OSPFv3 neighbor adjacency issues and route issues.

OSPFv3 Troubleshooting Commands

The **show ipv6 protocols** command as shown in Example 15-70 is used to verify which IPv6 routing protocols are running on your device. Specific to OSPFv3, you can verify the process ID (PID), the RID, the type of router: Area Border Router (ABR), Autonomous System Border Router (ASBR), the number of areas the router is a member of, whether any of the areas are stub or NSSA, the interfaces participating in the routing process and the area they belong to, and whether redistribution is occurring.

Key Topic

Example 15-70 *Identifying What Can Be Verified for OSPFv3 with show ipv6 protocols*

```
R2#show ipv6 protocols
...output omitted...
IPv6 Routing Protocol is "ospf 1"
  Router ID 2.2.2.2
  Area border and autonomous system boundary router
  Number of areas: 2 normal, 0 stub, 0 nssa
  Interfaces (Area 0):
    GigabitEthernet0/0
  Interfaces (Area 23):
    GigabitEthernet1/0
  Redistribution:
    None
```

The **show ipv6 ospf** command, as shown in Example 15-71, is used to display global OSPFv3 settings. For example, you can verify the OSPFv3 PID, the RID, the type of router: ABR, ASBR, various timers and statistics, the number of areas on the router and the type of area including normal, stub and NSSA, the reference bandwidth, and the parameters related to the different areas configured on the router (for example, if area authentication is enabled, if the area is a stub, totally stubby, NSSA, or totally NSSA).

Key Topic
Example 15-71 *Identifying What Can Be Verified with show ipv6 ospf*

```

R1#show ipv6 ospf
Routing Process "ospfv3 1" with ID 1.1.1.1
Supports NSSA (compatible with RFC 3101)
Event-log enabled, Maximum number of events: 1000, Mode: cyclic
It is an area border router
Router is not originating router-LSAs with maximum metric
Initial SPF schedule delay 5000 msecs
Minimum hold time between two consecutive SPF's 10000 msecs
Maximum wait time between two consecutive SPF's 10000 msecs
Minimum LSA interval 5 secs
Minimum LSA arrival 1000 msecs
LSA group pacing timer 240 secs
Interface flood pacing timer 33 msecs
Retransmission pacing timer 66 msecs
Retransmission limit dc 24 non-dc 24
Number of external LSA 1. Checksum Sum 0x009871
Number of areas in this router is 2. 1 normal 1 stub 0 nssa
Graceful restart helper support enabled
Reference bandwidth unit is 100 mbps
RFC1583 compatibility enabled

Area BACKBONE(0)
  Number of interfaces in this area is 2
  MD5 Authentication, SPI 257
  SPF algorithm executed 3 times
  Number of LSA 11. Checksum Sum 0x06DB20
  Number of DCbitless LSA 0
  Number of indication LSA 0
  Number of DoNotAge LSA 0
  Flood list length 0

Area 1
  Number of interfaces in this area is 1
  It is a stub area, no summary LSA in this area
  Generates stub default route with cost 1
  SPF algorithm executed 4 times
  Number of LSA 7. Checksum Sum 0x03A033
  Number of DCbitless LSA 0
  Number of indication LSA 0
  Number of DoNotAge LSA 0
  Flood list length 0

```


Key Topic

The command **show ipv6 ospf interface brief**, as shown in Example 15-72, enables you to verify which interfaces are participating in the OSPFv3 process. You can also identify the PID they are attached to, the area they are participating in, the IPv6 interface ID used to represent the interface, the cost of the interface (which by default is based on the reference bandwidth divided by the interface bandwidth), the DR/BDR state, and whether

there are any neighbor adjacencies established out the interface. Notice that R1 has interfaces in area 0 and area 1. Therefore, it is an ABR.

Example 15-72 *Identifying What Can Be Verified with show ipv6 ospf interface brief*

```
R1#show ipv6 ospf interface brief
```

Interface	PID	Area	Intf ID	Cost	State	Nbrs	F/C
Gi1/0	1	0	4	1	BDR	1/1	
Gi0/0	1	0	3	1	DR	0/0	
Fa3/0	1	1	6	1	BDR	1/1	



With the `show ipv6 ospf interface interface_type interface_number` command, you can obtain detailed information about the interfaces participating in the OSPF process, as shown in Example 15-73. The unique information that will draw you to this command for troubleshooting includes the network type, the cost, whether authentication is enabled on the interface, the current DR/BDR state, the interface priority, the DR and BDR IDs, and the timers (hello and dead).

Example 15-73 *Identifying What Can Be Verified with show ipv6 ospf interface interface_type interface_number*

```
R1#show ipv6 ospf interface fastEthernet 3/0
FastEthernet3/0 is up, line protocol is up

Link Local Address FE80::C809:13FF:FEB8:54, Interface ID 6
Area 1, Process ID 1, Instance ID 0, Router ID 1.1.1.1
Network Type BROADCAST, Cost: 1
MD5 authentication SPI 256, secure socket UP (errors: 0)
Transmit Delay is 1 sec, State BDR, Priority 1
Designated Router (ID) 4.4.4.4, local address FE80::C808:9FF:FE30:1C
Backup Designated router (ID) 1.1.1.1, local address FE80::C809:13FF:FEB8:54
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
  Hello due in 00:00:04
Graceful restart helper support enabled
Index 1/1/1, flood queue length 0
Next 0x0(0)/0x0(0)/0x0(0)
Last flood scan length is 1, maximum is 2
Last flood scan time is 0 msec, maximum is 0 msec
Neighbor Count is 1, Adjacent neighbor count is 1
  Adjacent with neighbor 4.4.4.4 (Designated Router)
Suppress hello for 0 neighbor(s)
```

The `show ipv6 ospf neighbor` command enables you to verify the routers that have successfully formed a neighbor adjacency with the local router, as shown in Example 15-74. You can verify the neighbor by its RID, which is displayed in the Neighbor ID column, the priority of the neighbor's interface used to form the neighbor adjacency, the state of the neighbor's interface, the dead timer, the IPv6 interface ID of the neighboring device, and the local interface used to form the adjacency.

Example 15-74 *Identifying What Can Be Verified with show ipv6 ospf neighbor*

```
R1#show ipv6 ospf neighbor
```

```
OSPFv3 Router with ID (1.1.1.1) (Process ID 1)
```

Neighbor ID	Pri	State	Dead Time	Interface ID	Interface
2.2.2.2	1	FULL/DR	00:00:36	3	GigabitEthernet1/0
4.4.4.4	1	FULL/DR	00:00:39	4	FastEthernet3/0

To verify the LSAs that have been collected and placed in the LSDB, you use the **show ipv6 ospf database** command, as shown in Example 15-75. In this example, R1 has information for area 0 and area 1 because it is an ABR.

Example 15-75 *Displaying the OSPFv3 LSDB*

```
R1#show ipv6 ospf database
```

```
OSPFv3 Router with ID (1.1.1.1) (Process ID 1)
```

```
Router Link States (Area 0)
```

ADV Router	Age	Seq#	Fragment ID	Link count	Bits
1.1.1.1	847	0x80000005	0	1	B
2.2.2.2	748	0x80000007	0	1	B E

```
Net Link States (Area 0)
```

ADV Router	Age	Seq#	Link ID	Rtr count
2.2.2.2	878	0x80000003	3	2

```
Inter Area Prefix Link States (Area 0)
```

ADV Router	Age	Seq#	Prefix
1.1.1.1	1136	0x80000001	2001:DB8:0:14::/64
2.2.2.2	1006	0x80000002	2001:DB8:0:23::/64
2.2.2.2	1006	0x80000002	2001:DB8:0:3::/64

```
Link (Type-8) Link States (Area 0)
```

ADV Router	Age	Seq#	Link ID	Interface
1.1.1.1	847	0x80000002	4	Gil/0
2.2.2.2	1006	0x80000002	3	Gil/0
1.1.1.1	847	0x80000002	3	Gi0/0

```
Intra Area Prefix Link States (Area 0)
```

ADV Router	Age	Seq#	Link ID	Ref-lsttype	Ref-LSID
1.1.1.1	847	0x80000006	0	0x2001	0
2.2.2.2	878	0x80000003	3072	0x2002	3
Router Link States (Area 1)					
ADV Router	Age	Seq#	Fragment ID	Link count	Bits
1.1.1.1	1151	0x80000004	0	1	B
4.4.4.4	1152	0x80000006	0	1	None
Net Link States (Area 1)					
ADV Router	Age	Seq#	Link ID	Rtr count	
4.4.4.4	1147	0x80000003	4	2	
Inter Area Prefix Link States (Area 1)					
ADV Router	Age	Seq#	Prefix		
1.1.1.1	847	0x80000002	::/0		
Link (Type-8) Link States (Area 1)					
ADV Router	Age	Seq#	Link ID	Interface	
1.1.1.1	1105	0x80000002	6	Fa3/0	
4.4.4.4	1158	0x80000003	4	Fa3/0	
Intra Area Prefix Link States (Area 1)					
ADV Router	Age	Seq#	Link ID	Ref-lsttype	Ref-LSID
4.4.4.4	1147	0x80000003	4096	0x2002	4
Type-5 AS External Link States					
ADV Router	Age	Seq#	Prefix		
2.2.2.2	748	0x80000002	::/0		

Notice in Example 15-75 that there are two new LSA types when compared to Table 15-4, the Link (Type 8) LSA and the Intra Area Prefix LSA (which is also known as Type 9). Table 15-5 defines both of these LSAs for OSPFv3. Also notice in Example 15-75 that the Type 3 LSA (Summary LSA) is now called the Inter Area Prefix LSA.

Table 15-5 *Additional OSPF LSAs for OSPFv3*

LSA Type	Description
8	This LSA type (Link) provides information to neighbors about link-local addresses and the IPv6 addresses associated with the link. Therefore, it is only flooded on the local link and will not be reflooded by other OSPF routers.
9	This LSA type (Intra Area Prefix) provides information for two different scenarios. 1) It will provide information about IPv6 address prefixes associated with a transit network by referencing a Network LSA. 2) It will provide information about IPv6 address prefixes associated with a router by referencing a Router LSA. Type 9 LSAs are only flooded within an area.

To verify the OSPFv3 routes that have been installed in the routing table, you use the **show ipv6 route ospf** command, as shown in Example 15-76. In this case, R1 only knows about an external OSPFv3 route, which is the default route, and two interarea routes (routes outside the area but still within the OSPFv3 domain).

Example 15-76 *Displaying the OSPFv3 Routes in the Routing Table*

```
R1#show ipv6 route ospf
IPv6 Routing Table - default - 10 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
        B - BGP, R - RIP, H - NHRP, I1 - ISIS L1
        I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary, D - EIGRP
        EX - EIGRP external, ND - ND Default, NDp - ND Prefix, DCE - Destination
        NDr - Redirect, O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1
        OE2 - OSPF ext 2, ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2, l - LISP
OE2 ::/0 [110/1], tag 1
    via FE80::C80A:13FF:FEB8:8, GigabitEthernet1/0
OI  2001:DB8:0:3::/64 [110/3]
    via FE80::C80A:13FF:FEB8:8, GigabitEthernet1/0
OI  2001:DB8:0:23::/64 [110/2]
    via FE80::C80A:13FF:FEB8:8, GigabitEthernet1/0
```

Use the **show ipv6 interface interface_type interface_id** command, as shown in Example 15-77, when troubleshooting OSPFv3 issues to verify whether the interface is listening to the multicast group addresses of FF02::5 (all OSPFv3 routers) and FF02::6 (OSPFv3 DR/BDR). You can also verify the MTU and whether there are any IPv6 ACLs applied to the interface that might be blocking OSPFv3 packets, or packets sourced from/destined to link-local addresses.

Example 15-77 *Displaying the IPv6 Interface Parameters*

```
R1#show ipv6 interface fastEthernet 3/0
FastEthernet3/0 is up, line protocol is up
    IPv6 is enabled, link-local address is FE80::C809:13FF:FEB8:54
    ...output omitted...
```

```

Joined group address(es) :
  FF02::1
  FF02::2
  FF02::5
  FF02::6
  FF02::1:FF00:1
  FF02::1:FFB8:54
MTU is 1500 bytes
ICMP error messages limited to one every 100 milliseconds
ICMP redirects are enabled
ICMP unreachable are sent
Input features: Access List IPsec
Output features: IPsec
Inbound access list TSHOOT_ACL
ND DAD is enabled, number of DAD attempts: 1
...output omitted...

```

OSPFv3 Trouble Tickets

This section presents various trouble tickets relating to the topics discussed earlier in the chapter. The purpose of these trouble tickets is to give a process that you can follow when troubleshooting in the real world or in an exam environment. All trouble tickets in this section are based on the topology depicted in Figure 15-13.

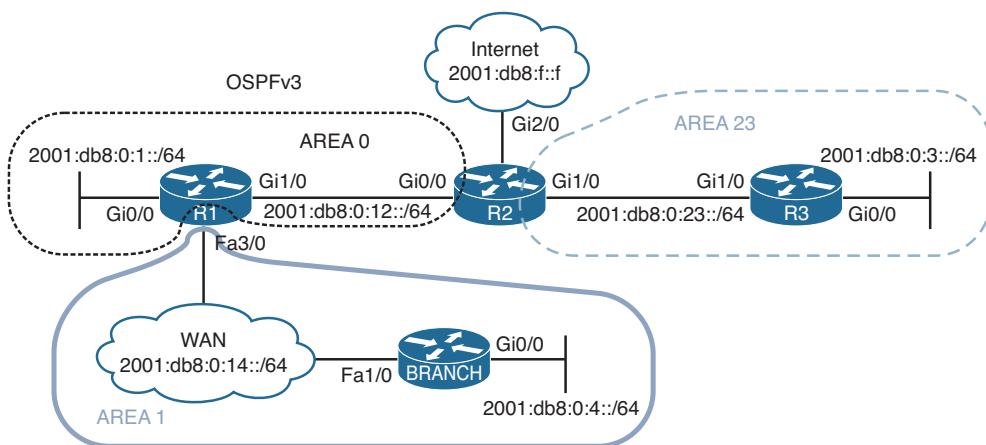


Figure 15-13 OSPFv3 Trouble Tickets Topology

Trouble Ticket 15-4

Problem: Recently, the network was updated to reduce the number of LSAs that would cross the WAN link from R1 to the Branch site. The only LSA that would be permitted is a Type 3 LSA about a default route. However, reports indicate that there are more Type 3 LSAs that are being sent from R1 to Branch.

You begin by reviewing the configuration change documents that were created when the change was implemented. You notice that the information is very vague. It only states that Area 1 was created as a totally stubby area. It does not indicate what changes were made to which devices and the commands that were used.

Your troubleshooting begins by verifying the problem with the **show ipv6 route ospf** command on Branch, as shown in Example 15-78. You confirm that there are more inter-area routes than just the default interarea route.

Example 15-78 *Displaying the IPv6 Routing Table on Branch*

```
Branch#show ipv6 route ospf
IPv6 Routing Table - default - 10 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
        B - BGP, R - RIP, H - NHRP, I1 - ISIS L1
        I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary, D - EIGRP
        EX - EIGRP external, ND - ND Default, NDp - ND Prefix, DCE - Destination
        NDr - Redirect, O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1
        OE2 - OSPF ext 2, ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2, l - LISP
OI ::/0 [110/2]
    via FE80::C801:10FF:FE20:54, FastEthernet1/0
OI 2001:DB8:0:1::/64 [110/2]
    via FE80::C801:10FF:FE20:54, FastEthernet1/0
OI 2001:DB8:0:3::/64 [110/4]
    via FE80::C801:10FF:FE20:54, FastEthernet1/0
OI 2001:DB8:0:12::/64 [110/2]
    via FE80::C801:10FF:FE20:54, FastEthernet1/0
OI 2001:DB8:0:23::/64 [110/3]
    via FE80::C801:10FF:FE20:54, FastEthernet1/0
```

Next you want to confirm if Branch is configured as a stub for area 1. You issue the command **show ipv6 ospf | include Area1stub** as shown in Example 15-79 and confirm that it is.

Example 15-79 *Verifying Whether Area 1 Is a Stub Area on Branch*

```
Branch#show ipv6 ospf | include Area1stub
Number of areas in this router is 1. 0 normal 1 stub 0 nssa
Area 1
    It is a stub area
```

You then issue the same command on R1, as shown in Example 15-80. The output indicates that area 1 is a stub area and that a default route is being injected into the area with a cost of 1.

Example 15-80 *Verifying Whether Area 1 Is a Stub Area on R1*

```
R1#show ipv6 ospf | include Area|stub
Number of areas in this router is 2. 1 normal 1 stub 0 nssa
    Area BACKBONE(0)
    Area 1
        It is a stub area
        Generates stub default route with cost 1
```

However, you realize that this output indicates that a stub area exists, not a totally stubby area. If it were a totally stubby area, it would also state *no summary LSA in this area*. To confirm this, you issue the command `show run | section ipv6 router ospf` on both R1 and Branch, as shown in Example 15-81. Reviewing the output, you notice that R1 is configured with `area 1 stub` and Branch is configured with `area 1 stub no-summary`. It appears that the commands were executed on the wrong routers.

Example 15-81 *Verifying IPv6 Router OSPF Configuration on R1 and Branch*

```
R1#show run | section ipv6 router ospf
ipv6 router ospf 1
router-id 1.1.1.1
area 1 stub
passive-interface GigabitEthernet0/0

Branch#show run | section ipv6 router ospf
ipv6 router ospf 1
router-id 4.4.4.4
area 1 stub no-summary
passive-interface default
no passive-interface FastEthernet1/0
```

To fix this issue, you issue the command `area 1 stub no-summary` on R1 and the commands `no area 1 stub no-summary` and `area 1 stub` on Branch. Once the change has been made, you issue the command `show run | section ipv6 router ospf` on both R1 and Branch to confirm the changes were made, as shown in Example 15-82.

Example 15-82 *Verifying IPv6 Router OSPF Configuration on R1 and Branch After Change*

```
R1#show run | section ipv6 router ospf
ipv6 router ospf 1
router-id 1.1.1.1
area 1 stub no-summary
passive-interface GigabitEthernet0/0

Branch#show run | section ipv6 router ospf
ipv6 router ospf 1
router-id 4.4.4.4
```

```
area 1 stub
passive-interface default
no passive-interface FastEthernet1/0
```

Next you issue the command **show ipv6 ospf | include Area1stub** on R1, as shown in Example 15-83, to verify that it states *no summary LSA in this area*, which means no Type 3. It does!

Example 15-83 Verifying Area 1 Is a Stub Area With No Summary LSAs On R1

```
R1#show ipv6 ospf | include Area1stub
Number of areas in this router is 2. 1 normal 1 stub 0 nssa
Area BACKBONE(0)
Area 1
It is a stub area, no summary LSA in this area
Generates stub default route with cost 1
```

The output of **show ipv6 route ospf** on Branch only contains the default route now. The issue is solved, as shown in Example 15-84.

Example 15-84 Verifying Branch Is Only Receiving a Default Route

```
Branch#show ipv6 route ospf
IPv6 Routing Table - default - 6 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
B - BGP, R - RIP, H - NHRP, I1 - ISIS L1
I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary, D - EIGRP
EX - EIGRP external, ND - ND Default, NDp - ND Prefix, DCE - Destination
Ndr - Redirect, O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1
OE2 - OSPF ext 2, ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2, l - LISP
OI ::/0 [110/2]
via FE80::C801:10FF:FE20:54, FastEthernet1/0
```

Trouble Ticket 15-5

Problem: Branch users are complaining that they are unable to access any resources outside the Branch office.

You access Branch and issue the extended **ping** command as shown in Example 15-85 to test connectivity and connectivity fails.

Example 15-85 Testing Connectivity from Branch to a Remote Network

```
Branch#ping
Protocol [ip]: ipv6
Target IPv6 address: 2001:db8:0:1::1
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
```

```

Extended commands? [no]: yes
Source address or interface: 2001:db8:0:4::4
UDP protocol? [no]:
Verbose? [no]:
Precedence [0]:
DSCP [0]:
Include hop by hop option? [no]:
Include destination option? [no]:
Sweep range of sizes? [no]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:0:1::1, timeout is 2 seconds:
Packet sent with a source address of 2001:DB8:0:4::4
.....
Success rate is 0 percent (0/5)

```

You issue the **show ipv6 route** command on Branch and notice that there are only local and connected routes, as shown in Example 15-86.

Example 15-86 Verifying IPv6 Routes in a Routing Table

```

Branch#show ipv6 route
...output omitted...
C   2001:DB8:0:4::/64 [0/0]
    via GigabitEthernet0/0, directly connected
L   2001:DB8:0:4::4/128 [0/0]
    via GigabitEthernet0/0, receive
C   2001:DB8:0:14::/64 [0/0]
    via FastEthernet1/0, directly connected
L   2001:DB8:0:14::4/128 [0/0]
    via FastEthernet1/0, receive
L   FF00::/8 [0/0]
    via Null0, receive

```

You conclude that no routes are being learned from R1. Therefore, there must be a neighbor issue. To confirm, you issue the command **show ipv6 ospf neighbor** on Branch, and as you suspected, Example 15-87 confirms that Branch is not a neighbor with R1.

Example 15-87 Verifying IPv6 OSPF Neighbors

```

Branch#show ipv6 ospf neighbor
Branch#

```

You suspect that the Branch interface connected to R1 is not enabled for the OSPFv3 process. You issue the **show ipv6 ospf interface brief** command to verify whether the interface is participating in the process. In Example 15-88, the output indicates that Fast Ethernet 1/0 is participating in the OSPFv3 process.

Example 15-88 *Verifying OSPFv3-Enabled Interfaces on Branch*

```
Branch#show ipv6 ospf interface brief
```

Interface	PID	Area	Intf ID	Cost	State	Nbrs	F/C
Gi0/0	1	1	3	1	DR	0/0	
Fa1/0	1	1	4	1	BDR	1/1	

You decide to shift your attention to R1 and check whether the interface connected to Branch is participating in the OSPFv3 process. R1 is using Fast Ethernet 3/0 to connect to Branch. Issuing the command **show ipv6 ospf interface brief** on R1, as shown in Example 15-89, reveals that Fa3/0 is participating in the OSPF process as well.

Example 15-89 *Verifying OSPFv3-Enabled Interfaces on R1*

```
R1#show ipv6 ospf interface brief
```

Interface	PID	Area	Intf ID	Cost	State	Nbrs	F/C
Gi1/0	1	0	4	1	BDR	1/1	
Gi0/0	1	0	3	1	DR	0/0	
Fa3/0	1	1	6	1	DR	0/0	

You revisit Branch and decide to issue the **debug ipv6 ospf hello** command to gather further information. The output displayed in Example 15-90 reveals that timers are mismatched from FE80::C801:10FF:FE20:54. You issue the **show cdp neighbors detail** command on Branch, as shown in Example 15-91, to confirm that R1 is using that link-local address. It is! Therefore, you conclude that the neighbor relationship is not formed because of mismatched timers.

Example 15-90 *Using debug ipv6 ospf hello to Gather Further Information*

```
Branch#debug ipv6 ospf hello
OSPFv3 hello events debugging is on for process 1, IPv6, Default vrf
Branch#
OSPFv3-1-IPv6 HELLO Fa1/0: Rcv hello from 1.1.1.1 area 1 from FE80::C801:10FF:FE20:54
interface ID 6
OSPFv3-1-IPv6 HELLO Fa1/0: Mismatched hello parameters from FE80::C801:10FF:FE20:54
OSPFv3-1-IPv6 HELLO Fa1/0: Dead R 40 C 120, Hello R 10 C 30
Branch#u all
All possible debugging has been turned off
```

Example 15-91 *Using show cdp neighbors details to Verify Neighbor IPv6 Address*

```
Branch#show cdp neighbors detail
-----
Device ID: R1
Entry address(es):
  IP address: 10.1.14.1
  IPv6 address: 2001:DB8:0:14::1 (global unicast)
  IPv6 address: FE80::C801:10FF:FE20:54 (link-local)
Platform: Cisco 7206VXR, Capabilities: Router
```

```
Interface: FastEthernet1/0, Port ID (outgoing port): FastEthernet3/0
...output omitted...
```

On R1, you issue the **show ipv6 ospf interface fastethernet3/0** command, and on Branch you issue the **show ipv6 ospf interface fastethernet1/0** command and use the spot-the-difference method, as shown in Example 15-92.

Example 15-92 *Spotting the Difference Between R1 and Branch*

```
R1#show ipv6 ospf interface fastEthernet 3/0
FastEthernet3/0 is up, line protocol is up
  Link Local Address FE80::C801:10FF:FE20:54, Interface ID 6
  Area 1, Process ID 1, Instance ID 0, Router ID 1.1.1.1
  Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 1.1.1.1, local address FE80::C801:10FF:FE20:54
  No backup designated router on this network
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:09
...output omitted...

Branch#show ipv6 ospf interface fastEthernet 1/0
FastEthernet1/0 is up, line protocol is up
  Link Local Address FE80::C800:FFF:FE7C:1C, Interface ID 4
  Area 1, Process ID 1, Instance ID 0, Router ID 4.4.4.4
  Network Type NON_BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 4.4.4.4, local address FE80::C800:FFF:FE7C:1C
  No backup designated router on this network
  Timer intervals configured, Hello 30, Dead 120, Wait 120, Retransmit 5
    Hello due in 00:00:25
...output omitted...
```

You immediately notice that the hello and dead timers do not match. However, you remember that they can be configured manually or manipulated by changing the OSPF interface network type. Therefore, you check the network type and R1 is using BROADCAST (default for Ethernet interfaces), and Branch is using NON_BROADCAST (not the default for Ethernet interfaces). Therefore, someone must have manually changed the network type on Branch.

You issue the command **show run interface fastethernet 1/0** on Branch, as shown in Example 15-93, and confirm that the network type was manually changed with the **ipv6 ospf network non-broadcast** command.

Example 15-93 *Verifying the Interface Configuration on Branch*

```
Branch#show run interface FastEthernet 1/0
Building configuration...

Current configuration : 169 bytes
!
interface FastEthernet1/0
 ip address 10.1.14.4 255.255.255.0
 duplex full
 ipv6 address 2001:DB8:0:14::4/64
 ipv6 ospf 1 area 1
 ipv6 ospf network non-broadcast
end
```

You remove this command with the **no ipv6 ospf network non-broadcast** command, which will change the network type back to the default of BROADCAST. Once that happens, a syslog message is generated indicating that a neighbor relationship is successfully formed between R1 and Branch:

%OSPFv3-5-ADJCHG: Process 1, Nbr 1.1.1.1 on FastEthernet1/0 from LOADING to FULL, Loading Done

You reissue the extended **ping** command on Branch, and it is successful, as shown in Example 15-94.

Example 15-94 *Testing Connectivity from Branch to a Remote Network*

```
Branch#ping
Protocol [ip]: ipv6
Target IPv6 address: 2001:db8:0:1::1
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands? [no]: yes
Source address or interface: 2001:db8:0:4::4
UDP protocol? [no]:
Verbose? [no]:
Precedence [0]:
DSCP [0]:
Include hop by hop option? [no]:
Include destination option? [no]:
Sweep range of sizes? [no]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:0:1::1, timeout is 2 seconds:
Packet sent with a source address of 2001:DB8:0:4::4
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 16/29/52 ms
```

Troubleshoot OSPFv3 Address Families

OSPFv3 address families (AFs) enable you to configure a single process that will support both IPv4 and IPv6. In addition, a single database is maintained for IPv4 and IPv6. However, adjacencies are established individually for each AF, and settings can be configured on an AF-by-AF basis.

In this section, you learn the commands that you can use to troubleshoot an OSPFv3 implementation that uses address families.

OSPFv3 Address Family Troubleshooting

Example 15-95 shows a sample OSPFv3 configuration with AFs. The OSPFv3 PID is 10 and is locally significant. Therefore, it does not have to match between neighbors. Any parameter configured under the main router OSPFv3 configuration mode will apply to all address families. In this example, the area 23 stub command was configured under the main router OSPFv3 configuration mode; therefore, area 23 will be a stub area for both IPv4 and IPv6 address families. Note that if there are conflicts between configurations in router OSPFv3 configuration mode and AF configuration mode, AF configuration mode wins. You still enable the OSPFv3 process on an interface-by-interface basis in interface configuration mode with the `ospfv3 process_id {ipv4|ipv6} area area_id` command. In addition, OSPFv3 interface parameters are still configured in interface configuration mode. However, remember that if you do not specify the AF (IPv4 or IPv6), the configured parameter applies to all address families. If you apply the configuration to the AF, it applies only to that AF. If a conflict exists, the AF configuration wins. Refer to the Gigabit Ethernet 0/0 configuration in Example 15-95. Notice that the hello interval is configured without an AF specified. Therefore, it applies to both IPv4 and IPv6. However, the hello interval is also configured for the IPv6 AF. Therefore, this configuration prevails for IPv6, and a hello interval of 10 is used; IPv4 uses the hello interval of 11.



Example 15-95 Sample OSPFv3 Configuration with Address Families

```
R2#show run | section router ospfv3
router ospfv3 10
  area 23 stub
  !
  address-family ipv4 unicast
    passive-interface default
    no passive-interface GigabitEthernet0/0
    no passive-interface GigabitEthernet1/0
    default-information originate
    router-id 2.2.2.2
  exit-address-family
  !
  address-family ipv6 unicast
    passive-interface default
    no passive-interface GigabitEthernet0/0
```

```

no passive-interface GigabitEthernet1/0
default-information originate
router-id 22.22.22.22
exit-address-family

```

```

R2#show run int gig 1/0
interface GigabitEthernet1/0
 ip address 10.1.23.2 255.255.255.0
 ipv6 address 2001:DB8:0:23::2/64
 ospfv3 10 ipv6 area 23
 ospfv3 10 ipv4 area 23
end

```

```

R2#show run int gig 0/0
interface GigabitEthernet0/0
 ip address 10.1.12.2 255.255.255.0
 ipv6 address 2001:DB8:0:12::2/64
 ospfv3 10 hello-interval 11
 ospfv3 10 ipv6 area 0
 ospfv3 10 ipv6 hello-interval 10
 ospfv3 10 ipv4 area 0
end

```

With OSPFv3 AFs, you can still use the **show ip protocols** and **show ipv6 protocols** commands, as shown in Example 15-96, to verify the same information previously discussed in the chapter.

Example 15-96 Using show ip protocols and show ipv6 protocols

```

R2#show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "ospfv3 10"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 2.2.2.2
  Area border and autonomous system boundary router
  Number of areas: 1 normal, 1 stub, 0 nssa
  Interfaces (Area 0):
    GigabitEthernet0/0
  Interfaces (Area 23):
    GigabitEthernet1/0
  Maximum path: 4
  Routing Information Sources:
    Gateway         Distance      Last Update
    2.2.2.2          110          00:12:39
    3.3.3.3          110          00:12:39

```

```

10.1.14.1          110      00:00:57
Distance: (default is 110)

R2#show ipv6 protocols
IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "ND"
IPv6 Routing Protocol is "static"
IPv6 Routing Protocol is "ospf 10"
  Router ID 22.22.22.22
  Area border and autonomous system boundary router
  Number of areas: 1 normal, 1 stub, 0 nssa
  Interfaces (Area 0):
    GigabitEthernet0/0
  Interfaces (Area 23):
    GigabitEthernet1/0
  Redistribution:
    None

```

The output of **show ospfv3**, as shown in Example 15-97, displays the same information you would find with the **show ip ospf** and **show ipv6 ospf** commands. Notice that the IPv4 AF is listed first followed by the IPv6 AF.



Example 15-97 *Using show ospfv3 to Verify General OSPFv3 Parameters for AFs*

```

R2#show ospfv3
  OSPFv3 10 address-family ipv4
  Router ID 2.2.2.2
  Supports NSSA (compatible with RFC 3101)
  Event-log enabled, Maximum number of events: 1000, Mode: cyclic
  It is an area border and autonomous system boundary router
  Redistributing External Routes from,
  Originate Default Route
  Router is not originating router-LSAs with maximum metric
  Initial SPF schedule delay 5000 msecs
  Minimum hold time between two consecutive SPF's 10000 msecs
  Maximum wait time between two consecutive SPF's 10000 msecs
  Minimum LSA interval 5 secs
  Minimum LSA arrival 1000 msecs
  LSA group pacing timer 240 secs
  Interface flood pacing timer 33 msecs
  Retransmission pacing timer 66 msecs
  Retransmission limit dc 24 non-dc 24
  Number of external LSA 1. Checksum Sum 0x0013EB
  Number of areas in this router is 2. 1 normal 1 stub 0 nssa
  Graceful restart helper support enabled
  Reference bandwidth unit is 100 mbps
  RFC1583 compatibility enabled

```

```

Area BACKBONE(0)
  Number of interfaces in this area is 1
  SPF algorithm executed 13 times
  Number of LSA 11. Checksum Sum 0x05A71D
  Number of DCbitless LSA 0
  Number of indication LSA 0
  Number of DoNotAge LSA 0
  Flood list length 0
Area 23
  Number of interfaces in this area is 1
  It is a stub area
  Generates stub default route with cost 1
  SPF algorithm executed 8 times
  Number of LSA 12. Checksum Sum 0x064322
  Number of DCbitless LSA 0
  Number of indication LSA 0
  Number of DoNotAge LSA 0
  Flood list length 0

OSPFv3 10 address-family ipv6
Router ID 22.22.22.22
Supports NSSA (compatible with RFC 3101)
Event-log enabled, Maximum number of events: 1000, Mode: cyclic
It is an area border and autonomous system boundary router
Originate Default Route
Router is not originating router-LSAs with maximum metric
Initial SPF schedule delay 5000 msecs
Minimum hold time between two consecutive SPF's 10000 msecs
Maximum wait time between two consecutive SPF's 10000 msecs
Minimum LSA interval 5 secs
Minimum LSA arrival 1000 msecs
LSA group pacing timer 240 secs
Interface flood pacing timer 33 msecs
Retransmission pacing timer 66 msecs
Retransmission limit dc 24 non-dc 24
Number of external LSA 1. Checksum Sum 0x00B8F5
Number of areas in this router is 2. 1 normal 1 stub 0 nssa
Graceful restart helper support enabled
Reference bandwidth unit is 100 mbps
RFC1583 compatibility enabled
Area BACKBONE(0)
  Number of interfaces in this area is 1
  SPF algorithm executed 13 times
  Number of LSA 11. Checksum Sum 0x0422C7
  Number of DCbitless LSA 0
  Number of indication LSA 0

```

```

Number of DoNotAge LSA 0
Flood list length 0
Area 23
Number of interfaces in this area is 1
It is a stub area
Generates stub default route with cost 1
SPF algorithm executed 11 times
Number of LSA 12. Checksum Sum 0x0591F5
Number of DCbitless LSA 0
Number of indication LSA 0
Number of DoNotAge LSA 0
Flood list length 0

```

Using the command **show ospfv3 interface brief** command will display the interfaces participating in the OSPFv3 process for each AF, as shown in Example 15-98. Notice the added column that indicates which AF the interface is participating in.

Key Topic

Example 15-98 Using show ospfv3 interface brief to Verify OSPFv3 Interfaces

```

R2#show ospfv3 interface brief

```

Interface	PID	Area	AF	Cost	State	Nbrs	F/C
Gi0/0	10	0	ipv4	1	BDR	1/1	
Gi1/0	10	23	ipv4	1	BDR	1/1	
Gi0/0	10	0	ipv6	1	BDR	1/1	
Gi1/0	10	23	ipv6	1	BDR	1/1	

The **show ospfv3 interface** command enables you to review detailed information about the interface configurations, as shown earlier in the chapter. Example 15-99 displays the IPv4 AF information at the top and the IPv6 AF information at the bottom.

Key Topic

Example 15-99 Using show ospfv3 interface to Verify Details of OSPFv3 Interfaces

```

R2#show ospfv3 interface gigabitEthernet 1/0
GigabitEthernet1/0 is up, line protocol is up
Link Local Address FE80::C802:10FF:FE20:1C, Interface ID 4
Internet Address 10.1.23.2/24
Area 23, Process ID 10, Instance ID 64, Router ID 2.2.2.2
Network Type BROADCAST, Cost: 1
Transmit Delay is 1 sec, State BDR, Priority 1
Designated Router (ID) 3.3.3.3, local address FE80::C804:10FF:FE74:1C
Backup Designated router (ID) 2.2.2.2, local address FE80::C802:10FF:FE20:1C
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
Hello due in 00:00:02
Graceful restart helper support enabled
Index 1/1/2, flood queue length 0
Next 0x0(0)/0x0(0)/0x0(0)
Last flood scan length is 4, maximum is 5
Last flood scan time is 4 msec, maximum is 4 msec

```



```

Neighbor Count is 1, Adjacent neighbor count is 1
  Adjacent with neighbor 3.3.3.3 (Designated Router)
  Suppress hello for 0 neighbor(s)
GigabitEthernet1/0 is up, line protocol is up
  Link Local Address FE80::C802:10FF:FE20:1C, Interface ID 4
  Area 23, Process ID 10, Instance ID 0, Router ID 22.22.22.22
  Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State BDR, Priority 1
  Designated Router (ID) 33.33.33.33, local address FE80::C804:10FF:FE74:1C
  Backup Designated router (ID) 22.22.22.22, local address FE80::C802:10FF:FE20:1C
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:03
  Graceful restart helper support enabled
  Index 1/1/2, flood queue length 0
  Next 0x0(0)/0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 4
  Last flood scan time is 0 msec, maximum is 4 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 33.33.33.33 (Designated Router)
  Suppress hello for 0 neighbor(s)

```

To verify the neighbor relationships that have been formed for each AF, issue the command **show ospfv3 neighbor**, as shown in Example 15-100. Again, the output is presenting the same information as discussed earlier in the chapter, except this time there are different sections for each AF.

Example 15-100 *Using show ospfv3 neighbor to Verify OSPFv3 Neighbors*

```

R2#show ospfv3 neighbor

      OSPFv3 10 address-family ipv4 (router-id 2.2.2.2)

Neighbor ID    Pri   State           Dead Time   Interface ID  Interface
10.1.14.1      1    FULL/DR         00:00:34    4             GigabitEthernet0/0
3.3.3.3        1    FULL/DR         00:00:36    4             GigabitEthernet1/0

      OSPFv3 10 address-family ipv6 (router-id 22.22.22.22)

Neighbor ID    Pri   State           Dead Time   Interface ID  Interface
10.1.14.1      1    FULL/DR         00:00:31    4             GigabitEthernet0/0
33.33.33.33    1    FULL/DR         00:00:34    4             GigabitEthernet1/0

```

To verify the information in the LSDB, you issue the command **show ospfv3 database**. When using AFs, the OSPFv3 database contains LSAs for both IPv4 and IPv6 as shown in Example 15-101.

Example 15-101 *Verifying the LSDB with show ospfv3 database*

```
R2#show ospfv3 database
```

```
OSPFv3 10 address-family ipv4 (router-id 2.2.2.2)
```

Router Link States (Area 0)

ADV Router	Age	Seq#	Fragment ID	Link count	Bits
2.2.2.2	1456	0x80000008	0	1	B E
10.1.14.1	1457	0x80000007	0	1	B

Net Link States (Area 0)

ADV Router	Age	Seq#	Link ID	Rtr count
10.1.14.1	1453	0x80000003	4	2

Inter Area Prefix Link States (Area 0)

ADV Router	Age	Seq#	Prefix
2.2.2.2	1618	0x80000003	10.1.23.0/24
2.2.2.2	94	0x80000002	10.1.3.0/24
10.1.14.1	1599	0x80000002	10.1.14.0/24
10.1.14.1	1599	0x80000002	10.1.4.0/24

Link (Type-8) Link States (Area 0)

ADV Router	Age	Seq#	Link ID	Interface
2.2.2.2	1618	0x80000003	3	Gi0/0
10.1.14.1	1599	0x80000002	4	Gi0/0

Intra Area Prefix Link States (Area 0)

ADV Router	Age	Seq#	Link ID	Ref-lstype	Ref-LSID
10.1.14.1	1457	0x80000007	0	0x2001	0
10.1.14.1	1453	0x80000003	4096	0x2002	4

Router Link States (Area 23)

ADV Router	Age	Seq#	Fragment ID	Link count	Bits
2.2.2.2	94	0x80000007	0	1	B
3.3.3.3	248	0x80000009	0	1	None

Net Link States (Area 23)

ADV Router	Age	Seq#	Link ID	Rtr count

```
3.3.3.3      248      0x80000007  4      2
```

Inter Area Prefix Link States (Area 23)

ADV Router	Age	Seq#	Prefix
2.2.2.2	1869	0x80000002	0.0.0.0/0
2.2.2.2	1442	0x80000001	10.1.1.0/24
2.2.2.2	1442	0x80000001	10.1.12.0/24
2.2.2.2	1442	0x80000001	10.1.4.0/24
2.2.2.2	1442	0x80000001	10.1.14.0/24

Link (Type-8) Link States (Area 23)

ADV Router	Age	Seq#	Link ID	Interface
2.2.2.2	1618	0x80000004	4	Gil/0
3.3.3.3	1758	0x80000004	4	Gil/0

Intra Area Prefix Link States (Area 23)

ADV Router	Age	Seq#	Link ID	Ref-lstyp	Ref-LSID
3.3.3.3	248	0x80000008	0	0x2001	0
3.3.3.3	248	0x80000007	4096	0x2002	4

Type-5 AS External Link States

ADV Router	Age	Seq#	Prefix
2.2.2.2	1618	0x80000003	0.0.0.0/0

```
OSPFv3 10 address-family ipv6 (router-id 22.22.22.22)
```

Router Link States (Area 0)

ADV Router	Age	Seq#	Fragment ID	Link count	Bits
10.1.14.1	330	0x80000007	0	1	B
22.22.22.22	198	0x8000000A	0	1	B E

Net Link States (Area 0)

ADV Router	Age	Seq#	Link ID	Rtr count
10.1.14.1	330	0x80000004	4	2

Inter Area Prefix Link States (Area 0)

ADV Router	Age	Seq#	Prefix
10.1.14.1	1598	0x80000002	2001:DB8:0:14::/64
10.1.14.1	1598	0x80000002	2001:DB8:0:4::/64

```

22.22.22.22      198      0x80000002  2001:DB8:0:3::/64
22.22.22.22      198      0x80000002  2001:DB8:0:23::/64

```

Link (Type-8) Link States (Area 0)

ADV Router	Age	Seq#	Link ID	Interface
10.1.14.1	1598	0x80000002	4	Gi0/0
22.22.22.22	1446	0x80000003	3	Gi0/0

Intra Area Prefix Link States (Area 0)

ADV Router	Age	Seq#	Link ID	Ref-lstype	Ref-LSID
10.1.14.1	330	0x80000006	0	0x2001	0
10.1.14.1	330	0x80000004	4096	0x2002	4

Router Link States (Area 23)

ADV Router	Age	Seq#	Fragment ID	Link count	Bits
22.22.22.22	198	0x8000000A	0	1	B
33.33.33.33	237	0x80000008	0	1	None

Net Link States (Area 23)

ADV Router	Age	Seq#	Link ID	Rtr count
33.33.33.33	237	0x80000007	4	2

Inter Area Prefix Link States (Area 23)

ADV Router	Age	Seq#	Prefix
22.22.22.22	198	0x80000005	2001:DB8:0:12::/64
22.22.22.22	1961	0x80000002	::/0
22.22.22.22	198	0x80000002	2001:DB8:0:1::/64
22.22.22.22	198	0x80000002	2001:DB8:0:4::/64
22.22.22.22	198	0x80000002	2001:DB8:0:14::/64

Link (Type-8) Link States (Area 23)

ADV Router	Age	Seq#	Link ID	Interface
22.22.22.22	1446	0x80000004	4	Gi1/0
33.33.33.33	1713	0x80000004	4	Gi1/0

Intra Area Prefix Link States (Area 23)

ADV Router	Age	Seq#	Link ID	Ref-lstype	Ref-LSID
33.33.33.33	237	0x8000000A	0	0x2001	0
33.33.33.33	237	0x80000007	4096	0x2002	4

Type-5 AS External Link States			
ADV Router	Age	Seq#	Prefix
22.22.22.22	1446	0x80000003	::/0

Keep in mind when troubleshooting OSPFv3 AFs that both OSPF for IPv4 and OSPF for IPv6 use IPv6 to exchange routing information. Therefore, IPv6 unicast routing must be enabled on the router. Also, classic OSPFv2 and the OSPFv3 AFs are not compatible. Therefore, a router using OSPFv3 AFs for IPv4 will not peer with a router using the classic OSPFv2 configuration for IPv4 because they are not compatible.

To verify the IPv4 OSPFv3 entries in the routing table, you can use the **show ip route ospfv3** command. To verify the IPv6 OSPFv3 entries in the routing table, you can use the **show ipv6 route ospf** command.

If you need to perform any debugging for OSPFv3, you can issue the **debug ospfv3** command followed by what you want to debug, such as **events**, **packets**, **hellos**, or **adj**. This will turn on the debug for all AFs. If you want to only turn it on for a specific AF, you need to include the AF in the command (for example **debug ospfv3 ipv6 hello**). In the command, **ipv6** is referring to the AF.

OSPFv3 AF Trouble Tickets

This section presents various trouble tickets relating to the topics discussed earlier in the chapter. The purpose of these trouble tickets is to give a process that you can follow when troubleshooting in the real world or in an exam environment. All trouble tickets in this section are based on the topology depicted in Figure 15-14.

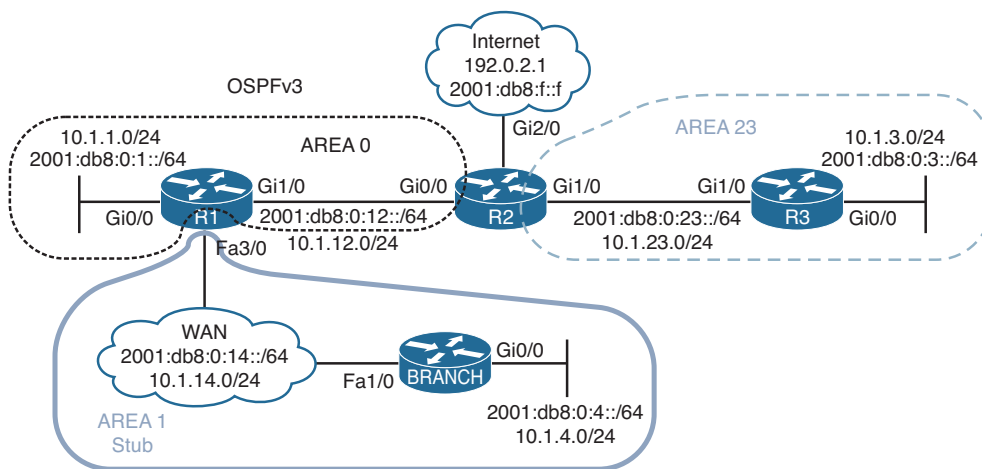


Figure 15-14 OSPFv3 AF Trouble Tickets Topology

Trouble Ticket 15-6

Problem: Users in Branch have indicated that they are not able to access any IPv6-enabled resources on the Internet but they can access IPv4-enabled resources.

An extended ping issued on Branch to the destination 2001:db8:f::f confirms the issue as shown in Example 15-102. In addition, you ping 192.0.2.1 and it is successful confirming connectivity to IPv4-enabled resources.

Example 15-102 *Verifying Connectivity*

```
Branch#ping
Protocol [ip]: ipv6
Target IPv6 address: 2001:db8:f::f
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands? [no]: yes
Source address or interface: 2001:db8:0:4::4
UDP protocol? [no]:
Verbose? [no]:
Precedence [0]:
DSCP [0]:
Include hop by hop option? [no]:
Include destination option? [no]:
Sweep range of sizes? [no]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:f::f, timeout is 2 seconds:
Packet sent with a source address of 2001:DB8:0:4::4
UUUUU
Success rate is 0 percent (0/5)

Branch#ping 192.0.2.1 source 10.1.4.4
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.0.2.1, timeout is 2 seconds:
Packet sent with a source address of 10.1.4.4
!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 80/112/152 ms
```

On the Branch router, you issue the command **show ipv6 route 2001:db8:f::f**, and Example 15-103 indicates that the Branch router has a default route that can be used to reach the IPv6 address. This explains why the ping returned UUUUU. It indicates that the destination is not reachable by some other router. But which router is returning this message?

Example 15-103 *Verifying Routes in the IPv6 Routing Table*

```
Branch#show ipv6 route 2001:db8:f::f
Routing entry for ::/0
  Known via "ospf 1", distance 110, metric 2, type inter area
  Route count is 1/1, share count 0
  Routing paths:
    FE80::C801:10FF:FE20:54, FastEthernet1/0
    Last updated 00:07:28 ago
```

To verify this, you issue a trace to see where it fails. Example 15-104 displays the results of the command **traceroute 2001:db8:f::f**. The trace indicates that R1 is returning the destination unreachable message.

Example 15-104 *Tracing the Path*

```
Branch#traceroute 2001:db8:f::f
Type escape sequence to abort.
Tracing the route to 2001:DB8:F::F

 1 2001:DB8:0:14::1 !U !U !U
```

You visit R1 and issue the **show ipv6 route 2001:db8:f::f** command, as shown in Example 15-105, and confirm that there is no route to reach that IPv6 address. Why would Branch have a default route but not R1? Reviewing the network diagram shows that area 1 is a stub area. Therefore, R1 is generating a default route and injecting it into the stub area. This is why the default route on Branch, as shown in Example 15-103, is of type interarea and not external.

Example 15-105 *Verifying Routes on R1*

```
R1#show ipv6 route 2001:db8:f::f
% Route not found
```

It seems that R2 might not be generating a default route when it should be. You access R2 and issue the **show ospfv3 ipv6** command, as shown in Example 15-106, and confirm that it is not an ASBR, when it should be if it is generating a default route.

Example 15-106 *Verifying OSPFv3 Parameters on R2*

```
R2#show ospfv3 ipv6
OSPFv3 10 address-family ipv6
Router ID 22.22.22.22
Supports NSSA (compatible with RFC 3101)
Event-log enabled, Maximum number of events: 1000, Mode: cyclic
It is an area border router
Router is not originating router-LSAs with maximum metric
Initial SPF schedule delay 5000 msecs
Minimum hold time between two consecutive SPF's 10000 msecs
```

```

Maximum wait time between two consecutive SPF's 10000 msec
Minimum LSA interval 5 secs
Minimum LSA arrival 1000 msec
LSA group pacing timer 240 secs
Interface flood pacing timer 33 msec
Retransmission pacing timer 66 msec
Retransmission limit dc 24 non-dc 24
Number of external LSA 0. Checksum Sum 0x000000
Number of areas in this router is 2. 1 normal 1 stub 0 nssa
Graceful restart helper support enabled
Reference bandwidth unit is 100 mbps
RFC1583 compatibility enabled
  Area BACKBONE(0)
    Number of interfaces in this area is 1
    SPF algorithm executed 14 times
    Number of LSA 11. Checksum Sum 0x04EDE6
    Number of DCbitless LSA 0
    Number of indication LSA 0
    Number of DoNotAge LSA 0
    Flood list length 0
  Area 23
    Number of interfaces in this area is 1
    It is a stub area
    Generates stub default route with cost 1
    SPF algorithm executed 11 times
    Number of LSA 12. Checksum Sum 0x06610D
    Number of DCbitless LSA 0
    Number of indication LSA 0
    Number of DoNotAge LSA 0
    Flood list length 0

```

Next you issue the command **show run | section router ospfv3**. The output in Example 15-107 confirms that the **default-information originate** command is missing from IPv6 AF configuration mode. It is only configured under IPv4 AF configuration mode.

Example 15-107 Verifying OSPFv3 Configuration on R2

```

R2#show run | section router ospfv3
router ospfv3 10
  area 23 stub
  !
  address-family ipv4 unicast
    passive-interface default
    no passive-interface GigabitEthernet0/0
    no passive-interface GigabitEthernet1/0
    default-information originate
  router-id 2.2.2.2

```



```

exit-address-family
!
address-family ipv6 unicast
  passive-interface default
  no passive-interface GigabitEthernet0/0
  no passive-interface GigabitEthernet1/0
  router-id 22.22.22.22
exit-address-family

```

You add the **default-information originate** command to IPv6 AF configuration mode and reissue the extended IPv6 ping on Branch, as shown in Example 15-108. The ping is successful.

Example 15-108 *Successful Ping to IPv6 Internet Resources*

```

Branch#ping
Protocol [ip]: ipv6
Target IPv6 address: 2001:db8:f::f
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands? [no]: yes
Source address or interface: 2001:db8:0:4::4
UDP protocol? [no]:
Verbose? [no]:
Precedence [0]:
DSCP [0]:
Include hop by hop option? [no]:
Include destination option? [no]:
Sweep range of sizes? [no]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:F::F, timeout is 2 seconds:
Packet sent with a source address of 2001:DB8:0:4::4
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 88/113/148 ms

```

Exam Preparation Tasks

As mentioned in the section “How to Use This Book” in the Introduction, you have a couple of choices for exam preparation: the exercises here; Chapter 22, “Final Preparation;” and the exam simulation questions on the CD-ROM.

Review All Key Topics

Review the most important topics in this chapter, noted with the Key Topic icon in the outer margin of the page. Table 15-6 lists a reference of these key topics and the page numbers on which each is found.



Table 15-6 *Key Topics for Chapter 15*

Key Topic Element	Description	Page Number
Example 15-1	Verifying OSPF Neighbors with show ip ospf neighbor	591
List	Identifies the reasons why an OSPF neighbor relationship might not form	591
Table 15-2	Describes adjacency states	592
Example 15-2	Verifying OSPF interfaces with show ip ospf interface	593
Example 15-4	Displaying OSPF interface timers on R1 Gigabit Ethernet 1/0	595
Section	Discusses how to identify mismatched OSPFv2 area numbers	596
Example 15-9	Determining the type of OSPF Areas	597
Paragraph	Discusses the passive interface feature and how to troubleshoot passive interface issues	599
Example 15-13	Verifying OSPF area authentication	600
Example 15-14	Verifying OSPF authentication key	600
Section	MTU mismatch	602
Table 15-3	OSPF network types and characteristics	604
List	Identifies the reasons why an OSPF route might be missing from either the LSDB or the routing table	606
List	Outlines what needs to be considered when troubleshooting route filtering	611
Section	Stub area configurations	613

Key Topic Element	Description	Page Number
Paragraph	Describes the importance of the DR election in a hub-and-spoke multiaccess network	615
Table 15-4	OSPFv2 LSAs	621
List	Outlines what needs to be considered when troubleshooting route summarization issues	622
Example 15-42	Verifying the Virtual Link	626
Example 15-70	Identifying what can be verified for OSPFv3 with show ipv6 protocols	641
Example 15-71	Identifying what can be verified with show ipv6 ospf	642
Paragraph	Discusses what can be verified during the troubleshooting process with the show ipv6 ospf interface brief command	642
Paragraph	Describes what can be verified during the troubleshooting process with the show ipv6 ospf interface command	643
Example 15-95	Sample OSPFv3 configuration with AFs	655
Example 15-97	Using show ospfv3 to verify general OSPFv3 parameters for AFs	657
Example 15-98	Using show ospfv3 interface brief to verify OSPFv3 interfaces	659
Example 15-99	Using show ospfv3 interface to verify details of OSPFv3 interfaces	659

Define Key Terms

Define the following key terms from this chapter and check your answers in the glossary:

OSPF interface table, OSPF neighbor table, OSPF link-state database, link-state advertisement (LSA), Dijkstra shortest path first (SPF) algorithm, OSPF area, virtual link, OSPF Area Border Router (ABR), OSPF Autonomous System Boundary Router (ASBR), OSPFv3, address families, designated router, backup designated router, stub area, totally stubby area, NSSA, totally NSSA

Complete Tables and Lists from Memory

Print a copy of Appendix C, “Memory Tables,” (found on the disc), or at least the section for this chapter, and complete the tables and lists from memory. Appendix D, “Memory Tables Answer Key,” also on the disc, includes completed tables and lists to check your work.

Command Reference to Check Your Memory

This section includes the most important **show** and **debug** commands covered in this chapter. It might not be necessary to memorize the complete syntax of every command, but you should be able to remember the basic keywords that are needed.

To test your memory of the commands, cover the right side of Table 15-7 with a piece of paper, read the description on the left side, and then see how much of the command you can remember.

The 300-135 TSHOOT exam focuses on practical, hands-on skills that are used by a networking professional. Therefore, you should be able to identify the commands needed to successfully troubleshoot the different issues outlined in this chapter.

Table 15-7 *show and debug commands*

Task	Command Syntax
Displays the IPv4 routing protocols enabled on the device. For OSPFv2, it displays whether any route filters are applied, the RID, the number of areas the router is participating in, the types of areas, the maximum paths for load balancing, the network area command, the interfaces explicitly participating in the routing process, passive interfaces, routing information sources, and the AD.	show ip protocols
Displays the IPv6 dynamic routing protocols enabled on the device. For OSPFv3, it displays the PID, the RID, the number of areas, the type of areas, the interfaces participating in the routing process, and redistribution information.	show ipv6 protocols
Displays general OSPF parameters, including the PID, the RID, the reference bandwidth, the areas configured on the router, the types of areas (stub, totally stubby, NSSA, and totally NSSA), and area authentication.	show {ip ipv6} ospf
Displays the interfaces that are participating in the OSPF process.	show {ip ipv6} ospf interface brief
Displays detailed information about the interfaces participating in the OSPF process including interface IPv4 address and mask, area ID, PID, RID, network type, cost, DR/BDR, priority, and timers.	show {ip ipv6} ospf interface
Displays the OSPF devices that have formed a neighbor adjacency with the local router.	show {ip ipv6} ospf neighbor
Displays the OSPF routes that have been installed in the IPv4/IPv6 routing table.	show {ip ipv6} route {ospf ospfv3}

Task	Command Syntax
Displays general OSPFv3 parameters for IPv4 and IPv6 address families, including the PID, the RID, the reference bandwidth, the areas configured on the router, the types of areas (stub, totally stubby, NSSA, and totally NSSA), and area authentication.	<code>show ospfv3</code>
Displays the interfaces that are participating in the OSPFv3 process and the AF they are participating in.	<code>show ospfv3 interface brief</code>
Displays detailed information about the interfaces participating in the OSPFv3 address families including interface IPv4 and IPv6 addresses, area ID, PID, RID, network type, cost, DR/BDR, priority, and timers.	<code>show ospfv3 interface</code>
Displays the OSPFv3 neighbor adjacencies that have been formed for each AF.	<code>show ospfv3 neighbor</code>
Displays the OSPF link-state database.	<code>show {ip ipv6} ospf database</code>
Displays the OSPFv3 link-state database.	<code>show ospfv3 database</code>
Provides information about the status of OSPF virtual links that are required for areas not physically adjacent to the backbone area (that is, area 0).	<code>show {ip ipv6} ospf virtual-links</code>
Displays real-time information related to the exchange of OSPF hello packets. Useful for identifying mismatched OSPF timers and mismatched OSPF area types.	<code>debug {ip ipv6} ospf hello</code> <code>debug ospfv3 {ip ipv6} hello</code>
Displays the transmission and reception of OSPF packets in real time.	<code>debug {ip ipv6} ospf packet</code> <code>debug ospfv3 {ip ipv6} packet</code>
Displays real-time updates about the formation of an OSPF adjacency. Useful for identifying mismatched area IDs and authentication information.	<code>debug {ip ipv6} ospf adj</code> <code>debug ospfv3 {ip ipv6} adj</code>
This command shows real-time information about OSPF events, including the transmission and reception of hello messages and LSAs. This command might be useful on a router that appears to be ignoring hello messages received from a neighboring router.	<code>debug {ip ipv6} ospf events</code> <code>debug ospfv3 {ip ipv6} events</code>

This page intentionally left blank



This chapter covers the following topics:

- **Troubleshooting Route Maps:** This section explains how to read route maps and how they operate so that you can determine whether they are or are not the issue while troubleshooting other features that have them applied.
- **Troubleshooting Policy-Based Routing:** In this section, you learn the different reasons that could cause PBR not to operate as expected. You will also learn the commands that are needed to successfully troubleshoot issues related to PBR.
- **Policy-Based Routing Trouble Tickets:** This section provides trouble tickets that demonstrate how you can use a structured troubleshooting process to solve a reported problem.

Troubleshooting Route Maps and Policy-Based Routing

There are many different uses for route maps. So much so that when I hear the word route map, I think of duct tape. That's right; I said it, duct tape! Just like duct tape, route maps can fix anything. Therefore, when you need to fix routing problems by using policy-based routing (PBR), or manipulate the attributes of individual routes as they are being redistributed or learned via Border Gateway Protocol (BGP), you will use route maps.

This chapter begins by examining route maps. It gives you the opportunity to review how route maps are read and the commands that you can use to verify a route map's configuration. The rest of the chapter is dedicated to PBR, which allows you to override the router's default routing behavior. Because PBR relies on route maps, it makes sense to cover PBR at this point. Therefore, you will discover what could cause PBR not to behave as expected and how you can troubleshoot it.

“Do I Know This Already?” Quiz

The “Do I Know This Already?” quiz allows you to assess whether you should read this entire chapter thoroughly or jump to the “Exam Preparation Tasks” section. If you are in doubt about your answers to these questions or your own assessment of your knowledge of the topics, read the entire chapter. Table 16-1 lists the major headings in this chapter and their corresponding “Do I Know This Already?” quiz questions. You can find the answers in Appendix A, “Answers to the ‘Do I Know This Already?’ Quizzes.”

Table 16-1 “Do I Know This Already?” Section-to-Question Mapping

Foundation Topics Section	Questions
Troubleshooting Route Maps	1–2
Troubleshooting Policy-Based Routing	3–7

Caution The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you do not know the answer to a question or are only partially sure of the answer, you should mark that question as wrong for purposes of the self-assessment. Giving yourself credit for an answer that you correctly guess skews your self-assessment results and might provide you with a false sense of security.

1. What is the correct order of processing for a route map?
 - a. Top-down processing, implicit deny all at the end, immediate execution upon a match
 - b. Top-down processing, immediate execution upon a match, implicit deny all at the end
 - c. Immediate execution upon a match, implicit deny all at the end, top-down processing
 - d. Immediate execution upon a match, top-down processing, implicit deny all at the end
2. What will happen if none of the sequences match in a route map that is applied to redistribution?
 - a. The route will not be redistributed.
 - b. The route will be redistributed with default values.
 - c. The route will be redistributed based on the last permit sequence.
 - d. The route will be redistributed based on the values in the **redistribute** command.
3. What command enables you to verify the interfaces that have a PBR route map applied to them?
 - a. **show ip route**
 - b. **show ip policy**
 - c. **show route-map**
 - d. **show ip local policy**
4. What command enables you to verify the number of packets that have been policy-based routed?
 - a. **show ip route**
 - b. **show ip policy**
 - c. **show route-map**
 - d. **show ip local policy**
5. What command enables you to verify which PBR route map has been applied to locally generated packets?
 - a. **show ip route**
 - b. **show ip policy**
 - c. **show route-map**
 - d. **show ip local policy**

6. What will happen to packets that match a deny sequence in a route map that is used for PBR?
 - a. The packets will be routed normally.
 - b. The packets will be policy-based routed.
 - c. The packets will be dropped.
 - d. The packets will be routed upon approval by the admin.
7. Which Cisco IOS command enables you to verify that PBR is sending packets on the desired path?
 - a. `traceroute`
 - b. `show ip route`
 - c. `show route-map`
 - d. `show ip policy`

Foundation Topics

Troubleshooting Route Maps

Route maps are used with other services and features to provide a more granular level of control that was not available with the services or features by default. For example, when you redistribute routes from one routing protocol to another, all routes are redistributed and treated the same way. However, by attaching a route map to the redistribution process, you can treat each route or a group of routes differently when they are redistributed. In addition, route maps are heavily utilized with BGP for path manipulation, and they are the driving force behind PBR.

Therefore, when troubleshooting a service or feature that has a route map attached to it, you need to be able to troubleshoot the route map so that you can determine whether it is the cause of the issue. In this section, you learn how to read route maps.

How to Read a Route Map

A route map is identified by a name. Within the route map, there can be one or more sequences, which are defined by a number. Within each sequence, you can find *match* clauses and *set* clauses. Example 16-1 displays the output of **show run | section route-map**. It is a sample route map called TSHOOT_ROUTE_MAP. This route map is for illustrative purposes so that you can see the various options that a route map has to offer. You would not want to copy this route map for the real world because we have combined multiple features into one route map to give you various examples we will walk through. Example 16-2 displays the same route map but using the **show route-map [map_name]** command.

Example 16-1 Sample Route Map

```
R1#show run | section route-map
route-map TSHOOT_ROUTE_MAP permit 10
  match ip address 10 11
  set metric 500
route-map TSHOOT_ROUTE_MAP permit 20
  match ip address prefix-list OSPF_ROUTE
  set metric-type type-1
route-map TSHOOT_ROUTE_MAP permit 25
  match interface FastEthernet3/0
  set ip next-hop 10.1.12.2
route-map TSHOOT_ROUTE_MAP deny 30
  match tag 88
route-map TSHOOT_ROUTE_MAP permit 100
  set local-preference 150
```

Example 16-2 *Output of show route-map TSHOOT_ROUTE_MAP*

```

R1#show route-map TSHOOT_ROUTE_MAP
route-map TSHOOT_ROUTE_MAP, permit, sequence 10
  Match clauses:
    ip address (access-lists): 10 11
  Set clauses:
    metric 500
  Policy routing matches: 0 packets, 0 bytes
route-map TSHOOT_ROUTE_MAP, permit, sequence 20
  Match clauses:
    ip address prefix-lists: OSPF_ROUTE
  Set clauses:
    metric-type type-1
  Policy routing matches: 0 packets, 0 bytes
route-map TSHOOT_ROUTE_MAP, permit, sequence 25
  Match clauses:
    interface FastEthernet3/0
  Set clauses:
    ip next-hop 10.1.12.2
  Policy routing matches: 0 packets, 0 bytes
route-map TSHOOT_ROUTE_MAP, deny, sequence 30
  Match clauses:
    tag 88
  Set clauses:
  Policy routing matches: 0 packets, 0 bytes
route-map TSHOOT_ROUTE_MAP, permit, sequence 100
  Match clauses:
  Set clauses:
    local-preference 150
  Policy routing matches: 0 packets, 0 bytes

```

Key Topic

Notice how a sequence can be permit or deny. In this case, sequence 10, 20, 25, and 100 are all permit sequences, and 30 is a deny sequence. This is usually the culprit of many troubleshooting issues that involve route maps. Admins sometimes forget to type deny and as a result the sequence defaults to permit. In addition, depending on what the route map is being used for will determine what *permit* or *deny* truly means. For redistribution, permit means *redistribute* the route, and deny means *do not redistribute* the route. For PBR, permit means *policy-base route* the packet, and deny means *route the packet normally* using the routing table.

Key Topic

Review sequence 10. It is a permit statement that has a single match clause that matches IP address 10 and 11. What this truly means is match the IP addresses within access control list (ACL) 10 or ACL 11. When you see multiple match criteria within a single match clause, it means OR; therefore, ACL 10 OR 11. The traffic in question does not have to match both ACL 10 and 11, just 10 or 11. If the traffic in question matches sequence 10, the metric of the traffic in question will have its metric set to 500. Metrics are usually

manipulated during the redistribution process; therefore, sequence 10 is an example of a route map entry that you might use during redistribution.

Review sequence 20. It is a permit statement that has a single match clause, which matches a prefix list called `OSPF_ROUTE`. If the traffic in question matches the prefix list used in sequence 20, the metric type of the traffic in question will be changed to E1. Changing the metric type is something you can do when redistributing routes into Open Shortest Path First (OSPF). Therefore, sequence 20 is an example of a route map entry that you might use when redistributing routes into OSPF so that you can manipulate the metric type.

Review sequence 25. It is a permit statement that has a single match clause, which is matching all packets that arrive inbound on interface Fast Ethernet 3/0. Those packets that arrive in Fa3/0 will be forwarded out the interface that reaches the next-hop IP address of 10.1.12.2. This is an example of a route map entry that you would use with PBR to manually control how packets will be forwarded.

Review sequence 30. It is a deny statement with a single match clause. The match clause is matching routes with a tag of 88. When this type of route map is applied to redistribution, all routes that have a route tag of 88 will not be redistributed because a deny sequence means do not redistribute.

Review sequence 100. This is an example of a route map that can be applied to BGP for attribute manipulation. Notice that there is no match clause. When the match clause is missing in a sequence, it means match all. Therefore, all routes in question would match sequence 100 because the match clause is missing. The set clause states that the local preference, which is a BGP attribute, will be changed to 150.

The logic of a route map is very similar to an ACL. The following steps outline the logic of a route map:



1. **Top-down processing:** A route map is processed in order of sequence, starting with the lowest sequence in the route map to the highest sequence. In Examples 16-1 and 16-2, sequence 10 is processed first followed by 20, 25, 30, and then 100.
2. **Immediate execution upon a match:** During processing, a match clause in a sequence is evaluated. If the match clause matches the traffic in question, the processing stops and the actions defined in the set clauses in the sequence are executed in the order they are configured. If no match is found, the next sequence is checked. Note: If multiple match criteria are specified in the same match clause in a sequence, a logical OR algorithm is applied, which means that any of the match criteria can match for it to be considered a match. If multiple match criteria are specified in different match clauses in the same sequence, a logical AND algorithm is applied, which means that all of the match criteria must match for it to be considered a match.
3. **Implicit deny all:** If no sequence matches the traffic in question, the traffic is treated as though it matched a deny sequence, and is processed accordingly because there is an *implicit deny all* sequence at the end of every route map, just like ACLs and prefix lists.

Troubleshooting Policy-Based Routing

With PBR, you can create user-defined policies that manipulate how traffic will be routed through the network. By default, traffic is routed based on the destination IP address of a packet. However, with PBR you can override this behavior and have traffic routed based on different parameters matched in an ACL, or an inbound interface, for example. As a result, you can route based on source IP address or a destination port number, to name a few.

In this section, you learn the commands needed to troubleshoot issues related to PBR.

PBR

The driving force of PBR is route maps. Therefore, if you are not able to read route maps and understand what they are doing, you cannot troubleshoot PBR. Review Example 16-3, which shows a sample PBR configuration based on Figure 16-1. Although it is a small example, notice that multiple configurations are involved with PBR that you will have to review when troubleshooting (in this case, an ACL, a route map with match and set clauses, and the interface PBR is applied to).

Example 16-3 Sample PBR configuration

```
Branch#
access-list 100 permit ip 10.1.4.0 0.0.0.255 10.1.1.0 0.0.0.255
!
route-map PBR_EXAMPLE permit 10
match ip address 100
set ip next-hop 10.1.14.1
!
interface GigabitEthernet0/0
ip policy route-map PBR_EXAMPLE
```

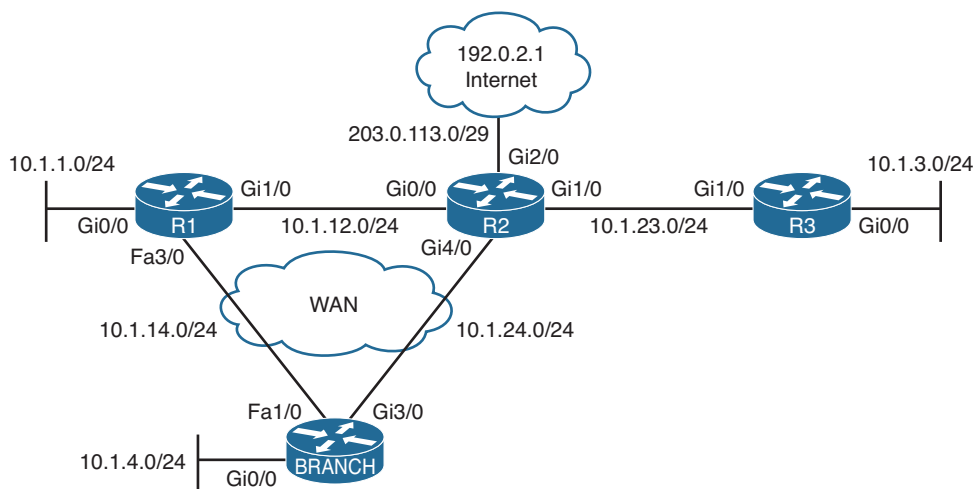


Figure 16-1 PBR Example Topology

When troubleshooting PBR, consider the following:



- **How the policy has been applied:** PBR is only applied to inbound packets on an interface or locally generated packets by the router. Therefore, you must ensure that you applied the correct PBR route map to the correct interface or the local router. You can use the **show ip policy** command to verify which interfaces are enabled for PBR and which route map has been applied, as shown in Example 16-4. You can use the **show ip local policy** command to display the route map that has been applied for local policy routing (traffic generated by the router).
- **How the route map is ordered:** Remember that route maps are processed from lowest sequence number to highest sequence number, and once a match is found within a sequence, the processing stops and the actions within that sequence are executed. Therefore, the order of the route map is important for proper execution. Use the **show route-map** command to verify the order of sequences within the route map, as shown in Example 16-5.
- **What permit and deny means:** When a PBR route map sequence is permit, it means *to policy-base route* the packet according to the action defined in the set clause. When a PBR route map sequence is deny, it means *do not policy-base route the packet*; therefore, route the packet normally. If you fail to specify permit or deny when creating the sequence, it defaults to permit. If by accident you specify permit or deny when you needed the opposite, you will have an issue because the desired results will not be achieved. Also, always remember that there is an implicit deny sequence at the end of a route map. Therefore, if the traffic in question does not match any of the explicit sequences within the route map, it ends up matching the implicit deny sequence. The implicit deny sequence within a route map for PBR means to route the traffic normally. Use the **show route-map** command to verify the permit and deny sequences within the route map, as shown in Example 16-5.
- **What traffic is being matched:** There are different methods of matching traffic for PBR within a route map. You can match ACLs, prefix lists, and inbound interfaces, to name a few. Based on the match clause, which you can verify with the **show route-map** command, as shown in Example 16-5, you need to verify whether the match criteria is correct using other **show** commands. For example, if the match clause is matching an IP ACL, you need to use the **show ip access-list** command to verify that the ACL is correct. If the match clause is matching an IP prefix list, you need to use the **show ip prefix-list** command to verify the prefix list is correct. Remember, if there is no match clause in the sequence, it means match all.
- **What action will be performed:** Once traffic matches a certain sequence, the action defined in the set clause is executed. When troubleshooting PBR, use the **show route-map** command, as shown in Example 16-5, to verify that the correct set clause has been configured.

**Key
Topic****Example 16-4** *Example of the show ip policy Command*

```
Branch#show ip policy
Interface      Route map
Gi0/0         PBR_EXAMPLE
```

**Key
Topic****Example 16-5** *Example of the show route-map Command*

```
Branch#show route-map
route-map PBR_EXAMPLE, permit, sequence 10
  Match clauses:
    ip address (access-lists): 100
  Set clauses:
    ip next-hop 10.1.14.1
Policy routing matches: 30 packets, 3420 bytes
```

When troubleshooting PBR, you will want to test the path that traffic is taking. You can accomplish this using a *traceroute*, as shown in Example 16-6. (On a PC, use **tracert**, and on a Cisco IOS device, use the **traceroute** command.) In this example, the packets destined to 10.1.1.1 are being policy-based routed to the next hop 10.1.14.1 even though the routing table entry states to use 10.1.24.2. All other packets are using 10.1.24.2 as the next hop because they are not being policy-based routed, as shown by the traceroute to 192.0.2.1. To verify that packets are being policy-based routed, use the **show route-map** command, as shown in Example 16-7. The output shows that 36 packets have been policy-based routed.

**Key
Topic****Example 16-6** *Example Traceroute to Verify the PBR Path*

```
C:\>tracert 10.1.1.1
Tracing route to 10.1.1.1 over a maximum of 30 hops

 1      6 ms      1 ms      2 ms  10.1.4.4
 2      6 ms      1 ms      2 ms  10.1.14.1
Trace complete.

Branch#show ip route 10.1.1.1
Routing entry for 10.1.1.0/24
...output omitted...
Routing Descriptor Blocks:
* 10.1.24.2, from 10.1.24.2, 00:14:36 ago, via GigabitEthernet3/0
  Route metric is 20480, traffic share count is 1
...output omitted...

C:\>tracert 192.0.2.1
Tracing route to 192.0.2.1 over a maximum of 30 hops

 1      6 ms      1 ms      2 ms  10.1.4.4
```



```

2      6 ms      1 ms      2 ms      10.1.24.2
...output omitted...
Trace complete.

```

Key Topic

Example 16-7 Using show route-map to Verify PBR Statistics

```

Branch#show route-map
route-map PBR_EXAMPLE, permit, sequence 10
  Match clauses:
    ip address (access-lists): 100
  Set clauses:
    ip next-hop 10.1.14.1
Policy routing matches: 36 packets, 3780 bytes

```

Key Topic

To see policy routing in real time, use the **debug ip policy** command, as shown in Example 16-8. In this example, the traffic sourced from 10.1.4.1 arriving inbound on Gig0/0 and destined to 10.1.1.1 has been policy matched to route map PBR_EXAMPLE sequence 10. Because it is a permit sequence, the packet is being policy-based routed from Gig0/0 to Fa1/0 with a next-hop address of 10.1.14.1.

Example 16-8 Using debug ip policy to view PBR in Real Time

```

Branch#debug ip policy
Policy routing debugging is on
Branch#
IP: s=10.1.4.1 (GigabitEthernet0/0), d=10.1.1.1, len 28, policy match
IP: route map PBR_EXAMPLE, item 10, permit
IP: s=10.1.4.1 (GigabitEthernet0/0), d=10.1.1.1 (FastEthernet1/0), len 28, policy
routed
IP: GigabitEthernet0/0 to FastEthernet1/0 10.1.14.1

```

Policy-Based Routing Trouble Tickets

This section presents various trouble tickets relating to the topics discussed earlier in the chapter. The purpose of these trouble tickets is to give a process that you can follow when troubleshooting in the real world or in an exam environment. All trouble tickets in this section are based on the topology depicted in Figure 16-2.

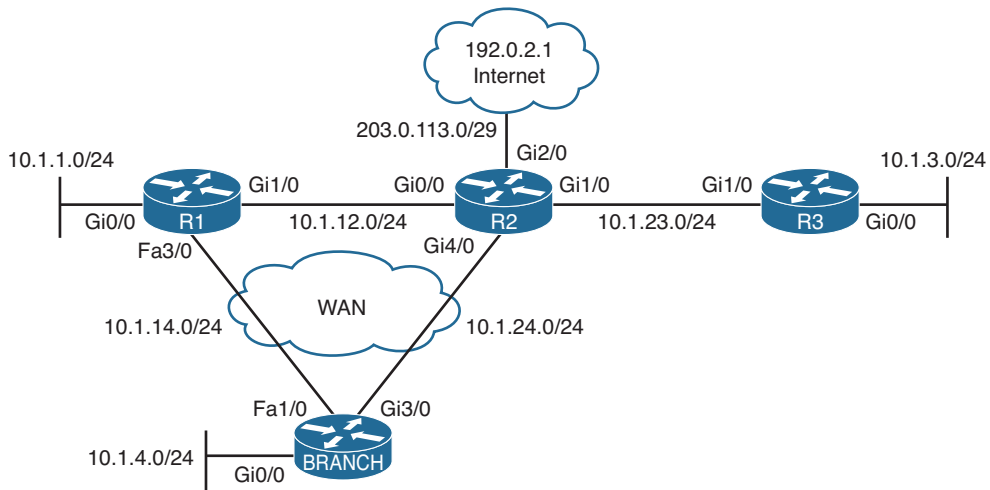


Figure 16-2 PBR Trouble Tickets Topology

Trouble Ticket 16-1

Problem: Traffic from 10.1.4.0/24 to 10.1.1.0/24 is routed though R2 using Gi3/0 when it should be routed directly to R1 using Fa1/0.

You begin troubleshooting by verifying the problem with a trace from a PC in 10.1.4.0/24 with a destination of 10.1.1.1. As shown in Example 16-9, the path to R2 is taken based on the hop 10.1.24.2.

Example 16-9 Verifying the Problem with a Trace to 10.1.1.1

```
C:\>tracert 10.1.1.1
Tracing route to 10.1.1.1 over a maximum of 30 hops

 1      6 ms      1 ms      2 ms  10.1.4.4
 2      8 ms      3 ms      4 ms  10.1.24.2
 3     12 ms      5 ms      8 ms  10.1.12.1
Trace complete.
```

You access Branch and issue the **show ip route** command. As shown in Example 16-10, the 10.1.1.0/24 network is reachable via a next hop of 10.1.24.2. However, as shown in Example 16-11, the Enhanced Interior Gateway Protocol (EIGRP) topology table indicates that there is another path that can be used via 10.1.14.1. It is not being used by EIGRP because it does not have the best feasible distance (metric). Therefore, you have confirmed that both paths exist and EIGRP is making the best decision. To force the traffic from 10.1.4.0 to 10.1.1.0 to use the Fast Ethernet link, PBR is being used. Therefore, you shift your attention to the PBR configuration.

Example 16-10 *Verifying Routing Table Entries*

```
Branch#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is 10.1.24.2 to network 0.0.0.0

D*EX 0.0.0.0/0 [170/15360] via 10.1.24.2, 01:10:05, GigabitEthernet3/0
      10.0.0.0/8 is variably subnetted, 10 subnets, 2 masks
D      10.1.1.0/24 [90/20480] via 10.1.24.2, 01:10:05, GigabitEthernet3/0
D      10.1.3.0/24 [90/20480] via 10.1.24.2, 01:10:05, GigabitEthernet3/0
C      10.1.4.0/24 is directly connected, GigabitEthernet0/0
L      10.1.4.4/32 is directly connected, GigabitEthernet0/0
D      10.1.12.0/24 [90/15360] via 10.1.24.2, 01:10:05, GigabitEthernet3/0
C      10.1.14.0/24 is directly connected, FastEthernet1/0
L      10.1.14.4/32 is directly connected, FastEthernet1/0
D      10.1.23.0/24 [90/15360] via 10.1.24.2, 01:10:05, GigabitEthernet3/0
C      10.1.24.0/24 is directly connected, GigabitEthernet3/0
L      10.1.24.4/32 is directly connected, GigabitEthernet3/0
      192.0.2.0/32 is subnetted, 1 subnets
D EX   192.0.2.1 [170/573440] via 10.1.24.2, 00:00:06, GigabitEthernet3/0
      203.0.113.0/29 is subnetted, 1 subnets
D      203.0.113.0 [90/15360] via 10.1.24.2, 01:10:05, GigabitEthernet3/0
```

Example 16-11 *Verifying All EIGRP Routes*

```
Branch#show ip eigrp topology
EIGRP-IPv4 VR(TSHOOT) Topology Table for AS(100)/ID(10.1.24.4)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 10.1.12.0/24, 1 successors, FD is 1966080
   via 10.1.24.2 (1966080/1310720), GigabitEthernet3/0
   via 10.1.14.1 (13762560/1310720), FastEthernet1/0
P 10.1.14.0/24, 1 successors, FD is 13107200
   via Connected, FastEthernet1/0
P 10.1.3.0/24, 1 successors, FD is 2621440
   via 10.1.24.2 (2621440/1966080), GigabitEthernet3/0
P 10.1.23.0/24, 1 successors, FD is 1966080
   via 10.1.24.2 (1966080/1310720), GigabitEthernet3/0
```

```

P 203.0.113.0/29, 1 successors, FD is 1966080
    via 10.1.24.2 (1966080/1310720), GigabitEthernet3/0
P 10.1.4.0/24, 1 successors, FD is 1310720
    via Connected, GigabitEthernet0/0
P 10.1.24.0/24, 1 successors, FD is 1310720
    via Connected, GigabitEthernet3/0
P 0.0.0.0/0, 1 successors, FD is 1966080
    via 10.1.24.2 (1966080/1310720), GigabitEthernet3/0
P 192.0.2.1/32, 1 successors, FD is 73400320, U
    via 10.1.24.2 (73400320/72744960), GigabitEthernet3/0
    via 10.1.14.1 (78643200/72089600), FastEthernet1/0
P 10.1.1.0/24, 1 successors, FD is 2621440
    via 10.1.24.2 (2621440/1966080), GigabitEthernet3/0
    via 10.1.14.1 (13762560/1310720), FastEthernet1/0

```

Because PBR is applied to ingress traffic, you start verifying that Gig0/0 on Branch has a PBR route map attached by using the **show ip policy** command. As shown in Example 16-12, the route map named PBR_EXAMPLE has been applied.

Example 16-12 Verifying a PBR Route Map Is Applied to the Correct Interface

```

Branch#show ip policy
Interface      Route map
Gi0/0          PBR_EXAMPLE

```

Next you issue the **show route-map** command to verify the route map, as shown in Example 16-13. There is only a single sequence, and it is a permit sequence that states any traffic matching the addresses in ACL 100 will be policy routed to a next-hop address of 10.1.14.1 *if and only if there is no specific route in the routing table*. Read that sentence again. Why is it *if and only if there is no specific route in the routing table*? This is because the **ip default next-hop** command was used. When this command is used, PBR examines the routing table, and if there is a specific route in the routing table, it is used. If there is no specific route in the routing table, the packet will be policy-based routed.

Example 16-13 Verifying Route Map Configuration

```

Branch#show route-map
route-map PBR_EXAMPLE, permit, sequence 10
  Match clauses:
    ip address (access-lists): 100
  Set clauses:
    ip default next-hop 10.1.14.1
  Policy routing matches: 0 packets, 0 bytes

```

Based on Example 16-10, there is a specific route in the routing table to reach 10.1.1.0/24. Therefore, the packets will not be policy-based routed. To solve this problem, you need

to change the **ip default next-hop** command to **ip next-hop**. Example 16-14 provides the configuration needed to solve this issue.

Example 16-14 *Modifying Route Map Configuration*

```
Branch#config t
Enter configuration commands, one per line. End with CNTL/Z.
Branch(config)#route-map PBR_EXAMPLE permit 10
Branch(config-route-map)#no set ip default next-hop 10.1.14.1
Branch(config-route-map)#set ip next-hop 10.1.14.1
Branch(config-route-map)#end
```

After the configuration has been modified, you verify the changes with the **show route-map** command, as shown in Example 16-15. Now it states *ip next-hop 10.1.14.1*.

Example 16-15 *Verifying the New Route Map Configuration*

```
Branch#show route-map
route-map PBR_EXAMPLE, permit, sequence 10
  Match clauses:
    ip address (access-lists): 100
  Set clauses:
    ip next-hop 10.1.14.1
Policy routing matches: 0 packets, 0 bytes
```

You issue the same trace from the client PC that you did at the start, and the trace confirms that packets are going across the Fast Ethernet link because of the hop with the IP 10.1.14.1, as shown in Example 16-16. To further confirm, you issue the command **show route-map** again on Branch, as shown in Example 16-17, and notice that packets have been successfully policy-based routed. Issue solved!

Example 16-16 *Confirming Packets Are Taking the Correct Path*

```
C:\>tracert 10.1.1.1
Tracing route to 10.1.1.1 over a maximum of 30 hops

  1      6 ms      1 ms      2 ms  10.1.4.4
  2      8 ms      3 ms      4 ms  10.1.14.1
Trace complete.
```

Example 16-17 *Verifying Policy Matches*

```
Branch#show route-map
route-map PBR_EXAMPLE, permit, sequence 10
  Match clauses:
    ip address (access-lists): 100
  Set clauses:
    ip next-hop 10.1.14.1
Policy routing matches: 6 packets, 360 bytes
```

Trouble Ticket 16-2

Problem: Traffic from 10.1.4.0/24 to 10.1.1.0/24 is routed though R2 using Gi3/0 when it should be routed directly to R1 using Fa1/0.

You begin troubleshooting by verifying the problem with a trace from a PC in 10.1.4.0/24 (Branch) with a destination of 10.1.1.1. As shown in Example 16-18, the path to R2 is used based on the hop 10.1.24.2.

Example 16-18 Verifying the Problem with a Trace to 10.1.1.1

```
C:\>tracert 10.1.1.1
Tracing route to 10.1.1.1 over a maximum of 30 hops

  1      6 ms      1 ms      2 ms  10.1.4.4
  2      8 ms      3 ms      4 ms  10.1.24.2
  3     12 ms      5 ms      8 ms  10.1.12.1
Trace complete.
```

Because the traffic is supposed to be policy-based routed, you access Branch and issue the **debug ip policy** command. You then perform the traceroute on the client again and observe the output of the **debug** commands on Branch. As shown in Example 16-19, there is a policy match for the deny sequence of 10 in the PBR_EXAMPLE route map. The **debug** then states that the policy is rejected, and the packet is routed based on the routing table.

So, even though there is a match, the packet is being routed normally. This is because it is a deny sequence that is matched. A deny sequence means *do not policy-base route, route normally* instead.

Example 16-19 Observing debug ip policy output

```
Branch#debug ip policy
Policy routing debugging is on
Branch#
IP: s=10.1.4.1 (GigabitEthernet0/0), d=10.1.1.1, len 28, policy match
IP: route map PBR_EXAMPLE, item 10, deny
IP: s=10.1.4.1 (GigabitEthernet0/0), d=10.1.1.1, len 28, policy rejected -- normal forwarding
Branch#
```

Next you issue the **show route-map** command to verify the route map, as shown in Example 16-20. There is only a single sequence, and it is a deny sequence that states any traffic matching the addresses in ACL 100 will be routed normally regardless of any set clauses because it is a deny sequence.

Example 16-20 *Verifying Route Map Configuration*

```
Branch#show route-map
route-map PBR_EXAMPLE, deny, sequence 10
  Match clauses:
    ip address (access-lists): 100
  Set clauses:
    ip next-hop 10.1.14.1
Nexthop tracking current: 0.0.0.0
10.1.14.1, fib_nh:0,oce:0,status:0

Policy routing matches: 0 packets, 0 bytes
```

To solve this problem, you need to change sequence 10 so that it is *permit* instead of deny. Example 16-21 displays the configuration needed to solve this issue.

Example 16-21 *Modifying Route Map Configuration*

```
Branch#config t
Enter configuration commands, one per line. End with CNTL/Z.
Branch(config)#route-map PBR_EXAMPLE permit 10
Branch(config-route-map)#end
```

After modifying the configuration, you verify the changes with the **show route-map** command, as shown in Example 16-22. Now sequence 10 is a permit sequence.

Example 16-22 *Verifying the New Route Map Configuration*

```
Branch#show route-map
route-map PBR_EXAMPLE, permit, sequence 10
  Match clauses:
    ip address (access-lists): 100
  Set clauses:
    ip next-hop 10.1.14.1
Policy routing matches: 0 packets, 0 bytes
```

You issue the same trace from the client PC that you did at the start, and the trace confirms that packets are going across the Fast Ethernet link because of the hop with the IP 10.1.14.1, as shown in Example 16-23. To further confirm, you observe the **debug** commands on Branch, as shown in Example 16-24, and it states that the traffic is being policy-based routed. Issue solved!

Example 16-23 *Confirming Packets Are Taking the Correct Path*

```
C:\>tracert 10.1.1.1
Tracing route to 10.1.1.1 over a maximum of 30 hops

  1      6 ms      1 ms      2 ms  10.1.4.4
  2      8 ms      3 ms      4 ms  10.1.14.1
Trace complete.
```

Example 16-24 *Verifying PBR with debug Commands*

```
Branch#debug ip policy
IP: s=10.1.4.1 (GigabitEthernet0/0), d=10.1.1.1, len 28, policy match
IP: route map PBR_EXAMPLE, item 10, permit
IP: s=10.1.4.1 (GigabitEthernet0/0), d=10.1.1.1 (FastEthernet1/0), len 28, policy
routed
IP: GigabitEthernet0/0 to FastEthernet1/0 10.1.14.1
```

Trouble Ticket 16-3

Problem: Traffic from 10.1.4.0/24 to 10.1.1.0/24 is routed though R2 using Gi3/0 when it should be routed directly to R1 using Fa1/0.

You begin troubleshooting by verifying the problem with a trace from a PC in 10.1.4.0/24 with a destination of 10.1.1.1. As shown in Example 16-25, the path to R2 is taken based on the hop 10.1.24.2. This traffic should have been policy-based routed to the next hop IP of 10.1.14.1.

Example 16-25 *Verifying the Problem with a Trace to 10.1.1.1*

```
C:\>tracert 10.1.1.1
Tracing route to 10.1.1.1 over a maximum of 30 hops

  1      6 ms      1 ms      2 ms  10.1.4.4
  2      8 ms      3 ms      4 ms  10.1.24.2
  3     12 ms      5 ms      8 ms  10.1.12.1
Trace complete.
```

Because PBR is applied to ingress traffic, you start verifying that Gig0/0 on Branch has a PBR route map attached by using the **show ip policy** command. As shown in Example 16-26, the route map named PBR_EXAMPLE has been applied to interface Fa0/1. There is no route map applied to Gig0/0 for PBR. However, before you conclude that the route map PBR_EXAMPLE was applied to the wrong interface, make sure that it is the route map that is needed to accomplish the goal. It would be bad if you removed this route map from Fa1/0 and applied it to Gig0/0 when that is not the true solution to the problem.

Example 16-26 *Verifying That the PBR Route Map Is Applied to the Correct Interface*

```
Branch#show ip policy
Interface      Route map
Fa1/0          PBR_EXAMPLE
```

Next you issue the **show route-map PBR_EXAMPLE** command to verify the route map, as shown in Example 16-27. There is only a single sequence, and it is a permit sequence that states any traffic matching the addresses in ACL 100 will be policy-base routed to a next-hop address of 10.1.14.1. Now it is time to verify ACL 100 with the **show access-list 100** command, as shown in Example 16-28. ACL 100 is matching traffic sourced with any address from 10.1.4.0 to 10.1.4.255 and destined to any address from 10.1.1.0 to

10.1.1.255. You have verified that this is the correct ACL, and the route map is correct as well. Therefore, the route map has been applied to the wrong interface.

Example 16-27 Verifying Route Map Configuration

```
Branch#show route-map PBR_EXAMPLE
route-map PBR_EXAMPLE, permit, sequence 10
  Match clauses:
    ip address (access-lists): 100
  Set clauses:
    ip next-hop 10.1.14.1
  Policy routing matches: 0 packets, 0 bytes
```

Example 16-28 Verifying ACL 100 Configuration

```
Branch#show access-lists 100
Extended IP access list 100
  10 permit ip 10.1.4.0 0.0.0.255 10.1.1.0 0.0.0.255
```

To solve this problem, you need to remove the **ip policy route-map** command from Fa1/0 and apply it to interface Gig0/0 instead. Example 16-29 provides the configuration needed to solve this issue.

Example 16-29 Modifying the ip policy route-map Configuration

```
Branch#config t
Enter configuration commands, one per line. End with CNTL/Z.
Branch(config)#int fa1/0
Branch(config-if)#no ip policy route-map PBR_EXAMPLE
Branch(config-if)#int gig 0/0
Branch(config-if)#ip policy route-map PBR_EXAMPLE
```

After modifying the configuration, you verify the changes with the **show ip policy** command, as shown in Example 16-30. Now the route map PBR_EXAMPLE is applied to Gig0/0.

Example 16-30 Verifying That the Route Map Is Applied to the Correct Interface

```
Branch#show ip policy
Interface      Route map
Gi0/0         PBR_EXAMPLE
```

You issue the same trace from the client PC that you did at the start, and the trace confirms that packets are going across the Fast Ethernet link because of the hop with the IP 10.1.14.1, as shown in Example 16-31. Issue solved!

Example 16-31 Confirming Packets Are Taking the Correct Path

```
C:\>tracert 10.1.1.1
Tracing route to 10.1.1.1 over a maximum of 30 hops

 1      6 ms      1 ms      2 ms  10.1.4.4
 2      8 ms      3 ms      4 ms  10.1.14.1
Trace complete.
```

Exam Preparation Tasks

As mentioned in the section “How to Use This Book” in the Introduction, you have a couple of choices for exam preparation: the exercises here; Chapter 22, “Final Preparation;” and the exam simulation questions on the CD-ROM.

Review All Key Topics

Review the most important topics in this chapter, noted with the Key Topic icon in the outer margin of the page. Table 16-2 lists a reference of these key topics and the page numbers on which each is found.



Table 16-2 *Key Topics for Chapter 16*

Key Topic Element	Description	Page Number
Paragraph	Describes the difference between a permit and deny sequence	679
Paragraphs	Examples of how to read route maps	679
Steps	Identifies the order that route maps are processed and how they are executed	680
List	Outlines what you should consider when troubleshooting issues related to PBR	682
Example 16-4	Example of show ip policy command	683
Example 16-5	Example of show route map command	683
Example 16-6	Example traceroute to verify PBR path	683
Example 16-7	Using show route map to verify PBR statistics	684
Paragraph	Describes how to verify PBR in real time	684

Define Key Terms

Define the following key terms from this chapter and check your answers in the glossary:

route map, match, set, implicit deny all, policy-based routing (PBR)

Command Reference to Check Your Memory

This section includes the most important **show** and **debug** commands covered in this chapter. It might not be necessary to memorize the complete syntax of every command, but you should be able to remember the basic keywords that are needed.

To test your memory of the commands, cover the right side of Table 16-3 with a piece of paper, read the description on the left side, and then see how much of the command you can remember.

The 300-135 TSHOOT exam focuses on practical, hands-on skills that are used by a networking professional. Therefore, you should be able to identify the commands needed to successfully troubleshoot the topics and concepts covered in this chapter.

Table 16-3 *show and debug commands*

Task	Command Syntax
Displays the commands that were used to configure all route maps on the router.	show run section route-map
Displays the route maps configured on the router. If you provide the name of the route map, it will only display that specific route map. The output provides all permit and deny sequences, the match clauses, the set clauses, and if used with PBR, it will also display the number of packets that have matched and been policy routed.	show route-map <i>[map_name]</i>
Displays the access lists configured on the device.	show access-list
Displays the PBR route maps that have been applied to the interfaces.	show ip policy
Displays the PBR route map that has been applied to the locally generated traffic of the device.	show ip local policy
Displays in real time the packets that have been policy-based routed.	debug ip policy

This page intentionally left blank



This chapter covers the following topics:

- **Troubleshooting IPv4 and IPv6 Redistribution:** This section examines the issues that you should look out for when troubleshooting redistribution for IPv4 and IPv6 routing protocols such as RIP, EIGRP, OSPF, and BGP.
- **Redistribution Trouble Tickets:** This section provides trouble tickets that demonstrate how you can use a structured troubleshooting process to solve a reported problem.
- **Troubleshooting Advanced Redistribution Issues:** This section explains the issues that could arise when you redistribute at multiple points in the network. In addition, you will discover how to recognize them and solve them.

Troubleshooting Redistribution

There are many reasons why you might need redistribution. It could be because you are performing a migration from one protocol to another, it might be because there are services or applications that need a specific routing protocol, it could be because you are in a mixed-vendor environment and only certain protocols are supported on the various devices, and it might even be because of political issues or country specific requirements. However, regardless of the reason, when you are using multiple routing protocols, you will more than likely be redistributing between the two so that all networks can be reached by all users in the network. As a result of this, you will more than likely experience some issues that will require you to troubleshoot.

This chapter explains the differences of redistributing into Enhanced Interior Gateway Routing Protocol (EIGRP), Open Shortest Path First (OSPF), Routing Information Protocol (RIP), and Border Gateway Protocol (BGP) for both IPv4 and IPv6. You will learn what to look out for so that you can quickly solve any issues related to redistribution. In addition, you will examine what could occur in environments that have multiple points of redistribution and how you can identify the issues and solve them.

“Do I Know This Already?” Quiz

The “Do I Know This Already?” quiz allows you to assess whether you should read this entire chapter thoroughly or jump to the “Exam Preparation Tasks” section. If you are in doubt about your answers to these questions or your own assessment of your knowledge of the topics, read the entire chapter. Table 17-1 lists the major headings in this chapter and their corresponding “Do I Know This Already?” quiz questions. You can find the answers in Appendix A, “Answers to the ‘Do I Know This Already?’ Quizzes.”

Table 17-1 “Do I Know This Already?” Section-to-Question Mapping

Foundation Topics Section	Questions
Troubleshooting IPv4 and IPv6 Redistribution	1–9
Troubleshooting Advanced Redistribution Issues	10

Caution The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you do not know the answer to a question or are only partially sure of the answer, you should mark that question as wrong for purposes of the self-assessment. Giving yourself credit for an answer that you correctly guess skews your self-assessment results and might provide you with a false sense of security.

1. What must be true for a route from one routing source to be redistributed into a different routing source?
 - a. The routing sources must have a similar metric.
 - b. The routing sources must have a similar administrative distance.
 - c. The route must be in the routing table on the router performing redistribution.
 - d. The route must be a directly connected route on the router performing redistribution.
2. Which of the following routing protocols have a default seed metric of unreachable? (Choose two answers.)
 - a. RIP
 - b. EIGRP
 - c. OSPF
 - d. BGP
3. Which of the following routing protocols have a default seed metric of 20?
 - a. RIPv6
 - b. EIGRP for IPv6
 - c. OSPFv3
 - d. BGP
4. When redistributing, you have four options for the seed metric: the default value, specifying it with the **default-metric** command, using the metric option with the **redistribute** command, and using a route map. If all four of these are configured with different values, which will be preferred?
 - a. Default values
 - b. **default-metric** command
 - c. Metric option with the **redistribute** command
 - d. Route map attached to **redistribute** command

5. Which option is mandatory when redistributing EIGRP or OSPF routes into RIP?
 - a. metric
 - b. metric type
 - c. subnets
 - d. match
6. Which option is mandatory when redistributing RIP or OSPF routes into EIGRP?
 - a. metric
 - b. metric type
 - c. subnets
 - d. match
7. Which option is mandatory when redistributing classless networks into OSPF?
 - a. metric
 - b. metric type
 - c. subnets
 - d. match
8. Which of the following are not included when redistributing from one IPv6 routing protocol into another IPv6 routing protocol?
 - a. A prefix
 - b. A seed metric
 - c. Directly connected routes participating in the routing process
 - d. An administrative distance
9. During redistribution that uses route maps, what will occur to a route that matches a deny entry in the route map?
 - a. It will be redistributed with default values.
 - b. It will be redistributed with the values in the set clause.
 - c. It will be redistributed only if there is a routing table entry for it.
 - d. It will not be redistributed.
10. Which of the following are methods that can be used to solve routing issues caused by multi-point redistribution?
 - a. Modify the seed metrics of the redistributed routes
 - b. Modify the administrative distance of redistributed routes
 - c. Tag routes as they are redistributed and then deny them from being redistributed back into the originating routing source
 - d. Modify the metric used to reach the boundary routers

Foundation Topics

Troubleshooting IPv4 and IPv6 Redistribution

Route redistribution allows routes learned via one source (for example, statically configured, locally connected, or learned via a routing protocol) to be injected into a routing protocol. If two routing protocols are mutually redistributed, the routes learned via each routing protocol are injected into the other routing protocol.

This section explains how to troubleshoot redistribution issues.

Route Redistribution Overview

A router that connects two or more routing domains and will be the point of redistribution is known as a boundary router, as illustrated in Figure 17-1. A boundary router can redistribute static routes, connected routes, and routes learned via one routing protocol into another routing protocol.

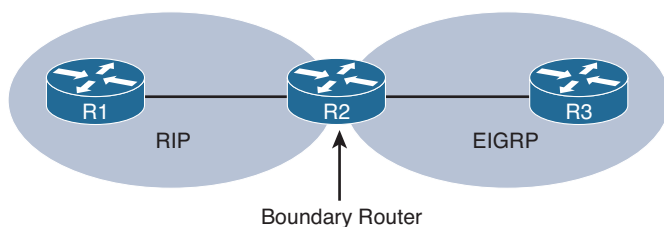


Figure 17-1 *Boundary Router*



Redistribution occurs from the routing table into a routing protocols data structure (such as the EIGRP topology table, or the OSPF link-state database [LSDB]), as shown in Figure 17-2. This is a key concept for troubleshooting purposes because if the route is not in the routing table, it cannot be redistributed. Keep in mind that if it is not in the routing table, some other underlying issue needs to be troubleshooted to get redistribution to work. For example, if you are redistributing EIGRP into OSPF and the EIGRP route is not in the routing table, that is not a redistribution problem; it is an EIGRP problem that has to be solved first.

Different routing protocols use different types of metrics, as illustrated in Figure 17-3. Therefore, when a route is redistributed into a routing protocol, a metric used by the destination routing protocol needs to be associated with the route being redistributed.

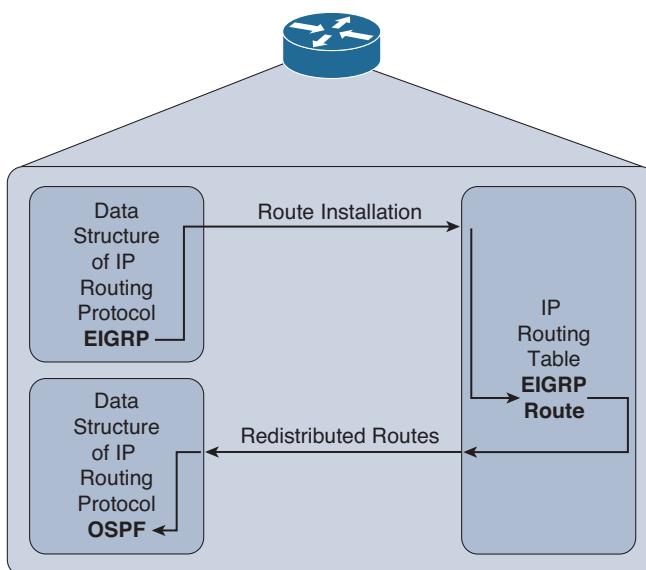


Figure 17-2 *Redistribution Occurs from the Routing Table into a Routing Protocols Data Structure*

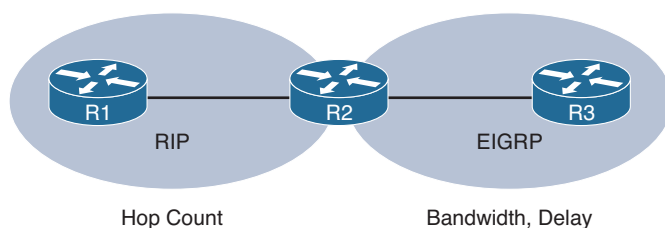


Figure 17-3 *Differing Metrics Between Routing Protocols*

The metric assigned to a route being redistributed into another routing process is called a seed metric. The seed metric is needed to communicate relative levels of reachability between dissimilar routing protocols. A seed metric can be defined in one of three ways:



- The **default-metric** command
- The **metric** parameter in the **redistribute** command
- A route map configuration applied to the **redistribute** command

The order of preference if multiple seed metrics are defined with the commands listed previously is 1) metric defined in route map that was applied to **redistribute** command; 2) metric parameter defined in **redistribute** command; 3) metric defined in **default-metric** command.

If a seed metric is not specified, a default seed metric is used. Keep in mind that RIP and EIGRP have a default seed metric that is considered unreachable. Therefore, if you do not manually configure a seed metric when redistributing routes into RIP or EIGRP, the

redistributed route will not be reachable and therefore not advertised to other routers in the routing domain. OSPF has a default seed metric of 20, unless it is a BGP route being redistributed, which would have a seed metric of 1. When redistributing into BGP, BGP will use the exact metric of the Interior Gateway Protocol (IGP).

Note For EIGRP and RIP you do not need to specify a metric when redistributing static or connected routes. In addition, for EIGRP you do not have to specify a metric when redistributing from another EIGRP autonomous system because the original metric is preserved.

Some routing protocols (for example, EIGRP and OSPF) can tag routes as either internal (that is, routes locally configured or connected) or external (that is, routes learned from another routing process) and give priority to internal routes versus external routes. The capability to distinguish between internal and external routes can help prevent a potential routing loop, where two routing protocols continually redistribute the same routes into one another at multiple redistribution points.

Before you move on to specific redistribution examples, keep the following in mind. Two prerequisites must be met for the routes of one IP routing protocol to be redistributed into another IP routing protocol:



- The route needs to be installed in the border routers (router performing redistribution) IP routing table by the protocol being redistributed.
- The destination IP routing protocol needs a reachable metric to assign to the redistributed routes.

Based on the previous two prerequisites, Table 17-2 lists various redistribution troubleshooting targets and recommendations for dealing with them.



Table 17-2 *Troubleshooting Targets for Route Redistribution*

Troubleshooting Target	Troubleshooting Recommendation
Source routing protocol	Verify that a route to be redistributed from a routing protocol has been learned by that routing protocol. Issue appropriate show commands for the data structures of the source routing protocol to ensure that the source routing protocol has learned the route in question.
Route selection	Because a route must be in a router's IP routing table to be redistributed, ensure that the routes of the source routing protocol are indeed being injected into the router's IP routing table.

Troubleshooting Target	Troubleshooting Recommendation
Redistribution configuration	If a route has been injected into a router's IP routing table from a source routing protocol but not redistributed into the destination routing protocol, check the redistribution configuration. This involves checking the metric applied to routes as they are redistributed into the destination routing protocol, checking for any route filtering that might be preventing redistribution, and checking the redistribution syntax to confirm that the correct routing process ID or autonomous system number is specified.
Destination routing protocol	If a route has been successfully redistributed into a destination routing protocol but the route has not been successfully learned by neighboring routers, you should investigate the destination routing protocol. You could use traditional methods of troubleshooting a destination routing protocol; however, keep in mind that the redistributed route might be marked as an external route. Therefore, check the characteristics of the destination routing protocol to determine whether it treats external routes differently from internal ones.

Troubleshooting Redistribution into RIP



Your options are limited when redistributing routes into RIPv2 and RIPv6. Review Example 17-1, it displays the options when redistributing OSPFv2 routes into RIPv2. This example is shown because it has the most options. When redistributing EIGRP, BGP, static or connected, you only have metric and route map as options. The most common issue you will run into when redistributing into RIPv2 is related to the metric. Remember that the seed metric by default is set to infinity (unreachable). Therefore, if you fail to manually set the metric using any of the options listed earlier in the chapter, routes will not be advertised to the other routers in the RIPv2 domain. In addition, if you configure the metric too high at the redistribution point, you could cause the route to become unreachable further in your RIPv2 domain, because RIP's metric is based on hop count. For example, if you specify a metric of 10 during redistribution, a router that is 6 hops away from the redistribution router will not receive the redistributed routes because the routes will be further than 15 hops away ($10 + 6 = 16$).

Also, if the wrong route map is applied, or there is an error within the route map, routes will not be redistributed properly.

Example 17-1 RIPv2 Redistribution Options

```
R1(config)#router rip
R1(config-router)#redistribute ospf 1 ?
  match      Redistribution of OSPF routes
  metric     Metric for redistributed routes
  route-map  Route map reference
  vrf        VPN Routing/Forwarding Instance
  <cr>
```

With RIP next generation (RIPng) redistribution, you can experience all the same issues that you do with RIPv2 in addition to another. By default, with RIPv2, the networks of the local interfaces participating in the routing process that is being redistributed on the border router into RIPv2 will be redistributed as well. However, with RIPng, they will not. Therefore, if you want to include the networks associated with the interfaces participating in the routing process that is being redistributed into RIPng on the boundary router, you need to use the **include-connected** keyword, as shown in Example 17-2.

Example 17-2 RIPng Redistribution Options

```
R1(config)#ipv6 router rip
R1(config-rtr)#redistribute ospf 1 ?
include-connected  Include connected
match             Redistribution of OSPF routes
metric            Metric for redistributed routes
route-map          Route map reference
<cr>
```

When redistributing OSPF into RIP, you also have the match option, which allows you to match just **internal**, just **external**, just **nssa-external** routes, or a combination of them. If the wrong options are chosen, the wrong routes will be redistributed resulting in missing routes.

On the boundary router, you can verify which routing protocols are being redistributed with the **show ip protocols** command and the routes that were redistributed with the **show ip rip database** command. In Example 17-3, the output of **show ip protocols** indicates that the EIGRP process with an autonomous system number of 100 is being redistributed into RIPv2. In Example 17-4, the output indicates that the 10.1.3.0/24 network is redistributed.

Example 17-3 Verifying Redistribution with show ip protocols

```
R2#show ip protocols
*** IP Routing is NSF aware ***
...output omitted...
Routing Protocol is "rip"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Sending updates every 30 seconds, next due in 21 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Redistributing: eigrp 100, rip
  Default version control: send version 2, receive version 2
    Interface          Send Recv Triggered RIP Key-chain
...output omitted...
```

Example 17-4 Verifying Redistribution with show ip rip database

```
R2#show ip rip database
10.0.0.0/8      auto-summary
10.1.1.0/24
```

```

[1] via 10.1.12.1, 00:00:19, GigabitEthernet0/0
10.1.3.0/24      redistributed
[5] via 10.1.23.3,
10.1.12.0/24    directly connected, GigabitEthernet0/0
10.1.14.0/24
[1] via 10.1.12.1, 00:00:19, GigabitEthernet0/0
10.1.23.0/24    directly connected, GigabitEthernet1/0

```

To verify that other RIP routers in the RIP domain are learning about the redistributed route, use the **show ip route** and **show ip rip database** commands on those routers, as shown in Example 17-5.

Example 17-5 *Verifying Redistributed Routes in the RIP Domain*

```

R1#show ip rip database
10.0.0.0/8      auto-summary
10.1.1.0/24     directly connected, GigabitEthernet0/0
10.1.3.0/24
[5] via 10.1.12.2, 00:00:00, GigabitEthernet1/0
10.1.12.0/24    directly connected, GigabitEthernet1/0
10.1.14.0/24    directly connected, FastEthernet3/0
10.1.23.0/24
[1] via 10.1.12.2, 00:00:00, GigabitEthernet1/0
R1#show ip route
...output omitted...
    10.0.0.0/8 is variably subnetted, 8 subnets, 2 masks
C       10.1.1.0/24 is directly connected, GigabitEthernet0/0
L       10.1.1.1/32 is directly connected, GigabitEthernet0/0
R       10.1.3.0/24 [120/5] via 10.1.12.2, 00:00:03, GigabitEthernet1/0
C       10.1.12.0/24 is directly connected, GigabitEthernet1/0
L       10.1.12.1/32 is directly connected, GigabitEthernet1/0
C       10.1.14.0/24 is directly connected, FastEthernet3/0
L       10.1.14.1/32 is directly connected, FastEthernet3/0
R       10.1.23.0/24 [120/1] via 10.1.12.2, 00:00:03, GigabitEthernet1/0

```

For RIPv6, the **show ipv6 protocols** output is more detailed for redistribution, as shown in Example 17-6. Notice how it states the protocol, the seed metric, and whether connected networks are included.

Example 17-6 *Verifying RIPv6 Redistribution with show ipv6 protocols*

```

R2#show ipv6 protocols
...output omitted...
IPv6 Routing Protocol is "rip TSHOOT_RIP"
  Interfaces:
    GigabitEthernet0/0
  Redistribution:
    Redistributing protocol eigrp 100 with metric 7 include-connected
...output omitted...

```



Troubleshooting Redistribution into EIGRP

When redistributing into EIGRP for IPv4 you can apply a metric with the **metric** keyword or a route map with the **route-map** keyword. If you are redistributing OSPF into EIGRP, as shown in Example 17-7, you will also have the option to specify the **match** option which allows you to match just **internal**, just **external**, just **nssa-external** routes, or a combination of them.

The most common issue you will run into when redistributing into EIGRP for IPv4 is related to the metric. Remember that the seed metric by default is set to infinity (unreachable). Therefore, if you fail to manually set the metric using any of the options listed earlier in the chapter, routes will not be advertised to the other routers in the EIGRP autonomous system. Unlike RIP, you will not have to worry about configuring a metric that is too high and causing networks to be unreachable. However, you have to consider if the metrics you specify will cause suboptimal routing if you have multiple redistribution points in the routing domain.

Also, if the wrong route map is applied, or there is an error within the route map, routes will not be redistributed properly.

Example 17-7 EIGRP for IPv4 Redistribution Options

```
R1(config)#router eigrp 1
R1(config-router)#redistribute ospf 1 ?
  match      Redistribution of OSPF routes
  metric     Metric for redistributed routes
  route-map  Route map reference
<cr>
```

With EIGRP for IPv6 you have the same **match**, **metric**, and **route-map** keywords, in addition to the **include-connected** keyword. By default, with EIGRP for IPv4, the networks associated with the local interfaces participating in the redistributed routing process will be redistributed as well. However, with EIGRP for IPv6 they will not. Therefore, if you want to include the networks associated with the local interfaces participating in the routing process that is being redistributed, you need to use the **include-connected** keyword, as shown in Example 17-8.

Example 17-8 EIGRP for IPv6 Redistribution Options

```
R1(config)#ipv6 router eigrp 1
R1(config-rtr)#redistribute ospf 1 ?
  include-connected Include connected
  match             Redistribution of OSPF routes
  metric            Metric for redistributed routes
  route-map         Route map reference
<cr>
```

On the boundary router, you can verify which protocols are being redistributed into EIGRP for IPv4 with the **show ip protocols** command. As shown in Example 17-9, RIP routes are being redistributed into EIGRP for IPv4.

Example 17-9 *Verifying Protocols That Are Being Redistributed into EIGRP for IPv4*

```

R2#show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "eigrp 100"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  Redistributing: rip
  EIGRP-IPv4 Protocol for AS(100)
    Metric weight K1=1, K2=0, K3=1, K4=0, K5=0
...output omitted...

```

When reviewing the EIGRP for IPv4 topology table with the **show ip eigrp topology** command, you can identify the routes that have been injected into the EIGRP process via redistribution because it states *via Redistributed*, as shown in Example 17-10.

Example 17-10 *Verifying Routes Redistributed into EIGRP for IPv4 (Topology Table)*

```

R2#show ip eigrp topology
EIGRP-IPv4 Topology Table for AS(100)/ID(203.0.113.1)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 10.1.12.0/24, 1 successors, FD is 2560000256
    via Redistributed (2560000256/0)
P 10.1.14.0/24, 1 successors, FD is 2560000256
    via Redistributed (2560000256/0)
P 10.1.13.0/24, 1 successors, FD is 3072
    via 10.1.23.3 (3072/2816), GigabitEthernet1/0
P 10.1.23.0/24, 1 successors, FD is 2816
    via Connected, GigabitEthernet1/0
P 10.1.1.0/24, 1 successors, FD is 2560000256
    via Redistributed (2560000256/0)

```

When examining a redistributed route in the routing table on the boundary router, as shown in Example 17-11, with the **show ip route ip-address** command, it indicates how the route is known, how it is being redistributed, and the EIGRP metric values that are being used at the redistribution point.

Example 17-11 *Verifying Routes Redistributed into EIGRP for IPv4 (Routing Table)*

```

R2#show ip route 10.1.1.0
Routing entry for 10.1.1.0/24
  Known via "rip", distance 120, metric 1
  Redistributing via eigrp 100, rip

```



```

Advertised by eigrp 100 metric 1 1 1 1 1
Last update from 10.1.12.1 on GigabitEthernet0/0, 00:00:19 ago
Routing Descriptor Blocks:
* 10.1.12.1, from 10.1.12.1, 00:00:19 ago, via GigabitEthernet0/0
  Route metric is 1, traffic share count is 1

```

When examining the routing table on other routers (not the boundary router) in the EIGRP for IPv4 autonomous system, the redistributed routes will have an administrative distance (AD) of 170 by default and a code of D EX, as shown in Example 17-12.

Example 17-12 Examining EIGRP for IPv4 Redistributed Routes in a Routing Table

```

R3#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

    10.0.0.0/8 is variably subnetted, 7 subnets, 2 masks
D EX    10.1.1.0/24
         [170/2560000512] via 10.1.23.2, 00:04:38, GigabitEthernet1/0
C       10.1.3.0/24 is directly connected, GigabitEthernet0/0
L       10.1.3.3/32 is directly connected, GigabitEthernet0/0
D EX    10.1.12.0/24
         [170/2560000512] via 10.1.23.2, 00:04:38, GigabitEthernet1/0
D EX    10.1.14.0/24
         [170/2560000512] via 10.1.23.2, 00:04:38, GigabitEthernet1/0
C       10.1.23.0/24 is directly connected, GigabitEthernet1/0
L       10.1.23.3/32 is directly connected, GigabitEthernet1/0

```

For EIGRP for IPv6, the **show ipv6 protocols** output is more detailed for redistribution, as shown in Example 17-13. Notice how it states the protocol, the seed metric, and whether connected networks are included.

Example 17-13 Verifying EIGRP for IPv6 Redistribution with show ipv6 protocols

```

R2#show ipv6 protocols
...output omitted...
IPv6 Routing Protocol is "eigrp 100"
EIGRP-IPv6 Protocol for AS(100)
  Metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  NSF-aware route hold timer is 240

```

```

Router-ID: 203.0.113.1
Topology : 0 (base)
  Active Timer: 3 min
  Distance: internal 90 external 170
  Maximum path: 16
  Maximum hopcount 100
  Maximum metric variance 1

Interfaces:
  GigabitEthernet1/0
Redistribution:
  Redistributing protocol rip TSHOOT_RIP with metric 1 1 1 1 1 include-connected

```

The output of **show ipv6 eigrp topology** on the boundary router also indicates which routes are redistributed, as shown in Example 17-14.

Example 17-14 *Verifying EIGRP for IPv6 Redistribution with show ipv6 eigrp topology*

```

R2#show ipv6 eigrp topology
EIGRP-IPv6 Topology Table for AS(100)/ID(203.0.113.1)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 2001:DB8:0:1::/64, 1 successors, FD is 2560000256
    via Redistributed (2560000256/0)
P 2001:DB8:0:3::/64, 1 successors, FD is 3072
    via FE80::C804:10FF:FE2C:1C (3072/2816), GigabitEthernet1/0
P 2001:DB8:0:12::/64, 1 successors, FD is 2560000256
    via Redistributed (2560000256/0)
P 2001:DB8:0:23::/64, 1 successors, FD is 2816
    via Connected, GigabitEthernet1/0

```

When examining the routing table on other routers (not the boundary router) in the EIGRP for IPv6 autonomous system, the redistributed routes will have an administrative distance of 170 by default and a code of EX, as shown in Example 17-15.

Example 17-15 *Verifying EIGRP for IPv6 Redistributed Routes*

```

R3#show ipv6 route
IPv6 Routing Table - default - 7 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, R - RIP, H - NHRP, I1 - ISIS L1
       I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary, D - EIGRP
       EX - EIGRP external, ND - ND Default, NDp - ND Prefix, DCE - Destination
       NDr - Redirect, O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1
       OE2 - OSPF ext 2, ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2, l - LISP
EX 2001:DB8:0:1::/64 [170/2560000512]
    via FE80::C802:AFF:FE88:1C, GigabitEthernet1/0

```

```

C   2001:DB8:0:3::/64 [0/0]
    via GigabitEthernet0/0, directly connected
L   2001:DB8:0:3::3/128 [0/0]
    via GigabitEthernet0/0, receive
EX  2001:DB8:0:12::/64 [170/2560000512]
    via FE80::C802:AFF:FE88:1C, GigabitEthernet1/0
C   2001:DB8:0:23::/64 [0/0]
    via GigabitEthernet1/0, directly connected
L   2001:DB8:0:23::3/128 [0/0]
    via GigabitEthernet1/0, receive
L   FF00::/8 [0/0]
    via Null0, receive

```

Troubleshooting Redistribution into OSPF



When redistributing into OSPF, you have more options than other routing protocols, as shown in Example 17-16. The **metric** option allows you to provide a seed metric at the redistribution point. The default seed metric is 20 with OSPF; therefore, providing a metric is not mandatory. If you forget to provide a metric, redistributed routes will still be advertised to other routers in the OSPF domain. The **metric-type** option is used to define the type of OSPF external route the redistributed route will be. By default, it will be Type 2, which is represented as E2 in the routing table. With E2, each router will preserve the seed metric for the external routes. Type 1, which is represented as E1 in the routing table, allows each router to take the seed metric and add to it all the other link costs to reach the redistribution point in the domain. Therefore, each router will have a metric that is a combination of the seed metric and the total cost to reach the redistribution router. With the **nssa-only** option, you can limit redistributed routes to the NSSA area only, and with the **route-map** option, you can reference a route map that provides more granular control over the routes that are being redistributed. The **subnets** keyword is an extremely important option. Without the **subnets** keyword, only classful networks will be redistributed (for example, a Class A address with a /8 mask, a Class B address with a /16 mask, and a Class C address with a /24 mask). With the **subnets** keyword, all classless and classful networks will be redistributed. Therefore, if you have any subnets that you want to redistribute, the **subnets** keyword is mandatory. The **tag** keyword can be used to add a numeric ID (tag) to the route so the route can be referenced by the tag at a later point for filtering or manipulation purposes.

Example 17-16 OSPFv2 Redistribution Options

```

R1(config)#router ospf 1
R1(config-router)#redistribute eigrp 100 ?
metric          Metric for redistributed routes
metric-type     OSPF/IS-IS exterior metric type for redistributed routes
nssa-only       Limit redistributed routes to NSSA areas
route-map       Route map reference
subnets        Consider subnets for redistribution into OSPF
tag             Set tag for routes redistributed into OSPF
<cr>

```

Look closely at Example 17-17, which displays the options available when redistributing into OSPFv3. What has been added and what is missing when compared to OSPFv2? The **include-connected** keyword has been added. By default, with OSPFv2, the networks associated with the local interfaces that are participating in the routing process that is being redistributed will be redistributed as well. However, with OSPFv3, they will not. Therefore, if you want to include the networks associated with the interfaces participating in the routing protocol that is being redistributed on the ASBR, you need to use the **include-connected** keyword.

The **subnets** keyword is not an option with OSPFv3 because the concept of classful and classless does not exist with IPv6.

Example 17-17 OSPFv3 Redistribution Options

```
R1(config)#ipv6 router ospf 1
R1(config-rtr)#redistribute eigrp 100 ?
  include-connected  Include connected
  metric             Metric for redistributed routes
  metric-type        OSPF/IS-IS exterior metric type for redistributed routes
  nssa-only          Limit redistributed routes to NSSA areas
  route-map          Route map reference
  tag                Set tag for routes redistributed into OSPF
<cr>
```

The **show ip protocols** command enables you to verify which routing protocols are being redistributed into the OSPFv2 process. In Example 17-18, you can see that EIGRP 100 routes, including subnets, are being redistributed into the OSPFv2 process.

Example 17-18 Verifying Protocols Being Redistributed into OSPFv2

```
R2#show ip protocols
...output omitted...
Routing Protocol is "ospf 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 203.0.113.1
  It is an autonomous system boundary router
  Redistributing External Routes from,
    eigrp 100, includes subnets in redistribution
  Number of areas in this router is 1. 1 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    10.1.12.2 0.0.0.0 area 0
  Routing Information Sources:
    Gateway         Distance      Last Update
    10.1.14.1        110          00:19:48
  Distance: (default is 110)
```

Routes redistributed into an OSPFv2 normal area will be advertised within a Type 5 link-state advertisement (LSA). Routes redistributed into an OSPFv2 NSSA or totally NSSA area will be advertised within a Type 7 LSA and then converted to a Type 5 LSA at an Area Border Router (ABR). You can view the redistributed routes that are injected into the OSPFv2 LSDB with the **show ip ospf database** command, as shown in Example 17-19. In this example, the 10.1.3.0 and 10.1.23.0 networks have been redistributed into the OSPFv2 routing process.

Example 17-19 *Verifying Redistributed Routes in the OSPFv2 LSDB*

```
R2#show ip ospf database
```

```
OSPF Router with ID (203.0.113.1) (Process ID 1)
```

```
Router Link States (Area 0)
```

Link ID	ADV Router	Age	Seq#	Checksum	Link count
10.1.14.1	10.1.14.1	738	0x80000003	0x009AEA	3
203.0.113.1	203.0.113.1	596	0x80000003	0x005829	1

```
Net Link States (Area 0)
```

Link ID	ADV Router	Age	Seq#	Checksum
10.1.12.1	10.1.14.1	738	0x80000002	0x001F8F

```
Type-5 AS External Link States
```

Link ID	ADV Router	Age	Seq#	Checksum	Tag
10.1.3.0	203.0.113.1	596	0x80000002	0x00EB67	0
10.1.23.0	203.0.113.1	596	0x80000002	0x000F30	0

When examining a redistributed route in the routing table on the boundary router (Autonomous System Boundary Router [ASBR]), as shown in Example 17-20, with the **show ip route ip_address** command, it indicates how the route is known, how it is being redistributed, and how it is being advertised. In this case, the route is known via EIGRP 100 and is being redistributed into the OSPF 1 process with the **subnets** keyword.

Example 17-20 *Verifying Redistributed Routes in the ASBR's Routing Table*

```
R2#show ip route 10.1.3.0
```

```
Routing entry for 10.1.3.0/24
```

```
Known via "eigrp 100", distance 90, metric 3072, type internal
```

```
Redistributing via eigrp 100, ospf 1
```

```
Advertised by ospf 1 subnets
```

```
Last update from 10.1.23.3 on GigabitEthernet1/0, 00:50:19 ago
```

```
Routing Descriptor Blocks:
```

```
* 10.1.23.3, from 10.1.23.3, 00:50:19 ago, via GigabitEthernet1/0
```

```
Route metric is 3072, traffic share count is 1
```

```
Total delay is 20 microseconds, minimum bandwidth is 1000000 Kbit
Reliability 255/255, minimum MTU 1500 bytes
Loading 1/255, Hops 1
```

When examining the routing table on other routers (not the ASBR) in the OSPFv2 domain, by default the redistributed routes will have an AD of 110 and a code of O E2, as shown in Example 17-21. If you change the metric type to E1, they will appear with a code of E1, and if it is an NSSA or totally NSSA area they will appear as O N1 or O N2.

Example 17-21 Examining OSPFv2 Redistributed Routes in a Routing Table

```
R1#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

    10.0.0.0/8 is variably subnetted, 8 subnets, 2 masks
C       10.1.1.0/24 is directly connected, GigabitEthernet0/0
L       10.1.1.1/32 is directly connected, GigabitEthernet0/0
O E2    10.1.3.0/24 [110/20] via 10.1.12.2, 00:49:11, GigabitEthernet1/0
C       10.1.12.0/24 is directly connected, GigabitEthernet1/0
L       10.1.12.1/32 is directly connected, GigabitEthernet1/0
C       10.1.14.0/24 is directly connected, FastEthernet3/0
L       10.1.14.1/32 is directly connected, FastEthernet3/0
O E2    10.1.23.0/24 [110/20] via 10.1.12.2, 00:49:11, GigabitEthernet1/0
```

For OSPFv3, the `show ipv6 protocols` output is seen in Example 17-22. Notice how it states the protocol, the seed metric, and if connected networks are included.

Example 17-22 Verifying OSPFv3 Redistribution with `show ipv6 protocols`

```
R2#show ipv6 protocols
...output omitted...
IPv6 Routing Protocol is "ospf 1"
  Router ID 2.2.2.2
  Autonomous system boundary router
  Number of areas: 1 normal, 0 stub, 0 nssa
  Interfaces (Area 0):
    GigabitEthernet0/0
  Redistribution:
    Redistributing protocol eigrp 100 with metric 10 include-connected
```

The output of **show ipv6 ospf database** on the ASBR will identify the external Type 5 routes just like OSPFv2, as shown in Example 17-23.

Example 17-23 *Verifying OSPFv3 Redistribution with show ipv6 ospf database*

```
R2#show ipv6 ospf database

        OSPFv3 Router with ID (2.2.2.2) (Process ID 1)

        Router Link States (Area 0)

ADV Router      Age      Seq#      Fragment ID  Link count  Bits
1.1.1.1         1429     0x80000004 0             1           B
2.2.2.2         1446     0x80000003 0             1           E

        Net Link States (Area 0)

ADV Router      Age      Seq#      Link ID      Rtr count
1.1.1.1         1429     0x80000002 4            2

        Inter Area Prefix Link States (Area 0)

ADV Router      Age      Seq#      Prefix
1.1.1.1         1693     0x80000002 2001:DB8:0:14::/64

        Link (Type-8) Link States (Area 0)

ADV Router      Age      Seq#      Link ID      Interface
1.1.1.1         1693     0x80000002 4            Gi0/0
2.2.2.2         1446     0x80000002 3            Gi0/0

        Intra Area Prefix Link States (Area 0)

ADV Router      Age      Seq#      Link ID      Ref-lstype  Ref-LSID
1.1.1.1         1429     0x80000006 0            0x2001      0
1.1.1.1         1429     0x80000002 4096         0x2002      4

        Type-5 AS External Link States

ADV Router      Age      Seq#      Prefix
2.2.2.2         46       0x80000003 2001:DB8:0:3::/64
2.2.2.2         46       0x80000003 2001:DB8:0:23::/64
```

When examining the routing table on other routers (not the ASBR) in the OSPFv3 domain, by default the redistributed routes will have an administrative distance of 110 and a code of OE2, as shown in Example 17-24. If the metric type is changed to Type 1, the code would be OE1. In an NSSA or totally NSSA area, the redistributed routes would be listed as ON1 or ON2.

Example 17-24 *Verifying OSPFv3 Redistributed Routes*

```

R1#show ipv6 route
IPv6 Routing Table - default - 9 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
        B - BGP, R - RIP, H - NHRP, I1 - ISIS L1
        I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary, D - EIGRP
        EX - EIGRP external, ND - ND Default, NDp - ND Prefix, DCE - Destination
        Ndr - Redirect, O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1
        OE2 - OSPF ext 2, ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2, l - LISP
C   2001:DB8:0:1::/64 [0/0]
    via GigabitEthernet0/0, directly connected
L   2001:DB8:0:1::1/128 [0/0]
    via GigabitEthernet0/0, receive
OE2 2001:DB8:0:3::/64 [110/10]
    via FE80::C802:AFF:FE88:8, GigabitEthernet1/0
C   2001:DB8:0:12::/64 [0/0]
    via GigabitEthernet1/0, directly connected
L   2001:DB8:0:12::1/128 [0/0]
    via GigabitEthernet1/0, receive
C   2001:DB8:0:14::/64 [0/0]
    via FastEthernet3/0, directly connected
L   2001:DB8:0:14::1/128 [0/0]
    via FastEthernet3/0, receive
OE2 2001:DB8:0:23::/64 [110/10]
    via FE80::C802:AFF:FE88:8, GigabitEthernet1/0
L   FF00::/8 [0/0]
    via Null0, receive

```

Note that if you are redistributing from BGP into OSPF, EIGRP, or RIP, only External BGP (eBGP) routes will be redistributed by default. If you want Internal BGP (iBGP) routes to be redistributed, in router BGP configuration mode, you must issue the **bgp redistribute-internal** command.

Troubleshooting Redistribution into BGP

**Key
Topic**

When redistributing into BGP for IPv4, you have the same options found with RIP and EIGRP. You can apply a metric with the **metric** keyword or a route map with the **route-map** keyword. If you are redistributing OSPF into BGP, as shown in Example 17-25, you will also have the option to specify the **match** option, which allows you to match just **internal**, just **external**, just **nssa-external** routes, or a combination of them. With BGP, only internal OSPF routes will be redistributed by default. If you want external OSPF routes to be redistributed, you have to indicate so during redistribution.

The **metric** keyword is not required because BGP will use the IGP metric by default. If the wrong route map is applied, or there is an error within the route map, routes will not be redistributed properly.

Example 17-25 *BGP for IPv4 Redistribution Options*

```
R1(config)#router bgp 65001
R1(config-router)#address-family ipv4 unicast
R1(config-router-af)#redistribute ospf 1 ?
    match      Redistribution of OSPF routes
    metric      Metric for redistributed routes
    route-map   Route map reference
    vrf         VPN Routing/Forwarding Instance
    <cr>
```

With BGP for IPv6, you have the same **match**, **metric**, and **route-map** keywords, in addition to the **include-connected** keyword. By default, with BGP for IPv4, the networks of the local interfaces participating in the routing protocol that is being redistributed on the border router will be redistributed as well. However, with BGP for IPv6, they will not. Therefore, if you want to redistribute the networks associated with the local interfaces participating in the routing process being redistributed into BGP for IPv6, you need to use the **include-connected** keyword, as shown in Example 17-26.

Example 17-26 *BGP for IPv6 Redistribution Options*

```
R1(config)#router bgp 65001
R1(config-router)#address-family ipv6 unicast
R1(config-router-af)#redistribute ospf 1 ?
    include-connected Include connected
    match              Redistribution of OSPF routes
    metric              Metric for redistributed routes
    route-map          Route map reference
    <cr>
```

Using the commands **show ip protocols** and **show ipv6 protocols**, you can verify which protocols are being redistributed into the BGP routing process, as shown in Example 17-27.

Example 17-27 *Verifying Protocols Being Redistributed into BGP*

```
R2#show ip protocols
...output omitted...
Routing Protocol is "bgp 65500"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  IGP synchronization is disabled
  Automatic route summarization is disabled
  Redistributing: ospf 1 (internal)

  Neighbor(s):
    Address          FiltIn FiltOut DistIn DistOut Weight RouteMap
    10.1.1.23.3
```

```

Maximum path: 1
Routing Information Sources:
  Gateway         Distance      Last Update
Distance: external 20 internal 200 local 200

R2#show ipv6 protocols
...output omitted...
IPv6 Routing Protocol is "bgp 65500"
IGP synchronization is disabled
Redistribution:
  Redistributing protocol ospf 1 (internal) include-connected
Neighbor(s):
  Address                               FiltIn FiltOut Weight RoutemapIn RoutemapOut
2001:DB8:0:23::3

```

In the BGP table, redistributed routes appear with a question mark (?) under the Path column, as shown in Example 17-28.

Example 17-28 Verifying Redistributed Routes in the BGP Table

```

R2#show bgp all
For address family: IPv4 Unicast

BGP table version is 4, local router ID is 203.0.113.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network          Next Hop           Metric LocPrf Weight Path
*>  10.1.1.0/24      10.1.12.1             2         32768 ?
*>  10.1.12.0/24     0.0.0.0               0         32768 ?
*>  10.1.14.0/24     10.1.12.1             2         32768 ?

For address family: IPv6 Unicast

BGP table version is 4, local router ID is 203.0.113.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network          Next Hop           Metric LocPrf Weight Path
*>  2001:DB8:0:1::/64
                                   ::              2         32768 ?

```

```
*> 2001:DB8:0:12::/64
      :: 0 32768 ?
*> 2001:DB8:0:14::/64
      :: 2 32768 ?
```

For address family: IPv4 Multicast

For address family: MVPNv4 Unicast

Troubleshooting Redistribution with Route Maps

When applying a route map to the **redistribution** command, you have a few extra items to verify during the troubleshooting process:



- Is the correct route map applied?
- Is permit or deny specified for the sequence, and is it correct? A permit sequence indicates that what is matched will be redistributed. A deny sequence indicates that what is matched will not be redistributed.
- If there is an access list or prefix list being used in the **match** statement, you need to verify that they are correct using the **show {ip|ipv6} access-list** command or the **show {ip|ipv6} prefix-list** command.
- If there are **set** statements, you need to verify that the correct values have been specified to accomplish the desired goal.
- If a route does not match any of the **match** statements in any of the sequences, it will fall into the implicit deny sequence at the end of the route map and not be redistributed.
- If a route map is attached to the **redistribution** command but that route map does not exist, none of the routes will be redistributed.

Redistribution Trouble Tickets

This section presents various trouble tickets relating to the topics discussed earlier in the chapter. The purpose of these trouble tickets is to give a process that you can follow when troubleshooting in the real world or in an exam environment. All trouble tickets in this section are based on the topology depicted in Figure 17-4.

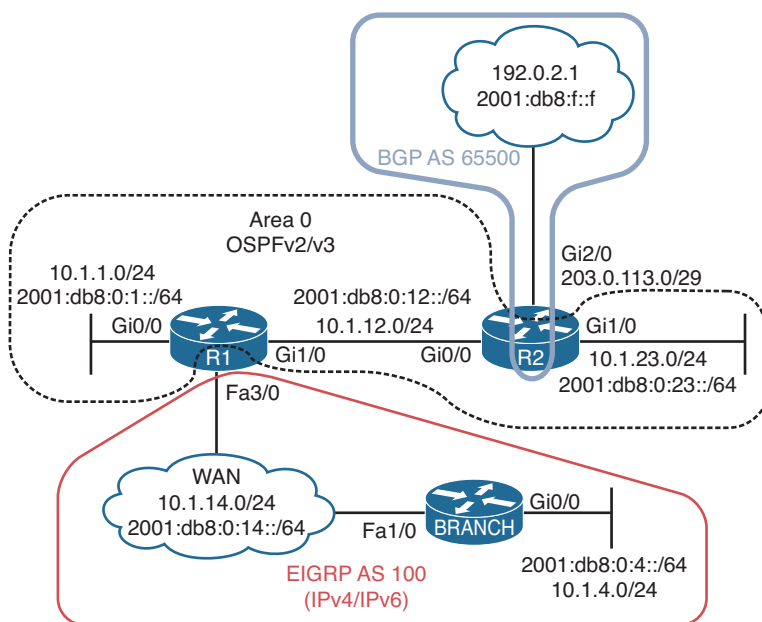


Figure 17-4 *Redistribution Trouble Tickets Topology*

Trouble Ticket 17-1

Problem: Users in the IPv4 Branch site indicate that they are not able to access any resources outside of the Branch office.

On Branch the first thing you check (using the **show ip route** command) is the routing table to see which routes Branch knows, as shown in Example 17-29. The output indicates that Branch only knows about connected and local routes.

Example 17-29 *Verifying the Routing Table on Branch*

```
Branch#show ip route
...output omitted...
    10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C       10.1.4.0/24 is directly connected, GigabitEthernet0/0
L       10.1.4.4/32 is directly connected, GigabitEthernet0/0
C       10.1.14.0/24 is directly connected, FastEthernet1/0
L       10.1.14.4/32 is directly connected, FastEthernet1/0
```

You decide that an EIGRP neighbor relationship might not have been formed with R1. Therefore, you issue the **show ip eigrp neighbors** command on Branch to confirm. As shown in Example 17-30, the device with an IP address of 10.1.14.1 has formed an adjacency with branch. Using the **show cdp neighbors detail** command reveals that the IP address belongs to R1 as shown in the same example.

Example 17-30 *Verifying EIGRP Neighbors on Branch*

```
Branch#show ip eigrp neighbors
EIGRP-IPv4 VR(TSHOOT) Address-Family Neighbors for AS(100)
H   Address                Interface                Hold Uptime    SRTT    RTO  Q  Seq
                               (sec)           (ms)          Cnt  Num
0   10.1.14.1              Fa1/0                12 01:40:12    62   372  0   6

Branch#show cdp neighbors detail
-----
Device ID: R1
Entry address(es):
  IP address: 10.1.14.1
  IPv6 address: 2001:DB8:0:14::1 (global unicast)
  IPv6 address: FE80::C801:AFF:FE88:54 (link-local)
...output omitted...
```

Because R1 and Branch are neighbors, but Branch is not learning any routes from R1, you decide to check whether there are any incoming route filters configured on Branch with the **show ip protocols** command. The output of Example 17-31 shows that there are no route filters.

Example 17-31 *Verifying Route Filters on Branch*

```
Branch#show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "eigrp 100"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  ...output omitted...
```

Next, you decide to check for outbound route filters on R1 using **show ip protocols**. As shown in Example 17-32, there are no route filters.

Example 17-32 *Verifying Route Filters on R1*

```
R1#show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "eigrp 100"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  Redistributing: ospf 1
  EIGRP-IPv4 Protocol for AS(100)
```

```
Metric weight K1=1, K2=0, K3=1, K4=0, K5=0
NSF-aware route hold timer is 240
...output omitted...
```

Because Figure 17-4 shows that R1 is a boundary router performing redistribution, you shift your attention over to R1's redistribution configuration to make sure that the OSPF routes are being redistributed into EIGRP. In Example 17-33, the output of **show ip protocols** indicates that OSPF process 1 is being redistributed into EIGRP autonomous system 100. However, so far all your troubleshooting efforts are indicating that Branch is not learning any redistributed routes.

Example 17-33 *Verifying OSPF Is Being Redistributed into EIGRP*

```
R1#show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "eigrp 100"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  Redistributing: ospf 1
  EIGRP-IPv4 Protocol for AS(100)
    Metric weight K1=1, K2=0, K3=1, K4=0, K5=0
    NSF-aware route hold timer is 240
  ...output omitted...
```

You now issue the **show ip eigrp topology** command on R1. This will confirm if routes are truly being redistributed from OSPF into EIGRP. As shown in Example 17-34, none of the OSPF routes are being redistributed into the EIGRP autonomous system.

Example 17-34 *Verifying Redistributed Routes are in the EIGRP Topology Table*

```
R1#show ip eigrp topology
EIGRP-IPv4 Topology Table for AS(100)/ID(10.1.14.1)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 10.1.14.0/24, 1 successors, FD is 28160
   via Connected, FastEthernet3/0
P 10.1.4.0/24, 1 successors, FD is 28416
   via 10.1.14.4 (28416/2816), FastEthernet3/0
```

You recall that for routes to be redistributed they have to be in the routing table. Therefore, on R1 you issue the **show ip route** command, as shown in Example 17-35, and confirm that there are routes in the routing table that should be redistributed.

Example 17-35 *Verifying Routes to be Redistributed Are in the Routing Table*

```

R1#show ip route
...output omitted...
    10.0.0.0/8 is variably subnetted, 8 subnets, 2 masks
C       10.1.1.0/24 is directly connected, GigabitEthernet0/0
L       10.1.1.1/32 is directly connected, GigabitEthernet0/0
D       10.1.4.0/24 [90/28416] via 10.1.14.4, 02:05:59, FastEthernet3/0
C       10.1.12.0/24 is directly connected, GigabitEthernet1/0
L       10.1.12.1/32 is directly connected, GigabitEthernet1/0
C       10.1.14.0/24 is directly connected, FastEthernet3/0
L       10.1.14.1/32 is directly connected, FastEthernet3/0
O       10.1.23.0/24 [110/2] via 10.1.12.2, 02:02:11, GigabitEthernet1/0
    192.0.2.0/32 is subnetted, 1 subnets
O E2    192.0.2.1 [110/1] via 10.1.12.2, 01:03:22, GigabitEthernet1/0

```

Next you review the **redistribute** command configured on R1 for the EIGRP process with the **show run | section router eigrp** command, as shown in Example 17-36. You notice that there is the command **redistribute ospf 1**; however, you quickly realize that the metric is missing. The metric is mandatory with EIGRP. If you fail to specify one, either with the **default-metric** command, the **metric** command, or in a route map, the routes to be redistributed will be unreachable and not redistributed. You have located the issue.

Example 17-36 *Verifying the redistribute Command on R1*

```

R1#show run | section router eigrp
router eigrp 100
 network 10.1.14.1 0.0.0.0
 redistribute ospf 1
ipv6 router eigrp 100
 redistribute ospf 1 metric 100000 100 255 1 1500 include-connected

```

To solve the issue, you reissue the **redistribute ospf 1** command with the metric values of **100000 100 255 1 1500**. You then issue the **show ip eigrp topology** command, as shown in Example 17-37, and confirm that routes are now redistributed.

Example 17-37 *Verifying Routes to Be Redistributed Are in the R1 Topology Table*

```

R1#show ip eigrp topology
EIGRP-IPv4 Topology Table for AS(100)/ID(10.1.14.1)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 10.1.12.0/24, 1 successors, FD is 51200
   via Redistributed (51200/0)
P 10.1.14.0/24, 1 successors, FD is 28160
   via Connected, FastEthernet3/0
P 10.1.23.0/24, 1 successors, FD is 51200
   via Redistributed (51200/0)

```

```

P 10.1.4.0/24, 1 successors, FD is 28416
    via 10.1.14.4 (28416/2816), FastEthernet3/0
P 192.0.2.1/32, 1 successors, FD is 51200
    via Redistributed (51200/0)
P 10.1.1.0/24, 1 successors, FD is 51200
    via Redistributed (51200/0)

```

The **show ip route** command on Branch, as shown in Example 17-38, allows you to conclude that the problem is solved, because there are now external EIGRP routes learned by Branch and users can successfully connect to resources outside of the Branch office.

Example 17-38 Verifying Routes to Be Redistributed Are in the Branch Routing Table

```

Branch#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

    10.0.0.0/8 is variably subnetted, 7 subnets, 2 masks
D EX    10.1.1.0/24 [170/614400] via 10.1.14.1, 00:02:58, FastEthernet1/0
C        10.1.4.0/24 is directly connected, GigabitEthernet0/0
L        10.1.4.4/32 is directly connected, GigabitEthernet0/0
D EX    10.1.12.0/24 [170/614400] via 10.1.14.1, 00:02:58, FastEthernet1/0
C        10.1.14.0/24 is directly connected, FastEthernet1/0
L        10.1.14.4/32 is directly connected, FastEthernet1/0
D EX    10.1.23.0/24 [170/614400] via 10.1.14.1, 00:02:58, FastEthernet1/0
        192.0.2.0/32 is subnetted, 1 subnets
D EX    192.0.2.1 [170/614400] via 10.1.14.1, 00:02:58, FastEthernet1/0

```

Trouble Ticket 17-2

Problem: Users in the 10.1.23.0/24 network indicate that they are not able to access resources in the 10.1.4.0/24 network.

You begin troubleshooting by verifying the problem on R2. You issue a ping to 10.1.4.4 from 10.1.23.2, but it fails, as shown in Example 17-39. Because R2 is not able to ping the destination network, you confirm that the clients in 10.1.23.0/24 are not able to connect with resources in 10.1.4.0/24.

Example 17-39 *Verifying the Problem from R2*

```
R2#ping 10.1.4.4 source 10.1.23.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.4.4, timeout is 2 seconds:
Packet sent with a source address of 10.1.23.2
.....
Success rate is 0 percent (0/5)
```

On R2, you decide to issue a **traceroute** to help identify where the issue might be. The trace to 10.1.4.4 from 10.1.23.2, as shown in Example 17-40, is headed toward 203.0.113.2, which is out interface Gig2/0, as confirmed in the output of **show ip interface brief** in Example 17-41.

Example 17-40 *Issuing a Trace to Identify Where the Issue Might Be*

```
R2#traceroute 10.1.4.4 source 10.1.23.2
Type escape sequence to abort.
Tracing the route to 10.1.4.4
VRF info: (vrf in name/id, vrf out name/id)
 1 203.0.113.2 28 msec 44 msec 32 msec
 2 * * *
...output omitted...
```

Example 17-41 *Verifying Interface IP Addresses*

```
R2#show ip interface brief
```

Interface	IP-Address	OK?	Method	Status	Protocol
Ethernet0/0	unassigned	YES	NVRAM	administratively down	down
GigabitEthernet0/0	10.1.12.2	YES	NVRAM	up	up
GigabitEthernet1/0	10.1.23.2	YES	NVRAM	up	up
GigabitEthernet2/0	203.0.113.1	YES	NVRAM	up	up

Next you decide to issue the **show ip route 10.1.4.4** command on R2, and the result, as shown in Example 17-42, is that the subnet is not in the table.

Example 17-42 *Verifying the Route on R2*

```
R2#show ip route 10.1.4.4
% Subnet not in table
```

You shift your attention over to R1 and issue the **show ip route 10.1.4.4** command, as shown in Example 17-43, and the result indicates that 10.1.4.4 is reachable using EIGRP out interface Fast Ethernet 3/0. In addition, based on the topology, it should be redistributed into the OSPF process for the OSPF domain to have routes to it. Based on Example 17-43, it is being redistributed into OSPF process 1.

Example 17-43 *Verifying the Route on R1*

```

R1#show ip route 10.1.4.4
Routing entry for 10.1.4.0/24
  Known via "eigrp 100", distance 90, metric 28416, type internal
  Redistributing via eigrp 100, ospf 1
  Last update from 10.1.14.4 on FastEthernet3/0, 2d14h ago
  Routing Descriptor Blocks:
    * 10.1.14.4, from 10.1.14.4, 2d14h ago, via FastEthernet3/0
      Route metric is 28416, traffic share count is 1
      Total delay is 110 microseconds, minimum bandwidth is 100000 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 1

```

You double-check the OSPF database on R1, as shown in Example 17-44, and notice that 10.1.4.0 is not listed as an External Type 5 LSA. This means that it is not being successfully redistributed into the OSPF process.

Example 17-44 *Verifying the Route on R1*

```

R1#show ip ospf database

        OSPF Router with ID (10.1.14.1) (Process ID 1)

        Router Link States (Area 0)

Link ID        ADV Router    Age         Seq#          Checksum Link count
10.1.14.1      10.1.14.1     1698        0x8000007D   0x0064CD 2
203.0.113.1    203.0.113.1   1274        0x80000084   0x005972 2

        Net Link States (Area 0)

Link ID        ADV Router    Age         Seq#          Checksum
10.1.12.2      203.0.113.1   1274        0x8000007C   0x0010FE

        Type-5 AS External Link States

Link ID        ADV Router    Age         Seq#          Checksum Tag
192.0.2.1      203.0.113.1   1274        0x8000007C   0x00FD38 0

```

You issue the **show run | section router ospf** command on R1 to verify the OSPF configuration on R1. As shown in Example 17-45, the **redistribute eigrp 100** command is listed in the configuration. However, as you discovered earlier, the EIGRP routes are not being redistributed. You double-check to make sure that the correct EIGRP autonomous system is being redistributed by issuing the **show run | section router eigrp** command, as shown in Example 17-46. This output confirms that the correct EIGRP autonomous system is being redistributed.

Example 17-45 *Verifying OSPF Configuration on R1*

```
R1#show run | section router ospf
router ospf 1
  redistribute eigrp 100
  network 10.1.1.1 0.0.0.0 area 0
  network 10.1.12.1 0.0.0.0 area 0
ipv6 router ospf 1
  redistribute eigrp 100 include-connected
```

Example 17-46 *Verifying EIGRP Configuration on R1*

```
R1#show run | section router eigrp
router eigrp 100
  network 10.1.14.1 0.0.0.0
  redistribute ospf 1 metric 100000 100 255 1 1500
ipv6 router eigrp 100
  redistribute ospf 1 metric 100000 100 255 1 1500 include-connected
```

After some thought, you realize that the 10.1.4.0/24 network is a classless network and that the current **redistribute eigrp 100** command will only redistribute classful networks. You need to add the **subnets** keyword to the **redistribute** command, as shown in Example 17-47, to redistribute classless networks. Issuing the **show ip ospf database** command in Example 17-48 confirms that the OSPF database is now learning the EIGRP route 10.1.4.0/24.

Example 17-47 *Adding subnets Keyword to Redistribute Command*

```
R1#config t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#router ospf 1
R1(config-router)#redistribute eigrp 100 subnets
```

Example 17-48 *Verifying That the 10.1.4.0 Route Is in the OSPF Database on R1*

```
R1#show ip ospf database

      OSPF Router with ID (10.1.14.1) (Process ID 1)

      Router Link States (Area 0)

Link ID        ADV Router    Age          Seq#          Checksum Link count
10.1.14.1      10.1.14.1     339         0x8000007E   0x0062CE 2
203.0.113.1    203.0.113.1   1923        0x80000084   0x005972 2

      Net Link States (Area 0)

Link ID        ADV Router    Age          Seq#          Checksum
```

10.1.12.2	203.0.113.1	1923	0x8000007C	0x0010FE
Type-5 AS External Link States				
Link ID	ADV Router	Age	Seq#	Checksum Tag
10.1.4.0	10.1.14.1	17	0x80000001	0x006215 0
10.1.14.0	10.1.14.1	17	0x80000001	0x00F379 0
192.0.2.1	203.0.113.1	1923	0x8000007C	0x00FD38 0

Next you visit R2 and issue the **show ip route 10.1.4.4** command and confirm that it has been added, as shown in Example 17-49.

Example 17-49 Verifying That R2 Now Knows About the 10.1.4.0 Network

```
R2#show ip route 10.1.4.4
Routing entry for 10.1.4.0/24
  Known via "ospf 1", distance 110, metric 20, type extern 2, forward metric 1
  Redistributing via bgp 65500
  Advertised by bgp 65500 match internal external 1 & 2
  Last update from 10.1.12.1 on GigabitEthernet0/0, 00:04:52 ago
  Routing Descriptor Blocks:
    * 10.1.12.1, from 10.1.14.1, 00:04:52 ago, via GigabitEthernet0/0
      Route metric is 20, traffic share count is 1
```

Finally, you confirm that the problem is solved with a ping from 10.1.23.2 to 10.1.4.4, and it is successful, as shown in Example 17-50.

Example 17-50 Successful Ping

```
R2#ping 10.1.4.4 source 10.1.23.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.4.4, timeout is 2 seconds:
Packet sent with a source address of 10.1.23.2
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 36/55/72 ms
```

Trouble Ticket 17-3

Problem: IPv6 users in the 2001:db8:0:4::/64 network report that they are not able to access resources in the 2001:db8:0:1::/64 network.

You begin troubleshooting by confirming the problem on Branch. As shown in Example 17-51, the ping from 2001:db8:0:4:4 to 2001:db8:0:1:1 fails.

Example 17-51 Confirming the Problem with a Ping

```
Branch#ping 2001:db8:0:1::1 source 2001:db8:0:4::4
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:0:1::1, timeout is 2 seconds:
```

```
Packet sent with a source address of 2001:DB8:0:4::4
.....
Success rate is 0 percent (0/5)
```

While gathering further information, you decide to ping an IPv6 address in the 2001:db8:0:23::/64 network. As shown in Example 17-52, the ping is successful. Therefore, you conclude that only some of the routes in the IPv6 OSPF domain are being redistributed into the EIGRP for IPv6 domain. You issue the **show ipv6 route** command on Branch, as shown in Example 17-53, and the output confirms that only two external routes are being learned by Branch: 2001:db8:0:23::/64 and 2001:db8:f::/64.

Example 17-52 *Gathering More Information with a Ping*

```
Branch#ping 2001:db8:0:23::2 source 2001:db8:0:4::4
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:0:23::2, timeout is 2 seconds:
Packet sent with a source address of 2001:DB8:0:4::4
!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/47/120 ms
```

Example 17-53 *Verifying Routes on Branch*

```
Branch#show ipv6 route
...output omitted...
C    2001:DB8:0:4::/64 [0/0]
    via GigabitEthernet0/0, directly connected
L    2001:DB8:0:4::4/128 [0/0]
    via GigabitEthernet0/0, receive
C    2001:DB8:0:14::/64 [0/0]
    via FastEthernet1/0, directly connected
L    2001:DB8:0:14::4/128 [0/0]
    via FastEthernet1/0, receive
EX 2001:DB8:0:23::/64 [170/614400]
    via FE80::C801:AFF:FE88:54, FastEthernet1/0
EX 2001:DB8:F::/64 [170/614400]
    via FE80::C801:AFF:FE88:54, FastEthernet1/0
L    FF00::/8 [0/0]
    via Null0, receive
```

Based on the information you have gathered, you decide to check whether redistribution is being performed on R1. You issue the **show ipv6 protocols** command on R1, as shown in Example 17-54. In the output, you focus on the EIGRP section and review the redistribution information. It clearly indicates that redistribution from OSPF process 1 into EIGRP autonomous system 100 is occurring. In addition, the metric values have been applied, which are mandatory for EIGRP, and internal and external routes are being redistributed. You think that a route map might be applied that is controlling the routes that are being redistributed. However, you notice that a route map is not listed under the *Redistribution* section of the **show ipv6 protocols** command. Therefore, that is not the issue.

Example 17-54 *Verifying IPv6 Redistribution on R1*

```

R1#show ipv6 protocols
IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "ND"
IPv6 Routing Protocol is "eigrp 100"
EIGRP-IPv6 Protocol for AS(100)
  Metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  NSF-aware route hold timer is 240
  Router-ID: 10.1.14.1
  Topology : 0 (base)
    Active Timer: 3 min
    Distance: internal 90 external 170
    Maximum path: 16
    Maximum hopcount 100
    Maximum metric variance 1

  Interfaces:
    FastEthernet3/0
  Redistribution:
    Redistributing protocol ospf 1 with metric 100000 100 255 1 1500 (internal,
external 1 & 2, nssa-external 1 & 2)
IPv6 Routing Protocol is "ospf 1"
  Router ID 10.1.14.1
  Autonomous system boundary router
  Number of areas: 1 normal, 0 stub, 0 nssa
  Interfaces (Area 0):
    GigabitEthernet1/0
    GigabitEthernet0/0
  Redistribution:
    Redistributing protocol eigrp 100 include-connected

```

On R1, you issue the **show ipv6 eigrp topology** command to confirm whether the routes are being redistributed into EIGRP from OSPF. As shown in Example 17-55, only the routes 2001:db8:0:1::/64 and 2001:db8:f::/64 are being redistributed.

Example 17-55 *Reviewing R1's EIGRP Topology*

```

R1#show ipv6 eigrp topology
EIGRP-IPv6 Topology Table for AS(100)/ID(10.1.14.1)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 2001:DB8:0:4::/64, 1 successors, FD is 28416
   via FE80::C800:CFF:FEE4:1C (28416/2816), FastEthernet3/0
P 2001:DB8:F::/64, 1 successors, FD is 51200
   via Redistributed (51200/0)
P 2001:DB8:0:14::/64, 1 successors, FD is 28160
   via Connected, FastEthernet3/0

```

```
P 2001:DB8:0:23::/64, 1 successors, FD is 51200
    via Redistributed (51200/0)
```

You check the output of **show ipv6 route** on R1 and note that 2001:db8:0:1::/64 and 2001:db8:0:12::/64 are both in R1's routing table as connected routes, as shown in Example 17-56. Therefore, for them to be redistributed, they either have to be redistributed as connected routes or participating in the OSPF process, because R1 is configured to redistribute OSPF into EIGRP. Therefore, on R1, you issue the **show ipv6 ospf interface** command, as shown in Example 17-57, and confirm that both Gig0/0 and Gig1/0 are participating in the OSPF process. However, based on your information gathering so far, you have determined that the routes are still not being redistributed.

Example 17-56 Reviewing R1's IPv6 Routing Table

```
R1#show ipv6 route
...output omitted...
C 2001:DB8:0:1::/64 [0/0]
    via GigabitEthernet0/0, directly connected
L 2001:DB8:0:1::1/128 [0/0]
    via GigabitEthernet0/0, receive
D 2001:DB8:0:4::/64 [90/28416]
    via FE80::C800:CFF:FEE4:1C, FastEthernet3/0
C 2001:DB8:0:12::/64 [0/0]
    via GigabitEthernet1/0, directly connected
L 2001:DB8:0:12::1/128 [0/0]
    via GigabitEthernet1/0, receive
C 2001:DB8:0:14::/64 [0/0]
    via FastEthernet3/0, directly connected
L 2001:DB8:0:14::1/128 [0/0]
    via FastEthernet3/0, receive
O 2001:DB8:0:23::/64 [110/2]
    via FE80::C802:AFF:FE88:8, GigabitEthernet1/0
OE2 2001:DB8:F::/64 [110/1]
    via FE80::C802:AFF:FE88:8, GigabitEthernet1/0
L FF00::/8 [0/0]
    via Null0, receive
```

Example 17-57 Reviewing R1's IPv6 OSPF Interfaces

```
R1#show ipv6 ospf interface brief
```

Interface	PID	Area	Intf ID	Cost	State	Nbrs	F/C
Gil/0	1	0	4	1	BDR	1/1	
Gi0/0	1	0	3	1	DR	0/0	

At this point, you recall that IPv6 redistribution behaves differently than IPv4 redistribution with directly connected networks. IPv6 directly connected networks are not redistributed by default. You need to use the **include-connected** keyword to force the directly

connected networks to be redistributed. Reviewing the *Redistribution* section in the `show ipv6 protocols` output of Example 17-54 again confirms that the `include-connected` keyword was not included in the command.

On R1, you issue the command `redistribute ospf 1 metric 100000 100 255 1 1500 include-connected` in IPv6 EIGRP configuration mode, as shown in Example 17-58, to fix the issue.

Example 17-58 *Modifying the redistribute Command*

```
R1#config t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#ipv6 router eigrp 100
R1(config-rtr)#redistribute ospf 1 metric 100000 100 255 1 1500 include-connected
```

You reissue the `show ipv6 protocols` command and the `show ipv6 eigrp topology` command and confirm that the directly connected routes are now being redistributed, as shown in Example 17-59.

Example 17-59 *Verifying That Routes Are Redistributed After Changes*

```
R1#show ipv6 protocols
IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "ND"
IPv6 Routing Protocol is "eigrp 100"
EIGRP-IPv6 Protocol for AS(100)
  Metric weight K1=1, K2=0, K3=1, K4=0, K5=0
...output omitted...
Redistribution:
  Redistributing protocol ospf 1 with metric 100000 100 255 1 1500 (internal,
external 1 & 2, nssa-external 1 & 2) include-connected
...output omitted...
R1#show ipv6 eigrp topology
EIGRP-IPv6 Topology Table for AS(100)/ID(10.1.14.1)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 2001:DB8:0:4::/64, 1 successors, FD is 28416
   via FE80::C800:CFF:FEE4:1C (28416/2816), FastEthernet3/0
P 2001:DB8:0:1::/64, 1 successors, FD is 51200
   via Redistributed (51200/0)
P 2001:DB8:F::/64, 1 successors, FD is 51200
   via Redistributed (51200/0)
P 2001:DB8:0:14::/64, 1 successors, FD is 28160
   via Connected, FastEthernet3/0
P 2001:DB8:0:12::/64, 1 successors, FD is 51200
   via Redistributed (51200/0)
P 2001:DB8:0:23::/64, 1 successors, FD is 51200
   via Redistributed (51200/0)
```


Going back to Branch, you issue the **show ipv6 route** command and notice that there is an entry in the routing table for 2001:db8:0:1::/64 and 2001:db8:0:12::/64 now, as shown in Example 17-60.

Example 17-60 *Verifying That Routes Are Learned By Branch*

```
Branch#show ipv6 route
IPv6 Routing Table - default - 9 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
        B - BGP, R - RIP, H - NHRP, I1 - ISIS L1
        I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary, D - EIGRP
        EX - EIGRP external, ND - ND Default, NDp - ND Prefix, DCE - Destination
        NDr - Redirect, O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1
        OE2 - OSPF ext 2, ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2, l - LISP
EX 2001:DB8:0:1::/64 [170/614400]
    via FE80::C801:AFF:FE88:54, FastEthernet1/0
C  2001:DB8:0:4::/64 [0/0]
    via GigabitEthernet0/0, directly connected
L  2001:DB8:0:4::4/128 [0/0]
    via GigabitEthernet0/0, receive
EX 2001:DB8:0:12::/64 [170/614400]
    via FE80::C801:AFF:FE88:54, FastEthernet1/0
C  2001:DB8:0:14::/64 [0/0]
    via FastEthernet1/0, directly connected
L  2001:DB8:0:14::4/128 [0/0]
    via FastEthernet1/0, receive
EX 2001:DB8:0:23::/64 [170/614400]
    via FE80::C801:AFF:FE88:54, FastEthernet1/0
EX 2001:DB8:F::/64 [170/614400]
    via FE80::C801:AFF:FE88:54, FastEthernet1/0
L  FF00::/8 [0/0]
    via Null0, receive
```

You verify that the problem is solved with a ping from Branch at 2001:db8:0:4::4 to 2001:db8:0:1::1, as shown in Example 17-61. The ping is successful, and the problem is solved.

Example 17-61 *Verifying That the Problem Is Solved with a Successful Ping*

```
Branch#ping 2001:db8:0:1::1 source 2001:db8:0:4::4
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:0:1::1, timeout is 2 seconds:
Packet sent with a source address of 2001:DB8:0:4::4
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/32/44 ms
```

Trouble Ticket 17-4

Problem: A junior administrator has approached you asking for help. He claims that users in BGP autonomous system 65500 are unable to access IPv4 resources in the EIGRP for IPv4 autonomous system 100. However, they can access resources in the OSPFv2 domain. Because you do not have access to any routers in BGP autonomous system 65500 (except for R2), he has asked you for help because he does not know what to do.

You start by reviewing Figure 17-4 to confirm which local router is running BGP. It is R2. You issue the **show bgp ipv4 unicast summary** command on R2 to confirm whether R2 has any BGP neighbors. As shown in Example 17-62, 203.0.113.2 is listed as a neighbor, and because the State/PfxRcd column has a number, it is an established neighborhood. To further confirm, you issue the **show bgp ipv4 unicast neighbors | include BGP** command, as shown in Example 17-63, and the output indicates that 203.0.113.2 is an established neighbor.

Example 17-62 Verifying BGP Neighbors

```
R2#show bgp ipv4 unicast summary
BGP router identifier 203.0.113.1, local AS number 65500
BGP table version is 33, main routing table version 33
4 network entries using 576 bytes of memory
4 path entries using 320 bytes of memory
3/3 BGP path/bestpath attribute entries using 408 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 1304 total bytes of memory
BGP activity 28/18 prefixes, 30/20 paths, scan interval 60 secs
```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
203.0.113.2	4	65500	496	500	33	0	0	07:26:42	1

Example 17-63 Verifying Established BGP Neighbor

```
R2#show bgp ipv4 unicast neighbors | include BGP
BGP neighbor is 203.0.113.2, remote AS 65500, internal link
  BGP version 4, remote router ID 192.0.2.1
  BGP state = Established, up for 07:31:19
  BGP table version 33, neighbor version 33/0
  Last reset 07:31:29, due to BGP Notification received of session 1, header syn-
  chronization problems
```

Next you verify whether any routes are being advertised to the neighbor at 203.0.113.2 by issuing the **show bgp ipv4 unicast neighbors 203.0.113.2 advertised-routes** command. In Example 17-64, you can see that three routes are being advertised to 203.0.113.2 from R2. The routes are 10.1.1.0/24, 10.1.12.0/24, and 10.1.23.0/24. Figure 17-4 indicates that the EIGRP networks are 10.1.14.0/24 and 10.1.4.0/24 and that they are not listed as routes being advertised.

Example 17-64 *Verifying Advertised BGP Routes*

```

R2#show bgp ipv4 unicast neighbors 203.0.113.2 advertised-routes
BGP table version is 33, local router ID is 203.0.113.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network          Next Hop           Metric LocPrf Weight Path
* > 10.1.1.0/24      10.1.12.1             2         32768 ?
* > 10.1.12.0/24     0.0.0.0                0         32768 ?
* > 10.1.23.0/24     0.0.0.0                0         32768 ?

Total number of prefixes 3

```

You issue the **show ip protocols** command on R2, as shown in Example 17-65, to verify the BGP configuration. You notice that there are no filters, no distribute lists, or no route maps applied to neighbor 203.0.113.2 that could be preventing routes from being advertised. However, you notice that only OSPF internal routes are being redistributed in the output. You issue the **show ip route** command on R2, as shown in Example 17-66, and confirm that 10.1.4.0/24 and 10.1.14.0/24 are both external OSPF routes. You conclude that the problem is related to BGP not redistributing OSPF external routes.

Example 17-65 *Verifying BGP Configuration with show ip protocols*

```

R2#show ip protocols
*** IP Routing is NSF aware ***
...output omitted...
Routing Protocol is "bgp 65500"

  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  IGP synchronization is disabled
  Automatic route summarization is disabled
  Redistributing: ospf 1 (internal)

Neighbor(s):
  Address          FiltIn FiltOut DistIn DistOut Weight RouteMap
  203.0.113.2
Maximum path: 1
Routing Information Sources:
  Gateway          Distance      Last Update
  203.0.113.2       200           07:54:48
Distance: external 20 internal 200 local 200

```

Example 17-66 *Verifying IPv4 Routes on R2*

```

R2#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is 203.0.113.2 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 203.0.113.2
      10.0.0.0/8 is variably subnetted, 7 subnets, 2 masks
O      10.1.1.0/24 [110/2] via 10.1.12.1, 4d20h, GigabitEthernet0/0
O E2   10.1.4.0/24 [110/20] via 10.1.12.1, 1d23h, GigabitEthernet0/0
C      10.1.12.0/24 is directly connected, GigabitEthernet0/0
L      10.1.12.2/32 is directly connected, GigabitEthernet0/0
O E2   10.1.14.0/24 [110/20] via 10.1.12.1, 1d23h, GigabitEthernet0/0
C      10.1.23.0/24 is directly connected, GigabitEthernet1/0
L      10.1.23.2/32 is directly connected, GigabitEthernet1/0
      192.0.2.0/32 is subnetted, 1 subnets
B      192.0.2.1 [200/0] via 203.0.113.2, 08:00:48
      203.0.113.0/24 is variably subnetted, 2 subnets, 2 masks
C      203.0.113.0/29 is directly connected, GigabitEthernet2/0
L      203.0.113.1/32 is directly connected, GigabitEthernet2/0

```

On R2, you issue the **show run | section router bgp** command, as shown in Example 17-67, to verify the BGP configuration. Under the IPv4 address family, you notice that the **redistribute ospf 1** command has been issued. However, that only redistributes internal OSPF routes. It does not redistribute OSPF external routes by default.

Example 17-67 *Verifying IPv4 Routes on R2*

```

R2#show run | section router bgp
router bgp 65500
  bgp log-neighbor-changes
  neighbor 2001:DB8:0:A::A remote-as 65500
  neighbor 203.0.113.2 remote-as 65500
  !
  address-family ipv4
    bgp redistribute-internal
    redistribute ospf 1
    no neighbor 2001:DB8:0:A::A activate
    neighbor 203.0.113.2 activate
  
```

```

exit-address-family
!
address-family ipv6
 redistribute ospf 1 match internal external 1 external 2 include-connected
 bgp redistribute-internal
 neighbor 2001:DB8:0:A::A activate
exit-address-family

```

Because the routes are external Type 2 OSPF routes, you issue the command **redistribute ospf 1 match internal external 2** in IPv4 BGP address family configuration mode, as shown in Example 17-68. You then issue the command **show ip protocols** to verify that external Type 2 routes are now being redistributed as well. As shown in Example 17-69, they are.

Example 17-68 *Modifying the redistribute Command in IPv4 Address Family Config Mode*

```

R2#config t
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#router bgp 65500
R2(config-router)#address-family ipv4 unicast
R2(config-router-af)#redistribute ospf 1 match internal external 2

```

Example 17-69 *Verifying Types of OSPF Routes Being Advertised into BGP*

```

R2#show ip protocols
...output omitted...
Routing Protocol is "bgp 65500"

  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  IGP synchronization is disabled
  Automatic route summarization is disabled
  Redistributing: ospf 1 (internal, external 2)

  Neighbor(s):

    Address           FiltIn FiltOut DistIn DistOut Weight RouteMap
    203.0.113.2

Maximum path: 1
Routing Information Sources:
  Gateway         Distance      Last Update
  203.0.113.2         200          1d07h
Distance: external 20 internal 200 local 200

```

You then reissue the **show bgp ipv4 unicast neighbors 203.0.113.2 advertised-routes** command to verify that 10.1.14.0/24 and 10.1.4.0/24 are being advertised in BGP autonomous system 65500. As shown in Example 17-70, they are.

Example 17-70 *Verifying OSPF Routes Are Advertised to BGP Neighbor*

```
R2#show bgp ipv4 unicast neighbors 203.0.113.2 advertised-routes
BGP table version is 35, local router ID is 203.0.113.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

	Network	Next Hop	Metric	LocPrf	Weight	Path
*>	10.1.1.0/24	10.1.12.1	2		32768	?
*>	10.1.4.0/24	10.1.12.1	20		32768	?
*>	10.1.12.0/24	0.0.0.0	0		32768	?
*>	10.1.14.0/24	10.1.12.1	20		32768	?
*>	10.1.23.0/24	0.0.0.0	0		32768	?

```
Total number of prefixes 5
```

Next you pick up the phone and call the administrator of the other routers in BGP autonomous system 65500 and confirm that they can access the resources in EIGRP autonomous system 100. They state that they can; therefore, you have solved the issue.

Troubleshooting Advanced Redistribution Issues

When route redistribution is misconfigured, it can lead to issues such as routing loops and suboptimal routing. Suboptimal routing can lead to users experiencing slow connectivity, and routing loops can lead to no connectivity. This section explains how you can recognize these issues and the options you have to fix them.

Troubleshooting Suboptimal Routing Caused by Redistribution

When redistributing routes from one routing source into another routing source, the original routing source's information is lost when the seed metric is injected at the redistribution point. Therefore, overall network visibility is lost or hidden from the destination routing source. This is not an issue when there is only one point of redistribution between two sources. However, if there are multiple points of redistribution between two sources, as shown in Figure 17-5, the suboptimal path may be chosen to reach routes.

From R1 and R2, the optimal path to reach 192.168.2.0/24 is from R2 because the 1-Gbps link is much faster than the 10-Mbps link. When you perform redistribution on R1 and R2 into EIGRP, EIGRP does not know that the 10-Mbps or the 1-Gbps link exists in the OSPF domain. Therefore, if an inappropriate seed metric is used during redistribution on R1 and R2, the traffic from 10.1.1.0/24 destined for 192.168.2.0/24 may take the suboptimal path through R1. However, realize, according to the EIGRP AS, it is the best path because all it sees is the seed metric and the 1-Gbps and 100-Mbps link in the EIGRP autonomous system. Therefore, if the seed metrics you define are the same on R1 and R2 when you redistribute into EIGRP, the 1-Gbps link in the EIGRP autonomous

system is preferred, and traffic goes to R1. Then R1 sends it across the 10-Mbps link to 192.168.2.0/24, which is suboptimal. It works, but it is suboptimal.

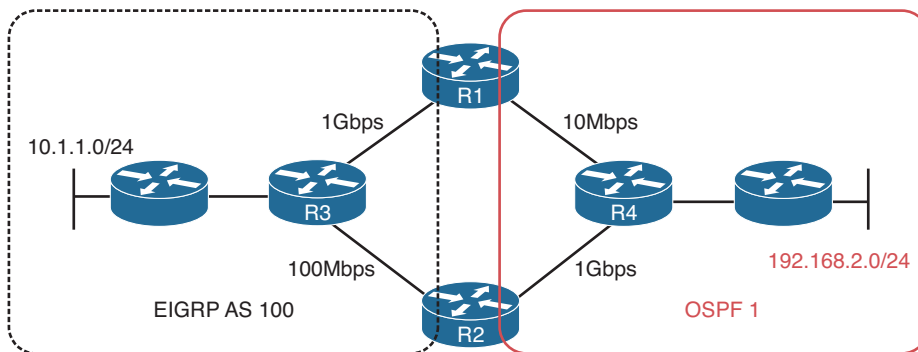


Figure 17-5 *Suboptimal Routing Topology*

You can recognize this issue from a topological diagram in addition to using the **tracert** command. In Figure 17-5, if the result of the traceroute from 10.1.1.0/24 to 192.168.2.0/24 goes through R1, suboptimal routing is occurring because of redistribution.



You can solve this issue by providing different seed metrics on the boundary routers (R1 and R2 in this case) that will ensure a certain path is preferred because it has a lower overall metric. So, R2's EIGRP seed metric would have to be significantly lower than R1's EIGRP seed metric to ensure that R3 chooses the path through R2 even though it is a slower link between R3 and R2 than R3 and R1. The key is to make sure that the traffic avoids the 10-Mbps link.

Going in reverse, when redistributing from EIGRP into OSPF, the redistributed routes will have a default seed metric of 20 and be classified as E2 routes; therefore, the metric will remain as 20 throughout the OSPF domain. At first, you might think that load balancing will occur from R4 to R1 and R2 when sending traffic from 192.168.2.0/24 to 10.1.1.0/24. You would be correct only if the metrics (forward metric) to reach the ASBRs are equal as well as the E2 seed metric. In this case, the forward metrics are not equal. The 10-Mbps link has a much higher cost than the 1-Gbps link. Therefore, all the traffic from 192.168.2.0/24 to 10.1.1.0/24 will go through R2 across the 1-Gbps link (lower metric to reach ASBR) in the OSPF domain. However, if the seed metric was set higher than 20 on R2 and left at 20 on R1, R1 will be used as the path because it now has the lower seed metric, but in this case it would be the suboptimal path. Therefore, if the metric type is E2, you can simply make the preferred ASBR advertise the lowest seed metric to ensure that optimal routing is achieved. If you are using a metric type of E1, the cost of the links within the network are added to the seed metric to come up with the overall cost to reach the destination network. Therefore, if suboptimal routing is occurring, you need to determine which seed metrics are most appropriate with E2 to ensure the optimal path is chosen, or use a metric type of E1 so that internal costs are used with the seed metric to determine the overall cost.

When troubleshooting suboptimal routing caused by redistribution, keep the following in mind:

Key Topic

- Based on the topology, be able to recognize that mutual redistribution is occurring at multiple points in the network.
- Based on the connections, be able to recognize the different speeds of the links.
- Based on the routing protocols in use, be able to identify how the seed metric is determined and how it behaves for the different protocols.
- Based on the business requirements, know how to fix the suboptimal routing by manipulating the metrics on the boundary routers with the **default-metric** command, the **metric** parameter in the **redistribute** command, or within a route map.

Troubleshooting Routing Loops Caused by Redistribution

Key Topic

Examine Figure 17-6. The 10.1.1.0/24 network is redistributed into the EIGRP autonomous system, and then it is redistributed into the OSPF domain on R1 and R2. This does not appear to be an issue; however, it is because of AD. Let's explore what happens.

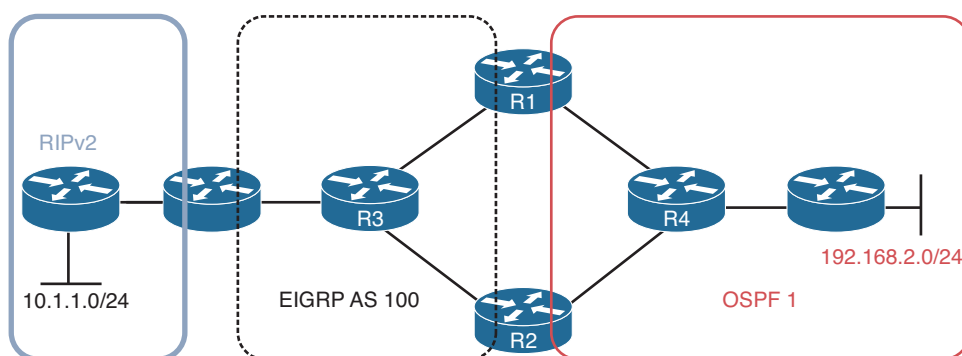


Figure 17-6 Routing Loop Routing Topology

When the 10.1.1.0/24 network is redistributed from RIPv2 into EIGRP autonomous system 100, it is classified as an external route in the EIGRP autonomous system. R1 and R2 place the route in the routing table with the code D EX and an AD of 170, as shown in Figure 17-7.

When R1 and R2 redistribute the 10.1.1.0/24 network in the OSPF domain, by default, the Type 5 LSA is advertising 10.1.1.0/24 as an O E2 route, with an AD of 110, as shown in Figure 17-8. Do not forget that it is flooded through the area. Therefore, R1 will receive R2's LSA and R2 will receive R1's LSA, which creates the problem. Look closely at R1's two entries for 10.1.1.0/24. Which one will be preferred? The OSPF route because it has a lower AD. Therefore, R1 points to R2 to reach 10.1.1.0/24. Look closely at R2's two entries for 10.1.1.0/24. Which one will be preferred? The OSPF route because it has a lower AD. Therefore, R2 points to R1 to reach 10.1.1.0/24.

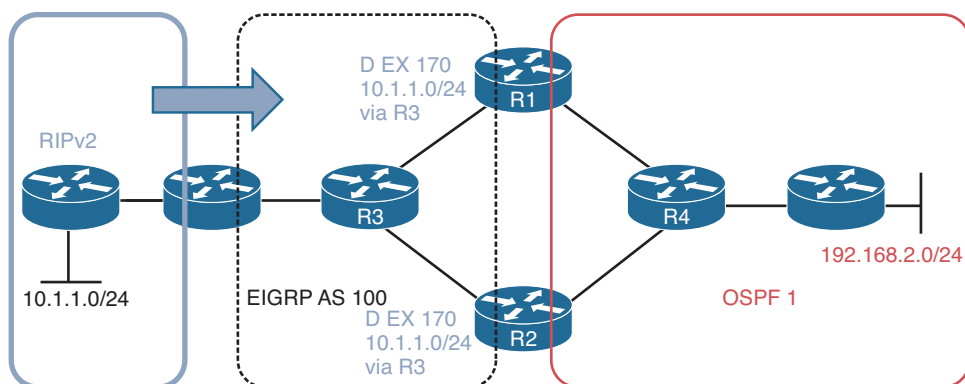


Figure 17-7 Redistributing the RIPv2 Route into the EIGRP Autonomous System

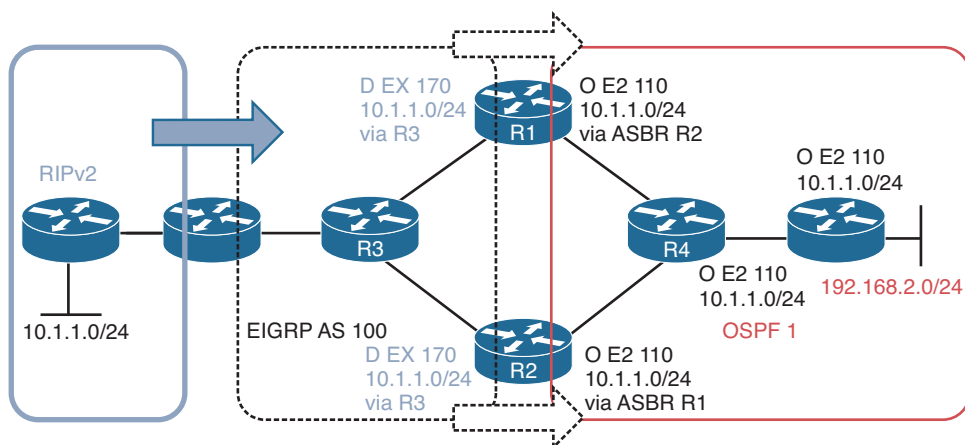


Figure 17-8 Redistributing the RIPv2 Route into the OSPF Domain on R1 and R2

Now when traffic is sent from 192.168.2.0/24 to 10.1.1.0/24, it will bounce back and forth between R1 and R2, which is classified as a routing loop.

However, this scenario gets worse because of how redistribution works. Remember that to redistribute a route from one routing source to another (EIGRP into OSPF), that route must be in the routing table as an entry for the routing source that you are redistributing the route from.

With that in mind, consider Figure 17-8 again. When R1 and R2 originally learned about the network 10.1.1.0/24 from R3, it was an EIGRP external route. There was no other source of information in the routing table at the time for 10.1.1.0/24; therefore, it was considered the best source and installed in the routing table as an EIGRP route. Because redistribution is occurring from EIGRP into OSPF, the 10.1.1.0/24 network is redistributed from the routing table into the OSPF process and advertised. Now, when R1 and R2 learn about the OSPF 10.1.1.0/24 route from each other, they notice that it is a better source of information because the AD is lower (110) than the one for EIGRP (170) currently in the routing table. Therefore, the OSPF route replaces the EIGRP route. What

happens now? Well, the EIGRP route is no longer in the routing table on R1 and R2. It is still in the EIGRP topology table, but not in the routing table. Therefore, the 10.1.1.0/24 network is no longer available for redistribution into OSPF, and therefore, there are no more Type 5 LSAs to advertise. As a result of this, R1 and R2 have to notify the routers in the OSPF domain that 10.1.1.0/24 no longer exists. When this happens, R1 and R2 no longer have the 10.1.1.0/24 network that they learned via OSPF from each other in the routing table. What does this cause? The EIGRP external route 10.1.1.0/24 is reinstalled in the routing table, and because redistribution from EIGRP into OSPF is occurring, the issue repeats all over again. As you can see, the routing table is not stable, because routes are inserted then removed and inserted and removed over and over again. You can see this happening with the **debug ip routing** command, which displays changes as they occur to the routing table.

Let's take this even further, examine Figure 17-9, which shows the 10.1.1.0/24 network being redistributed back into the EIGRP autonomous system when the OSPF route is in the routing table on R1 and R2. Now R3 thinks that 10.1.1.0/24 is reachable via the boundary router between the RIPv2 domain and the EIGRP autonomous system, as well as R1 and R2. So now, additional CPU cycles are being used in addition to memory.

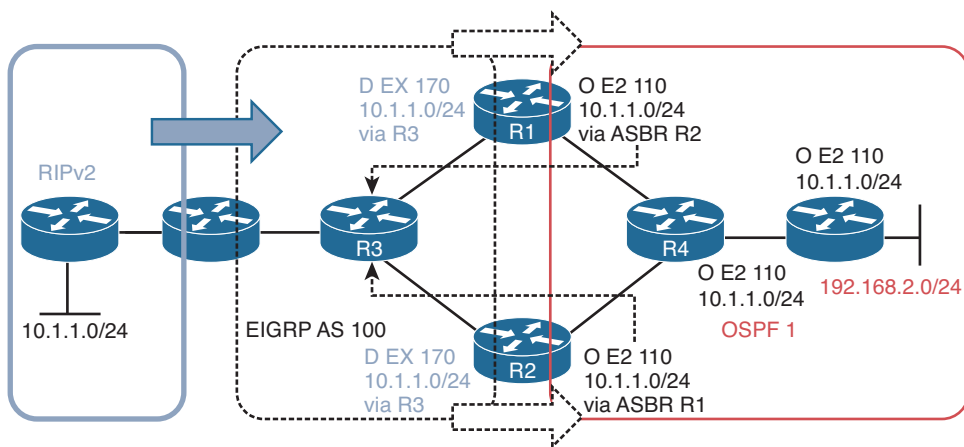


Figure 17-9 Redistributing the RIPv2 Route Back into the EIGRP Autonomous System from OSPF

This is definitely a bad situation to be in. It is recognized through the analysis of a diagram. Notice how we could identify this problem without using any **show** commands. In addition, the symptoms are wide ranging. For example, a user might have a connection from 192.168.2.0/24 to 10.1.1.0/24 for one moment and then the connection is lost, then it is back, then lost, all because the routes are being added and removed over and over again, causing a loop, and then no loop, and so on. Therefore, you need to be able to look at the topology and identify where this type of issue might occur and implement the necessary measures to stop it from happening. Or, if it is happening, identify why it is happening and propose how to fix it.

Remember that this issue was caused by AD; 110 is better than 170. Therefore, you need to either lower the AD of the EIGRP routes on R1 and R2 for 10.1.1.0/24 or increase the

AD of the OSPF learned routes on R1 and R2 for 10.1.1.0/24. Your goal is to make sure that the EIGRP learned route is the preferred route. Regardless of what you choose to do, you need to use the **distance** command on R1 and R2 and specify what the AD will be for the 10.1.1.0/24 network. Because you only want to affect the 10.1.1.0/24 network in this example, you could use an ACL and attach it to the **distance** command to single out the 10.1.1.0/24 network. If you lower the EIGRP AD, it will need to be 109 or lower, and if you decide to increase the OSPF AD, it will need to be 171 or higher.

There is another way to solve this issue. You could attach a distribute list to the OSPF process on R1 and R2. When a distribute list is used with OSPF, it can control what routes are installed in the routing table from the OSPF database. Therefore, if you deny the 10.1.1.0/24 route in the OSPF database from being installed in the routing table with a distribute list, the EIGRP route will be installed in the routing table instead.

And finally, you do not want the routes that are redistributed from EIGRP into OSPF to be redistributed back into the EIGRP autonomous system. This can cause routing issues such as loops, which prevent packets from being correctly delivered to their destination (in addition to wasting CPU and memory resources on various devices in the network). The most robust way to deal with this is route tags. Figure 17-10 shows how R1 and R2 can add a tag (which is just an arbitrary value that can be used to identify the route) when the route is redistributed. This is accomplished with route maps. In this example, when R1 redistributes the 10.1.1.0/24 route into the OSPF domain, it adds a tag of 10. When R2 redistributes the 10.1.1.0/24 route into the OSPF domain, it adds a tag of 20.

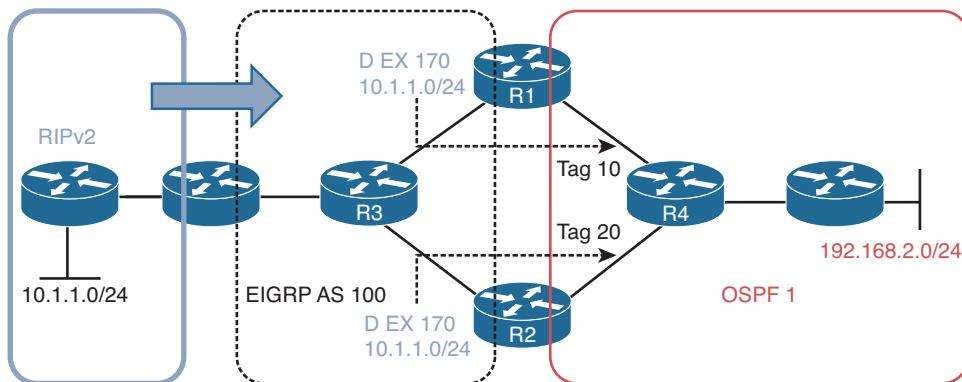


Figure 17-10 Adding Tags to Routes During Redistribution

Example 17-71 displays the commands that you could use to tag the 10.1.1.0/24 routes as they are redistributed on R1 and R2. First you have to define the routes you want to tag with an ACL or prefix list. Then you create a route map that will have a sequence that matches the ACL or prefix list created, which will then set the desired tag upon a match. In this case, R1 sets a tag of 10, and R2 sets a tag of 20. Do not forget about all the other routes you want to redistribute without a tag. That is what sequence 20 is for in the route map. If you forget it, all other routes are denied and not redistributed. You then attach the route map to the redistribution command.

Example 17-71 *Tagging Routes as They Are Being Redistributed*

```

R1#
ip prefix-list TAG_10.1.1.0/24 seq 5 permit 10.1.1.0/24
!
route-map REDIS_EIGRP_TO_OSPF permit 10
  match ip address prefix-list TAG_10.1.1.0/24
  set tag 10
route-map REDIS_EIGRP_TO_OSPF permit 20
!
router ospf 1
  redistribute eigrp 100 subnets route-map REDIS_EIGRP_TO_OSPF

R2#
ip prefix-list TAG_10.1.1.0/24 seq 5 permit 10.1.1.0/24
!
route-map REDIS_EIGRP_TO_OSPF permit 10
  match ip address prefix-list TAG_10.1.1.0/24
  set tag 20
route-map REDIS_EIGRP_TO_OSPF permit 20
!
router ospf 1
  redistribute eigrp 100 subnets route-map REDIS_EIGRP_TO_OSPF

```

You are not done yet. To prevent R1 and R2 from redistributing the OSPF-learned 10.1.1.0/24 routes with their tags back into EIGRP, you deny the routes based on their tags. As shown in Figure 17-11, on R1 you deny the routes with a tag of 20 from being redistributed into the EIGRP autonomous system, and on R2 you deny the routes with a tag of 10 from being redistributed into the EIGRP autonomous system.

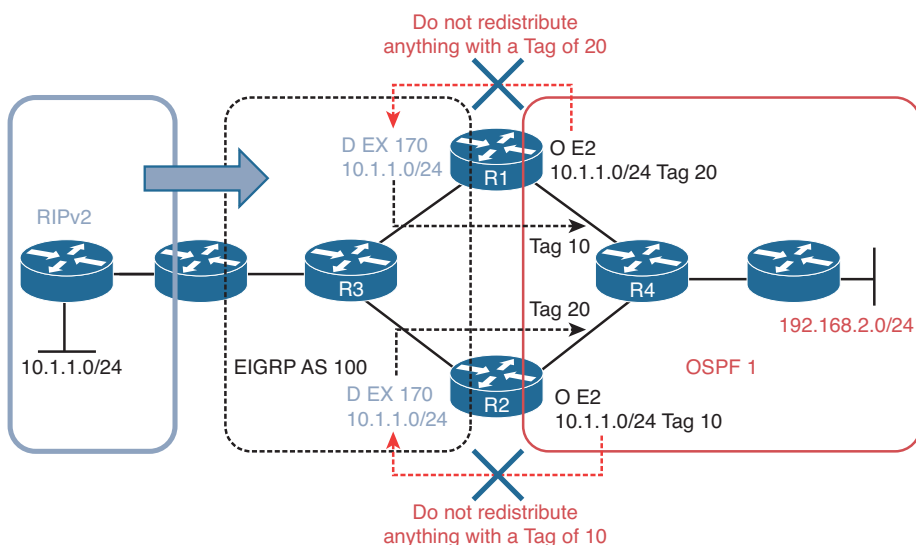


Figure 17-11 *Deny Routes with Certain Tags During Redistribution*

Example 17-72 displays the commands that would be used to ensure that R1 and R2 do not redistribute the 10.1.1.0/24 networks back into the EIGRP autonomous system. Notice the very first sequence in this route map. In this case, it is deny, and when deny is used with redistribution, it indicates that whatever matches will not be redistributed. Therefore, R1 will not redistribute from OSPF into EIGRP any routes that have a tag of 20, as shown in sequence 10, and sequence 20 allows all other routes to be redistributed. For R2, it will not redistribute any routes with a tag of 10 from OSPF into EIGRP based on sequence 10, and all other routes will be redistributed based on sequence 20.

Example 17-72 *Using Route Tags to Prevent Routes from Being Reinjecte*

```
R1#
route-map REDIS_OSPF_INT0_EIGRP deny 10
  match tag 20
route-map REDIS_OSPF_INT0_EIGRP permit 20
!
router eigrp 100
  redistribute ospf 1 metric 100000 100 255 1 1500 route-map REDIS_OSPF_INT0_EIGRP

R2#
route-map REDIS_OSPF_INT0_EIGRP deny 10
  match tag 10
route-map REDIS_OSPF_INT0_EIGRP permit 20
!
router eigrp 100
  redistribute ospf 1 metric 100000 100 255 1 1500 route-map REDIS_OSPF_INT0_EIGRP
```

So, to wrap up our coverage on advanced redistribution scenarios, keep these points in mind:

- Internal prefix information should always be preferred over external prefix information.
- Prefixes should never be redistributed back into a routing domain that they were originally redistributed from.
- A topological diagram is mandatory if you expect to solve the issues quickly and efficiently.

Exam Preparation Tasks

As mentioned in the section “How to Use This Book” in the Introduction, you have a couple of choices for exam preparation: the exercises here; Chapter 22, “Final Preparation;” and the exam simulation questions on the CD-ROM.

Review All Key Topics

Review the most important topics in this chapter, noted with the Key Topic icon in the outer margin of the page. Table 17-3 lists a reference of these key topics and the page numbers on which each is found.



Table 17-3 *Key Topics for Chapter 17*

Key Topic Element	Description	Page Number
Paragraph	Describes the redistribution process	700
List	Displays the three methods that can be used to configure a seed metric	701
List	Displays the prerequisites for redistributing a route	702
Table 17-2	Troubleshooting targets for route redistribution	702
Section	Troubleshooting redistribution into RIP	703
Section	Troubleshooting redistribution into EIGRP	706
Section	Troubleshooting redistribution into OSPF	710
Section	Troubleshooting redistribution into BGP	715
List	Identifies what to look out for when troubleshooting redistribution that uses route maps	718
Paragraph	Describes how to prevent suboptimal routing caused by redistribution	738
List	Outlines what you should review when troubleshooting suboptimal routing issues that were caused by redistribution	739
Section	Troubleshooting routing loops caused by redistribution	739

Define Key Terms

Define the following key terms from this chapter and check your answers in the glossary:

redistribution, boundary router, metric, seed metric, **subnets** keyword, Type 5 LSA, ASBR, routing loop, single-point redistribution, multipoint redistribution, route tag, administrative distance

Command Reference to Check Your Memory

This section includes the most important **show** commands covered in this chapter. It might not be necessary to memorize the complete syntax of every command, but you should be able to remember the basic keywords that are needed.

To test your memory of the commands, cover the right side of Table 17-4 with a piece of paper, read the description on the left side, and then see how much of the command you can remember.

The 300-135 TSHOOT exam focuses on practical, hands-on skills that are used by a networking professional. Therefore, you should be able to identify the commands needed to successfully troubleshoot the topics and concepts covered in this chapter.

Table 17-4 *show commands*

Task	Command Syntax
Displays the IPv4 sources of routing information that are being redistributed into the various IPv4 routing protocols enabled on the device.	show ip protocols
Displays the IPv6 sources of routing information that are being redistributed into the various IPv6 routing protocols enabled on the device.	show ipv6 protocols
For redistribution, it shows which routes have been redistributed into the RIPv2 process on the boundary router.	show ip rip database
For redistribution, it shows which IPv4 routes have been redistributed into the EIGRP for IPv4 process on the boundary router.	show ip eigrp topology
For redistribution, it shows which IPv6 routes have been redistributed into the EIGRP for IPv6 process on the boundary router.	show ipv6 eigrp topology
Shows which IPv4 routes have been redistributed into the OSPFv2 process. They are represented as Type 5 or Type 7 LSAs.	show ip ospf database
Shows which IPv6 routes have been redistributed into the OSPFv3 process. They are represented as Type 5 or Type 7 LSAs.	show ipv6 ospf database
Displays the IPv4 and IPv6 BGP learned routes. Routes originally learned via redistribution have a question mark (?) in the Path column.	show bgp all
Displays a router's BGP router ID, autonomous system number, information about the BGP's memory usage, and summary information about IPv4 unicast BGP neighbors.	show bgp ipv4 unicast summary
Displays detailed information about all the IPv4 BGP neighbors of a router.	show bgp ipv4 unicast neighbors

This page intentionally left blank



This chapter covers the following topics:

- **Troubleshooting BGP Neighbor Adjacencies:** This section examines issues that may prevent a BGP neighbor relationship from forming, how you can recognize them, and how you can troubleshoot them. Although it centers primarily on IPv4 unicast BGP, the same issues will arise with IPv6 unicast BGP neighbor relationships.
- **Troubleshooting BGP Routes:** This section focuses on issues that may prevent BGP routes from being learned or advertised, how you can recognize them, and how you can troubleshoot them. Although it focuses mostly on IPv4 unicast BGP, the same issues will arise with IPv6 unicast BGP routes as well.
- **Troubleshooting BGP Path Selection:** This section explains how BGP determines the best path to reach a destination network and the importance of understanding how this process works for troubleshooting purposes.
- **Troubleshooting BGP for IPv6:** This section discusses the methods needed to successfully troubleshoot additional issues related to BGP for IPv6 that are not seen with BGP for IPv4.
- **BGP Trouble Tickets:** This section provides trouble tickets that demonstrate how you can use a structured troubleshooting process to solve a reported problem.
- **MP-BGP Trouble Tickets:** This section provides trouble tickets that demonstrate how you can use a structured troubleshooting process to solve a reported problem.

Troubleshooting BGP

Border Gateway Protocol (BGP) is the protocol of the Internet. It has been designed to exchange routing information between different autonomous systems (networks under different administrative control). That is why it is classified as an Exterior Gateway Protocol (EGP). It makes best path decisions based on attributes such as local preference, length of autonomous system path, and even BGP router ID (RID), instead of bandwidth like Open Shortest Path First (OSPF), bandwidth and delay like Enhanced Interior Gateway Routing Protocol (EIGRP), or router hops like Routing Information Protocol (RIP). BGP is the most scalable, robust, controllable protocol. However, with that comes a price. That price is mistakes that lead to issues that you have to troubleshoot.

BGP will primarily be used by organizations to connect to their Internet service provider (ISP). If not, static routes are used. However, ISPs use BGP extensively to share Internet routes with each other. The 300-135 TSHOOT exam is not based on ISP-to-ISP BGP connectivity. It is based on enterprise-to-ISP connectivity. Therefore, you need to focus your efforts on troubleshooting the basics of BGP for IPv4 and IPv6 connectivity and route advertising.

In this chapter, you learn the various issues that you may face when trying to establish an IPv4 and IPv6 External Border Gateway Protocol (eBGP) and Internal Border Gateway Protocol (iBGP) neighbor adjacency and how you can identify them and troubleshoot them. The chapter also covers issues that may arise when exchanging IPv4 and IPv6 eBGP and iBGP routes and how you can recognize them and troubleshoot them successfully. Because BGP is classified as a path vector protocol and its decisions are based on attributes, you need to be very familiar with the decision-making process that BGP uses to be an efficient troubleshooter. Therefore, you will spend time exploring this process in the chapter as well.

“Do I Know This Already?” Quiz

The “Do I Know This Already?” quiz allows you to assess whether you should read this entire chapter thoroughly or jump to the “Exam Preparation Tasks” section. If you are in doubt about your answers to these questions or your own assessment of your knowledge of the topics, read the entire chapter. Table 18-1 lists the major headings in this chapter and their corresponding “Do I Know This Already?” quiz questions. You can find the answers in Appendix A, “Answers to the ‘Do I Know This Already?’ Quizzes.”

Table 18-1 “Do I Know This Already?” Section-to-Question Mapping

Foundation Topics Section	Questions
Troubleshooting BGP Neighbor Adjacencies	1–5
Troubleshooting BGP Routes	6–10
Troubleshooting BGP Path Selection	11
Troubleshooting BGP for IPv6	12–13

Caution The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you do not know the answer to a question or are only partially sure of the answer, you should mark that question as wrong for purposes of the self-assessment. Giving yourself credit for an answer that you correctly guess skews your self-assessment results and might provide you with a false sense of security.

- Which commands enable you to identify the IPv4 unicast BGP neighbor adjacencies that have been formed? (Choose two answers.)
 - show ip route bgp
 - show bgp ipv4 unicast
 - show bgp ipv4 unicast summary
 - show bgp ipv4 unicast neighbors
- In the output of **show bgp ipv4 unicast summary**, how can you determine whether a neighbor relationship is successfully established?
 - The neighbor is listed in the output.
 - The version column has a 4 in it.
 - The State/PfxRcd column has a number in it.
 - The State/PfxRcd column has the word Active in it.
- Which of the following are reasons as to why a BGP neighbor relationship might not form? (Choose two answers.)
 - The BGP timers are mismatched.
 - The BGP packets are sourced from wrong IP.
 - The neighbor is reachable via a default route.
 - The **network** command is misconfigured.
- Which TCP port number is used to form BGP sessions?
 - 110
 - 123
 - 179
 - 443

5. What is the BGP state of a neighbor if a TCP session cannot be formed?
 - a. Open
 - b. Idle
 - c. Active
 - d. Established
6. What could prevent a route from being advertised to another BGP router? (Choose three answers.)
 - a. Mismatched timers
 - b. Split-Horizon rule
 - c. Missing **network mask** command
 - d. Route Filtering
7. Which command enables you to verify the IPv4 BGP routes that have been learned from all BGP neighbors?
 - a. **show ip route bgp**
 - b. **show bgp ipv4 unicast**
 - c. **show bgp ipv4 unicast summary**
 - d. **show bgp ipv4 unicast neighbors**
8. What occurs when the next hop of a BGP-learned route is not reachable?
 - a. The route is discarded.
 - b. The route is placed in the BGP table and advertised to other neighbors.
 - c. The route is placed in the BGP table and not marked as valid.
 - d. The route is placed in the BGP table and the routing table.
9. Which successfully describes the BGP split-horizon rule?
 - a. A BGP router that receives a BGP route via an iBGP peering shall not advertise that route to another router that is an iBGP peer.
 - b. A BGP router that receives a BGP route via an eBGP peering shall not advertise that route to another router that is an iBGP peer.
 - c. A BGP router that receives a BGP route via an eBGP peering shall not advertise that route to another router that is an eBGP peer.
 - d. A BGP router that receives a BGP route via an iBGP peering shall discard the route.

- 10.** Which administrative distances are correct? (Choose two answers.)
- a.** 20 for eBGP
 - b.** 20 for iBGP
 - c.** 200 for eBGP
 - d.** 200 for iBGP
- 11.** Which of the following correctly identify the order of BGP attributes for the best path decision process?
- a.** Weight, local preference, route origin, autonomous system Path, Origin Code, MED
 - b.** Autonomous system path, origin code, MED, weight, local preference, route origin
 - c.** Local preference, weight, route origin, autonomous system path, origin code, MED
 - d.** Weight, local preference, route origin, autonomous system path, MED, origin code
- 12.** What must be done when using MP-BGP? (Choose two answers.)
- a.** The IPv6 neighbors need to be activated in address family configuration mode.
 - b.** The IPv6 neighbors need to be activated in router configuration mode.
 - c.** The IPv6 neighbors need to be defined in router configuration mode.
 - d.** The IPv6 neighbors need to be defined in address family configuration mode.
- 13.** Which command enables you to verify the IPv6 unicast BGP routes that have been learned?
- a.** `show bgp ipv6 unicast`
 - b.** `show bgp ipv6 unicast summary`
 - c.** `show bgp ipv6 unicast neighbor`
 - d.** `show ipv6 route bgp`

Foundation Topics

Troubleshooting BGP Neighbor Adjacencies

BGP establishes neighbor adjacencies manually. This is unlike EIGRP and OSPF, where you enable the process on an interface and neighbor adjacencies are formed dynamically. As a result, BGP configuration is more prone to human error, which leads to greater efforts during the troubleshooting process. In addition, there are two flavors of BGP, Internal BGP (iBGP) and External BGP (eBGP). Being able to understand the differences between the two and recognize issues related to each is important for troubleshooting.

This section covers how BGP neighbor relationships are formed and how to recognize issues that would prevent the neighbor relationships from forming.

To verify IPv4 unicast BGP neighbors, you can use two **show** commands: **show bgp ipv4 unicast summary** (which is the same as using the old **show ip bgp summary** command), and **show bgp ipv4 unicast neighbors** (which is the same as using the old **show ip bgp neighbors** command). For initial verification of neighbors, it is best to use **show bgp ipv4 unicast summary** because it provides a condensed output. The output of **show bgp ipv4 unicast neighbors** is very verbose and is not needed for initial neighbor verification. Example 18-1, which is a sample output of the **show bgp ipv4 unicast summary** command, indicates that R1 has two BGP neighbors. One is at IP address 10.1.12.2 and the other is at 10.1.13.3. They are both eBGP neighbors because their autonomous system number does not match the local autonomous system number. Focus your attention on the State/PfxRcd column. If there is a number in this column (as in this case), it means that we have successfully established a BGP neighbor relationship. If you see Idle or Active, there is a problem forming the neighbor relationship.

Key Topic

Example 18-1 Verifying BGP Neighbors with **show bgp ipv4 unicast summary**

```
R1#show bgp ipv4 unicast summary
BGP router identifier 10.1.13.1, local AS number 65501
BGP table version is 1, main routing table version 1
```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
10.1.12.2	4	65502	16	16	1	0	0	00:11:25	0
10.1.13.3	4	65502	15	12	1	0	0	00:09:51	0

In addition, when a neighbor relationship is formed, a syslog message is generated similar to the following:

```
%BGP-5-ADJCHANGE: neighbor 10.1.12.2 Up
```

Here is a listing of reasons why a BGP neighbor relationship might not form:

- **Interface is down:** The interface has to be up/up.
- **Layer 3 connectivity is broken:** You need to be able to reach the IP address you are trying to form the adjacency with.

Key Topic

- **Path to neighbor is via default route:** You must be able to reach the neighbor using a route other than the default route.
- **Neighbor does not have a route to the local router:** Both routers forming a BGP peering must have routes to each other.
- **Incorrect neighbor statement:** The IP address and autonomous system number in the `neighbor ip_address remote-as as_number` statement must be accurate.
- **ACLs:** An access control list (ACL) or a firewall is blocking TCP port 179.
- **BGP packets sourced from wrong IP address:** The source IP of an inbound BGP packet must match the local neighbor statement.
- **TTL of BGP packet expires:** Peer is further away than permitted
- **Mismatched Authentication:** Both routers must agree on authentication parameters
- **Misconfigured Peer Group:** Peer groups simplify repetitive BGP configurations; however, if not carefully implemented can prevent neighbor relationships from forming or routes from being learned.
- **Timers:** Timers do not have to match; however, if the *minimum holddown from neighbor* option is set, this could prevent a neighbor adjacency.

When troubleshooting BGP neighbor adjacencies, you need to be able to identify these different issues and understand the reasons why they occur. Let's look at them individually.

Interface Is Down

The interface with the IP address that is being used to form BGP neighbor relationships must be up/up. Let's be clear: This could be a physical or logical interface. Remember that you can use a loopback interface to source BGP packets. This practice is popular when you have redundant paths between neighbors. In such a case, if one path fails, for example a local physical interface goes down, the neighbor relationship will still be available using another local physical interface since a loopback interface is the source and destination of the packets. Therefore, if you are sourcing BGP packets with the IP address of Loopback 0, the loopback interface has to be up/up as well as any physical interface that can get you to the IP address you are trying to form the neighbor relationship with. As you have seen numerous times, you can verify the status of an interface with the **show ip interface brief** command.

Layer 3 Connectivity Is Broken

You do not have to be directly connected to form a BGP neighbor relationship or in the same subnet; however, you do have to have Layer 3 connectivity. To verify Layer 3 connectivity, you use the **ping** command. If the ping is successful, you have Layer 3 connectivity. Note that for a router to have Layer 3 connectivity, it needs to have a route in the routing table that will point it in the right direction. If no route to the neighbor exists, a neighbor relationship cannot form.

When reviewing the output of **show bgp ipv4 unicast summary** in Example 18-2, you can see in the State/PfxRcd field it states Idle. This state occurs when the local router is not able to make a TCP connection with the neighbor. In this example, it is the router at 2.2.2.2 R5 is trying to form an adjacency with. Reviewing the routing table on R5 with the **show ip route 2.2.2.2 255.255.255.255** command and pinging 2.2.2.2 from R5, as shown in Example 18-3, proves that Layer 3 connectivity does not exist. It is a good idea to specify the source when pinging. The source will be the IP address of the local device you plan on making the BGP peering with.

Example 18-2 Verifying BGP State with show bgp ipv4 unicast summary

```
R5#show bgp ipv4 unicast summary
BGP router identifier 10.1.45.5, local AS number 65502
BGP table version is 1, main routing table version 1

Neighbor      V      AS MsgRcvd MsgSent  TblVer  InQ  OutQ  Up/Down  State/PfxRcd
2.2.2.2       4      65502      0       0        1    0    0 never    Idle
```

Example 18-3 Verifying Whether a Route Exists to the Neighbor and Whether a Ping Is Successful

```
R5#show ip route 2.2.2.2 255.255.255.255
% Network not in table

R5#ping 2.2.2.2 source 5.5.5.5
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2.2.2.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

Path to Neighbor Is via Default Route



Continuing with the previous discussion on Layer 3 connectivity being broken, Example 18-4 shows that no route to 2.2.2.2 exists; however, the ping to 2.2.2.2 is successful. This is because there is a default route in the routing table on R5, as shown in Example 18-5.

Example 18-4 No Route to Neighbor, but Ping Is Successful

```
R5#show ip route 2.2.2.2 255.255.255.255
% Network not in table

R5#ping 2.2.2.2 source 5.5.5.5
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2.2.2.2, timeout is 2 seconds:
Packet sent with a source address of 5.5.5.5
!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 84/91/104 ms
```


Example 18-5 *Verifying Default Route Exists in Routing Table*

```

R5#show ip route
...output omitted...

Gateway of last resort is 10.1.45.4 to network 0.0.0.0

D*EX 0.0.0.0/0 [170/3328] via 10.1.45.4, 00:08:37, GigabitEthernet1/0
      3.0.0.0/32 is subnetted, 1 subnets
D      3.3.3.3 [90/131072] via 10.1.45.4, 00:53:34, GigabitEthernet1/0
      4.0.0.0/32 is subnetted, 1 subnets
D      4.4.4.4 [90/130816] via 10.1.45.4, 00:53:19, GigabitEthernet1/0
...output omitted...

```

Even though we can reach the neighbor via the default route, BGP does not consider it a valid route to form an adjacency. When looking at the output of **show bgp ipv4 unicast summary** on R5, in Example 18-6, you can see that the state is Idle, which indicates that we cannot form a TCP session.

Example 18-6 *Verifying BGP State on R5 with show bgp ipv4 unicast summary*

```

R5#show bgp ipv4 unicast summary
BGP router identifier 10.1.45.5, local AS number 65502
BGP table version is 1, main routing table version 1

Neighbor      V      AS MsgRcvd MsgSent   TblVer  InQ  OutQ Up/Down  State/PfxRcd
2.2.2.2        4      65502      0       0         1    0    0 never      Idle

```

Neighbor Does Not Have a Route to the Local Router

So far, we have seen that the local router will display a state of idle when it does not have a route to the IP address they are trying to peer with. However, idle will also appear on a router when the neighbor does not have a route back to the local router. In Example 18-7, you can see that the router trying to form a BGP peering with R5 (it is R2) also displays a state of idle even though it has a route to 5.5.5.5, as shown in Example 18-7 also. The idle state is because the routers cannot form the TCP session.

Example 18-7 *Verifying BGP State on R2 and Route to 5.5.5.5*

```

R2#show bgp ipv4 unicast summary
BGP router identifier 2.2.2.2, local AS number 65502
BGP table version is 1, main routing table version 1

Neighbor      V      AS MsgRcvd MsgSent   TblVer  InQ  OutQ Up/Down  State/PfxRcd
5.5.5.5        4      65502      0       0         1    0    0 00:00:13 Idle
10.1.12.1      4      65501      2       2         1    0    0 00:00:12      0

R2#show ip route 5.5.5.5 255.255.255.255

```

```

Routing entry for 5.5.5.5/32
  Known via "eigrp 100", distance 90, metric 131072, type internal
  Redistributing via eigrp 100
  Last update from 10.1.24.4 on GigabitEthernet2/0, 00:23:58 ago
  Routing Descriptor Blocks:
  * 10.1.24.4, from 10.1.24.4, 00:23:58 ago, via GigabitEthernet2/0
    Route metric is 131072, traffic share count is 1
    Total delay is 5020 microseconds, minimum bandwidth is 1000000 Kbit
    Reliability 255/255, minimum MTU 1500 bytes
    Loading 1/255, Hops 2

```

Incorrect neighbor Statement



To form a BGP peering, you use the **neighbor ip_address remote-as as_number** command in BGP configuration mode. Example 18-8 displays two **neighbor remote-as** commands on R2. The **neighbor 5.5.5.5 remote-as 65502** command forms an iBGP peering, and **neighbor 10.1.12.1 remote-as 65501** forms an eBGP peering. The iBGP peering is established because the *remote-as 65502* matches the local autonomous system number used to create the BGP process (*router bgp 65502*). The eBGP peering is established because the *remote-as 65501* is different from the local autonomous system number used to create the BGP process (*router bgp 65502*).

Example 18-8 Verifying neighbor remote-as Commands on R2

```

R2#show run | s router bgp
router bgp 65502
  bgp log-neighbor-changes
  neighbor 5.5.5.5 remote-as 65502
  neighbor 5.5.5.5 update-source Loopback0
  neighbor 10.1.12.1 remote-as 65501

```

There are two very important parts to this command: 1) the address of the peer you will form the peering with; 2) the autonomous system that the peer is in. If you make a mistake with either of these, you will see either the active or idle state.

As we have discussed, if there is no route for the IP address you specify, the state will be idle. However, if a route is found and a three-way TCP handshake is complete, an open message is sent. If there is no response to the open message, the state will be active.

If the autonomous system number specified does not match the peer's autonomous system number, the state will toggle between idle and active.

You can verify the state of the TCP session on the routers using the **show tcp brief all** command. In Example 18-9, you can see that R2 has an established TCP session with a device at 5.5.5.5 and another device at 10.1.12.1.

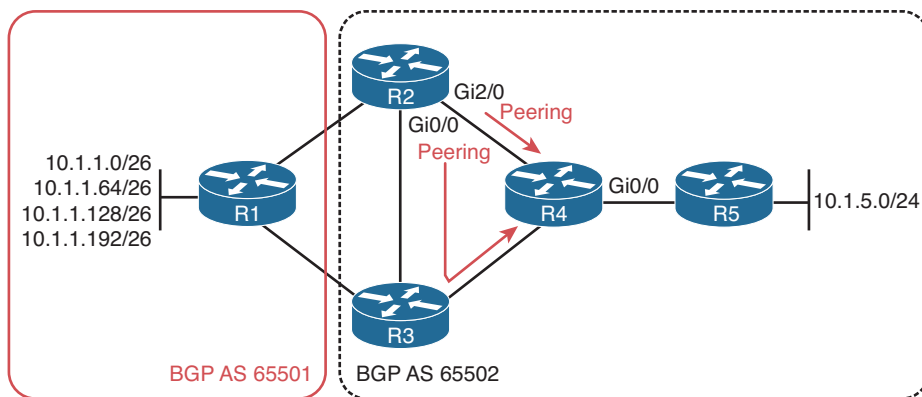
Example 18-9 *Verifying State of TCP Sessions*

```
R2#show tcp brief all
```

TCB	Local Address	Foreign Address	(state)
68DD357C	10.1.12.2.179	10.1.12.1.35780	ESTAB
68DD24DC	2.2.2.2.179	5.5.5.5.45723	ESTAB

BGP Packets Sourced from Wrong IP Address

In a redundant topology, a BGP router will have multiple active IP addresses configured across its various interfaces. Figure 18-1 displays two BGP autonomous systems. Notice that R2, R3, and R4 could form a BGP peering with each other using any physical interface because of the multiple paths. For example, R2 could form a peering with R4 over the direct connection or through the connection via R3.

**Figure 18-1** *Sample BGP autonomous system with redundancy*

When you issue the **neighbor ip_address remote-as as_number** command on a router, the *address* specified is used by the router to determine whether the BGP open message came from a router it should establish a BGP peering with. The BGP open message will have a source IP address, and the source IP address is compared with the address in the local **neighbor remote-as** command. If they match, a BGP peering is formed; if not, no BGP peering is formed. The source address is based on the exit interface of the router sending the BGP open message. Therefore, if R2 sends the BGP open message from Gi2/0 to R4, R4 needs to have a **neighbor** statement with R2's Gi2/0 IP address. Now, if the link between R2 and R4 fails, R2 and R4 can still peer using the links through R3. However, now R2 sends the BGP open message with the source IP of Gi0/0, but R4's **neighbor remote-as** statement is using the Gi2/0 IP address of R2 still, and as a result, no BGP peering is formed because the BGP packets are sourced from the wrong IP address.



To control the IP address that is used when sending BGP messages, you use the **neighbor ip_address update-source interface_type interface_number** command. Example 18-10 displays the output of **show run | section router bgp** on R2. Notice how the peering with R4 is using the address 4.4.4.4 (which is a loopback interface on R4) and all BGP messages sent to 4.4.4.4 will use the IP address of loopback 0 which is 2.2.2.2, as shown in Example 18-10 as well.

Example 18-10 *Verifying State of TCP Sessions*

```

R2#show run | section router bgp
router bgp 65502
  bgp log-neighbor-changes
  neighbor 4.4.4.4 remote-as 65502
  neighbor 4.4.4.4 update-source Loopback0
  neighbor 10.1.12.1 remote-as 65501

R2#show ip interface brief | include Loopback
Loopback0      2.2.2.2          YES manual up          up

```

It is imperative that R4 is configured appropriately as well. In this case, R4 would need to have a **neighbor remote-as** statement using R2's address of 2.2.2.2 in addition to a **neighbor** statement with the **update-source** option that allows it to control the source address of BGP messages sent to R2. Example 18-11 displays the appropriate configuration on R4 to ensure that a BGP peering is successful.

Example 18-11 *Verifying that R4's BGP Configuration Mirrors R2*

```

R4#show run | section router bgp
router bgp 65502
  bgp log-neighbor-changes
  neighbor 2.2.2.2 remote-as 65502
  neighbor 2.2.2.2 update-source Loopback0

R4#show ip interface brief | include Loopback
Loopback0      4.4.4.4          YES manual up          up

```

ACLs

BGP uses TCP port 179 to establish TCP sessions. The TCP session is then used to form the BGP peering. If there is an access control list (ACL) configured that blocks TCP port 179 anywhere in the path between the routers attempting to form a BGP peering, the peering will not happen. In Example 18-12, R4 (refer to Figure 18-1) has ACL 100 attached to interface Gig0/0, which denies packets sourced or destined to port 179 (BGP). As a result, a BGP peering between R2 and R5 is not possible as the packets relating to BGP port 179 are being denied. At the bottom of Example 18-12, the state is idle on R5 because the TCP session cannot be established with the neighbor at 2.2.2.2 because R4 is denying TCP traffic related to port 179.

Example 18-12 *Verifying ACLs Blocking BGP Packets and the State of R5's Neighbor Relationship*

```

R4#show access-lists
Extended IP access list 100
 10 deny tcp any any eq bgp
 20 deny tcp any eq bgp any
 30 permit ip any any

```

```
R4#show ip interface gigabitEthernet 0/0 | include access list
```

```
Outgoing access list is 100
```

```
Inbound access list is not set
```

```
R5#show bgp ipv4 unicast summary
```

```
BGP router identifier 10.1.45.5, local AS number 65502
```

```
BGP table version is 1, main routing table version 1
```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
2.2.2.2	4	65502	0	0	1	0	0	00:02:24	Idle



In Example 18-12, the access list is denying BGP packets sourced or destined to port 179. However, what if the ACL were only blocking BGP port 179 packets in one direction? For example, the entry was only **deny tcp any any eq bgp** while still being applied to Gig0/0 outbound. This means that only packets destined to port 179 outbound on Gig0/0 will be blocked. What if they were sourced from 179 going outbound instead? They would no longer be blocked. So, in this case, if you could control who the server and clients are for the BGP TCP sessions, you could still form the BGP TCP session.

That's right, BGP sessions are a server/client relationship. One router is using port 179 (server), and the other router is using an ephemeral port (client). By default, both routers will try to establish a TCP session using the three-way handshake because both routers will send a TCP syn packet sourced from an ephemeral port and destined to port 179. They both respond with a syn/ack sourced from 179 destined to the ephemeral port, and then both send an ack sourced from the ephemeral port destined to port 179. This causes two BGP sessions between the devices when there can only be one. This situation is called a BGP connection collision, and BGP will sort it out automatically. In a nutshell, the router with the higher BGP RID becomes the server.

If you want to avoid this issue, you can control who the server and client are right from the start by using the **neighbor ip_address transport connection-mode {active | passive}** command. By specifying **active**, you are indicating that you want the router to actively initiate the TCP session; therefore, active means client. By specifying **passive**, you are indicating that you want the router to passively wait for another router to initiate the TCP session; therefore, passive means server.

Using the command **show bgp ipv4 unicast neighbor** will show the local and remote port numbers that are being used. If the local port is port 179 and the remote port is an ephemeral port, the local router is the server. If the remote port is 179 and the local port is an ephemeral port, the local router is the client. In Example 18-13, the command **show bgp ipv4 unicast neighbors | i ^BGP neighbor|Local port|Foreign port** was used to just display R2's neighbors along with the local port number and the foreign port number. Notice how R2 is the client for the TCP sessions with R1 (1.1.1.1), R4 (4.4.4.4), and R5 (5.5.5.5) because the local port is a random port number. R2 is the server for the TCP session with R3 because the local port is the BGP port number of 179.

Example 18-13 *Verifying Local and Foreign BGP Port Numbers*

```
R2#show bgp ipv4 unicast neighbors | i ^BGP neighbor|Local port|Foreign port
BGP neighbor is 1.1.1.1, remote AS 65501, external link
Local host: 2.2.2.2, Local port: 23938
Foreign host: 1.1.1.1, Foreign port: 179
BGP neighbor is 3.3.3.3, remote AS 65502, internal link
Local host: 2.2.2.2, Local port: 179
Foreign host: 3.3.3.3, Foreign port: 45936
BGP neighbor is 4.4.4.4, remote AS 65502, internal link
Local host: 2.2.2.2, Local port: 34532
Foreign host: 4.4.4.4, Foreign port: 179
BGP neighbor is 5.5.5.5, remote AS 65502, internal link
Local host: 2.2.2.2, Local port: 49564
Foreign host: 5.5.5.5, Foreign port: 179
```

TTL of BGP Packet Expires

By default, an eBGP peering occurs between directly connected routers. This means the routers forming the eBGP peering are expected to be within one router hop of each other. With an iBGP peering, the routers can be up to 255 router hops from each other and still form a peering. Example 18-14 shows the output of **show bgp ipv4 unicast neighbors | include BGP neighbor|TTL**, which displays that the eBGP neighbor at 10.1.12.1 must be reachable in 1 router hop, and the iBGP neighbor at 5.5.5.5 can be up to 255 hops away. If the neighbor is not reachable in the number of hops listed, the BGP packet expires, and no neighbor relationship is formed.

Example 18-14 *Verifying the TTLs of eBGP and iBGP Packets*

```
R2#show bgp ipv4 unicast neighbors | include BGP neighbor|TTL
BGP neighbor is 5.5.5.5, remote AS 65502, internal link
Minimum incoming TTL 0, Outgoing TTL 255
BGP neighbor is 10.1.12.1, remote AS 65501, external link
Minimum incoming TTL 0, Outgoing TTL 1
```

If the TTL is not large enough to support the distance required to form a BGP peering, the packet will be discarded. For example, let's form an eBGP peering between R1 and R2 in Figure 18-2 using their loopback interfaces. R1 has a loopback interface of 1.1.1.1, and R2 has a loopback interface of 2.2.2.2. Layer 3 connectivity has been tested with a ping, and it is successful. It is also not via a default route.

Example 18-15 displays the configuration of R1 and R2. Notice that R1 is peering with R2 using the neighbor address 2.2.2.2 (R2 loopback) and that source address of loopback 0 (1.1.1.1). R2 is peering with R1 using the neighbor address 1.1.1.1 (R1 loopback) and source address of loopback 0 (2.2.2.2). Note that these loopback interfaces are not directly connected (one hop away), and because it is an eBGP neighbor relationship, we expect the peering to fail.

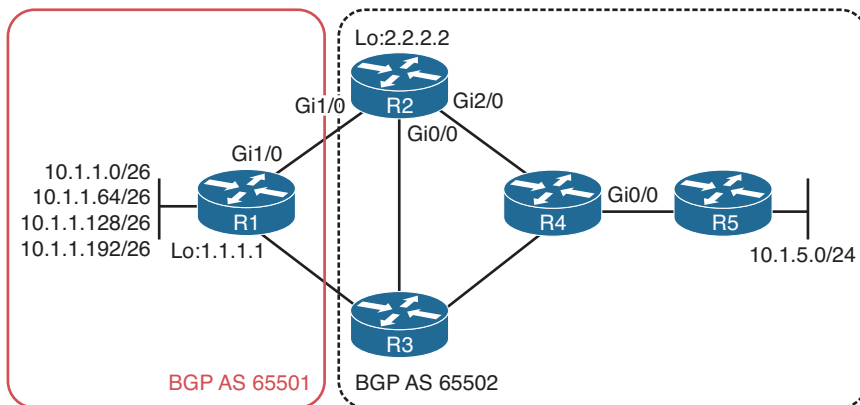


Figure 18-2 Forming BGP Peering Between R1 and R2 Using Loopback Interfaces

Example 18-15 Verifying BGP Configuration on R1 and R2

```
R1#show run | s router bgp
router bgp 65501
  bgp log-neighbor-changes
  neighbor 2.2.2.2 remote-as 65502
  neighbor 2.2.2.2 update-source Loopback0
  neighbor 10.1.13.3 remote-as 65502

R2#show run | s router bgp
router bgp 65502
  bgp log-neighbor-changes
  neighbor 1.1.1.1 remote-as 65501
  neighbor 1.1.1.1 update-source Loopback0
  neighbor 5.5.5.5 remote-as 65502
  neighbor 5.5.5.5 update-source Loopback0
```

Reviewing the output of **show bgp ipv4 unicast summary**, as shown in Example 18-16, clearly indicates that the peering is not forming as both routers are in the idle state. This is a result of the eBGP peers addresses not being directly connected (one router hop).

Example 18-16 Verifying BGP States on R1 and R2

```
R1#show bgp ipv4 unicast summary
BGP router identifier 10.1.13.1, local AS number 65501
BGP table version is 1, main routing table version 1

Neighbor      V      AS MsgRcvd MsgSent   TblVer  InQ  OutQ  Up/Down  State/PfxRcd
2.2.2.2       4    65502      0       0        1    0    0  never    Idle
10.1.13.3     4    65502     36     35        1    0    0  00:29:49    0

R2#show bgp ipv4 unicast summary
```

```
BGP router identifier 2.2.2.2, local AS number 65502
BGP table version is 1, main routing table version 1
```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
1.1.1.1	4	65501	0	0	1	0	0	never	Idle
5.5.5.5	4	65502	27	26	1	0	0	00:20:52	0

Key Topic

To solve this issue with eBGP neighbors, you can modify the TTL of eBGP packets using the **neighbor ip_address ebgp-multihop [TTL]** command. In this case, two would be enough to solve the issue. Therefore, on R1, you can type **neighbor 2.2.2.2 ebgp-multihop 2**, and on R2, you can type **neighbor 1.1.1.1 ebgp-multihop 2**. As you can see in Example 18-17, it now states on R2 that neighbor 1.1.1.1 can be up to two hops away and that the peering is established, as shown in the output of **show bgp ipv4 unicast summary**.

Example 18-17 Verifying Modified TTLs of eBGP Packets

```
R2#show bgp ipv4 unicast neighbors | include BGP neighbor|TTL
BGP neighbor is 1.1.1.1, remote AS 65501, external link
  External BGP neighbor may be up to 2 hops away.
BGP neighbor is 5.5.5.5, remote AS 65502, internal link
Minimum incoming TTL 0, Outgoing TTL 255
```

```
R2#show bgp ipv4 unicast summary
```

```
BGP router identifier 2.2.2.2, local AS number 65502
BGP table version is 1, main routing table version 1
```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
1.1.1.1	4	65501	2	4	1	0	0	00:00:04	0
5.5.5.5	4	65502	38	37	1	0	0	00:30:57	0

Mismatched Authentication

BGP supports message digest 5 (MD5) authentication between peers. Like all discussions on authentication, if any of the parameters do not match, a peering will not form. If you have syslog messaging turned on, a BGP authentication mismatch will generate a syslog message from the TCP facility, as follows:

```
%TCP-6-BADAUTH: No MD5 digest from 2.2.2.2(179) to 1.1.1.1(45577) tableid - 0
```

In addition, the BGP state will be idle, as shown in Example 18-18.

Example 18-18 Verifying Neighbor State With Mismatched Authentication

```
R1#show bgp ipv4 unicast summary
```

```
BGP router identifier 1.1.1.1, local AS number 65501
BGP table version is 1, main routing table version 1
```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
2.2.2.2	4	65502	0	0	1	0	0	00:02:49	Idle
10.1.13.3	4	65502	7	5	1	0	0	00:02:48	0

Misconfigured Peer Groups

When a BGP-enabled router needs to send updates, it will build a separate update for each of the neighbors it has. When a router has a large number of BGP neighbors, this can have a significant impact on the routers CPU. To conserve processing power, you can implement BGP peer groups. With BGP peer groups, the router only has to run the BGP update for the entire group instead of on a neighbor-by-neighbor basis. However, even though the update is run only once, the TCP transmission has to occur on a per-neighbor basis. In addition to saving CPU cycles, peer groups allow you to type or copy and paste less. Example 18-19 displays a sample peer group configuration. When troubleshooting peer group issues, you need to look out for a few general things:

- **You forgot to associate the neighbor ip address with the peer group:** Once the peer group is created, you need to use the `neighbor ip_address peer-group peer_group_name` command to associate the neighbor with the configurations in the peer group. If you forget to do this, the neighbor IP address is not using the configs in the peer group. It will be using the BGP configs outside the peer group, which could prevent a neighbor relationship from forming.
- **The peer group is not configured correctly:** It is possible that you overlooked the fact that what works for one neighbor might not work for the other. For example, using an update source of Loopback 0 may work well for the iBGP peer but not for the eBGP peer.
- **The route filter applied to the group is not appropriate for all the peers:** The filter applied via a route map or any other means may not provide the result you expect on all the routers. Be careful with filters and make sure that they produce the desired result for all neighbors in the peer group.
- **Order of operations produces undesired result:** If there are conflicting entries between the peer group and a specific `neighbor` statement, the `neighbor` statement wins. In Example 18-19, the peer group states the update source is Loopback 0. However, for neighbor 3.3.3.3, it states specifically that Loopback 1 will be used with the command `neighbor 3.3.3.3 update-source Loopback1`. This specific `neighbor` statement overrides the peer group.

Example 18-19 Peer Group Configuration Example

```
R2#show run | section router bgp
router bgp 65502
  bgp log-neighbor-changes
  network 10.1.5.0 mask 255.255.255.0
  neighbor TSHOOT_IBGP_NEIGHBORS peer-group
  neighbor TSHOOT_IBGP_NEIGHBORS transport connection-mode passive
  neighbor TSHOOT_IBGP_NEIGHBORS update-source Loopback0
  neighbor TSHOOT_IBGP_NEIGHBORS next-hop-self
  neighbor TSHOOT_IBGP_NEIGHBORS route-map TSHOOT_BGP_FILTER out
  neighbor 1.1.1.1 remote-as 65501
  neighbor 1.1.1.1 password CISCO
```

```

neighbor 1.1.1.1 ebgp-multihop 2
neighbor 1.1.1.1 update-source Loopback0
neighbor 3.3.3.3 remote-as 65502
neighbor 3.3.3.3 peer-group TSHOOT_IBGP_NEIGHBORS
neighbor 3.3.3.3 update-source Loopback1
neighbor 4.4.4.4 remote-as 65502
neighbor 4.4.4.4 peer-group TSHOOT_IBGP_NEIGHBORS
neighbor 5.5.5.5 remote-as 65502
neighbor 5.5.5.5 peer-group TSHOOT_IBGP_NEIGHBORS

```

Timers

Let's be clear, BGP timers do not have to match. This is because BGP will use the lowest timers set between the two neighbors. If R1 is configured with a default hello of 60 and hold time of 180 and R3 is configured with a hello of 30 and hold time of 90, a hello of 30 and hold time of 90 will be used between the two neighbors, as shown in Example 18-20.



Notice how R3 was configured with a *minimum hold-time* of 90 seconds; this is done to ensure that if a neighbor is using aggressive timers, they will not be used. However, it is far worse than the timers simply not being used. The neighbor relationship will not form at all. Refer to Example 18-21. In this case, R1 has a hello interval set to 10 and hold time set to 30. R3 has the minimum hold time set to 90 seconds. Therefore, it will not agree with the 30-second hold time set by R1, and the neighbor relationship fails. You can see in the output a BGP notification is received stating that the hold time is not acceptable.

Example 18-20 Verifying BGP Timers

```

R1#show bgp ipv4 unicast neighbors 10.1.13.3 | include hold time|holdtime
  Last read 00:00:02, last write 00:00:29, hold time is 90, keepalive interval is 30
seconds
R3#show bgp ipv4 unicast neighbors 10.1.13.1 | include hold time|holdtime
  Last read 00:00:10, last write 00:00:23, hold time is 90, keepalive interval is 30
seconds
  Configured hold time is 90, keepalive interval is 30 seconds
  Minimum holdtime from neighbor is 90 seconds

```

Example 18-21 Modifying BGP Timers to Values That Are Not Acceptable on R1

```

R1#config t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#router bgp 65501
R1(config-router)#neighbor 10.1.13.3 timers 10 30
R1(config-router)#do clear ip bgp 10.1.13.3
R1(config-router)#
%BGP-5-ADJCHANGE: neighbor 10.1.13.3 Down User reset
%BGP_SESSION-5-ADJCHANGE: neighbor 10.1.13.3 IPv4 Unicast topology base removed from
session User reset

```

```
%BGP-3-NOTIFICATION: received from neighbor 10.1.13.3 active 2/6 (unacceptable hold
time) 0 bytes
R1(config-router)#
%BGP-5-NBR_RESET: Neighbor 10.1.13.3 active reset (BGP Notification received)
%BGP-5-ADJCHANGE: neighbor 10.1.13.3 active Down BGP Notification received
%BGP_SESSION-5-ADJCHANGE: neighbor 10.1.13.3 IPv4 Unicast topology base removed from
session BGP Notification received
R1(config-router)#
%BGP-3-NOTIFICATION: received from neighbor 10.1.13.3 active 2/6 (unacceptable hold
time) 0 bytes
R1#
```

To summarize timers, they do not have to match, but if the minimum hold time is set, the lowest timers must not be less than the minimum; otherwise, a neighbor relationship will not form.

Troubleshooting BGP Routes

Once a BGP adjacency is formed, BGP routers exchange their BGP routes with each other. However, there are various reasons as to why BGP routes might be missing from either the BGP table or the routing table. This section explains those reasons and how we can identify them using our troubleshooting methods.

As discussed already, peers are the foundation for BGP information sharing. If we have no peers, we will not learn BGP routes. So, besides the lack of peers, what would be reasons for missing routes in a BGP network? Following is a listing of some common reasons as to why BGP routes might be missing either in the BGP table or the routing table:



- **Missing or bad network mask command:** An accurate network command is needed to advertise routes.
- **Next-hop router not reachable:** To use a BGP route, the next hop must be reachable.
- **BGP split-horizon rule:** A router that learns BGP routes through an iBGP peering will not share those routes with another iBGP peer.
- **Better source of information:** If the exact same network is learned from a more reliable source, it is used instead of the BGP-learned information.
- **Route filtering:** A filter might be set up that prevents a route from being shared with neighbors or learned from neighbors.

To verify the IPv4 unicast BGP-learned routes or routes locally injected into the BGP table, you use the **show bgp ipv4 unicast** command (which is the same as the old **show ip bgp** command), as shown in Example 18-22. Routes will appear in this table for the following reasons:

- Another BGP router advertises them to the local router.
- The **network mask** command matches a route in the local routing table.

- A **redistribute** command is used to import the route from another local source.
- The **summary-address** command is used to create a summary route.

It is not easy to determine the exact sources for all of the networks by looking only at the BGP table. Reviewing the commands in the running configuration along with the output of the BGP table will give you the most accurate information. However, in the BGP table, a network with a next hop other than 0.0.0.0 indicates the router learned it from a peer. If the next hop is 0.0.0.0, it means that the local router originated the route. If the Path column ends in ?, you can conclude that it was redistributed into the BGP process at some point. If the Path column ends in i, it means that the route was injected with the **summary-address** command or the **network mask** command.

**Key
Topic**

Example 18-22 Examining the BGP Table

```
R1#show bgp ipv4 unicast
BGP table version is 10, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

	Network	Next Hop	Metric	LocPrf	Weight	Path
*>	1.1.1.1/32	0.0.0.0	0		32768	?
*>	10.1.1.0/26	0.0.0.0	0		32768	i
*>	10.1.1.0/24	0.0.0.0			32768	i
*>	10.1.1.64/26	0.0.0.0	0		32768	i
*>	10.1.1.128/26	0.0.0.0	0		32768	i
*>	10.1.1.192/26	0.0.0.0	0		32768	i
*	10.1.5.0/24	10.1.13.3	3328		0	65502 i
*>		2.2.2.2	3328		0	65502 i
*>	10.1.12.0/24	0.0.0.0	0		32768	?
*>	10.1.13.0/24	0.0.0.0	0		32768	?

To display the routing table, use the **show ip route** command. To view only the BGP routes, issue the command **show ip route bgp**, as shown in Example 18-23. All BGP routes appear with the code B at the beginning of the entry.

Example 18-23 Examining the BGP Routes in the Routing Table

```
R2#show ip route bgp
...output omitted...

Gateway of last resort is 10.1.12.1 to network 0.0.0.0

    10.0.0.0/8 is variably subnetted, 15 subnets, 3 masks
B       10.1.1.0/24 [20/0] via 1.1.1.1, 00:19:11
```

```

B      10.1.1.0/26 [20/0] via 1.1.1.1, 00:41:04
B      10.1.1.64/26 [20/0] via 1.1.1.1, 00:36:45
B      10.1.1.128/26 [20/0] via 1.1.1.1, 00:36:15
B      10.1.1.192/26 [20/0] via 1.1.1.1, 00:36:15
B      10.1.13.0/24 [20/0] via 1.1.1.1, 00:20:23

```

Let's take a look at each of the reasons individually and identify how we can recognize them during the troubleshooting process.

Missing or Bad network mask Command

The **network mask** command is used to advertise routes into BGP. If you only remember one thing about this command, remember that it is extremely picky. The following list describes why the command is picky:



- The network/prefix you want to advertise with BGP has to be in the routing table from some other source (connected, static, or some other routing protocol).
- The **network mask** command must be a perfect match to the network/prefix listed in the routing table.

If these two requirements are not met, the prefix/network will not be advertised. Review Example 18-24 and determine whether the 10.1.1.0/26 network will be advertised.

Example 18-24 Determining Whether the 10.1.1.0/26 Network Will Be Advertised

```

R1#config t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#router bgp 65501
R1(config-router)#network 10.1.1.0 mask 255.255.255.192
R1(config-router)#end
R1#show ip route
...output omitted...

Gateway of last resort is not set

    1.0.0.0/32 is subnetted, 1 subnets
C      1.1.1.1 is directly connected, Loopback0
    2.0.0.0/32 is subnetted, 1 subnets
S      2.2.2.2 [1/0] via 10.1.12.2
    10.0.0.0/8 is variably subnetted, 12 subnets, 3 masks
C      10.1.1.0/26 is directly connected, GigabitEthernet0/0.1
L      10.1.1.1/32 is directly connected, GigabitEthernet0/0.1
C      10.1.1.64/26 is directly connected, GigabitEthernet0/0.2
L      10.1.1.65/32 is directly connected, GigabitEthernet0/0.2
C      10.1.1.128/26 is directly connected, GigabitEthernet0/0.3
L      10.1.1.129/32 is directly connected, GigabitEthernet0/0.3
C      10.1.1.192/26 is directly connected, GigabitEthernet0/0.4

```

```

L      10.1.1.193/32 is directly connected, GigabitEthernet0/0.4
C      10.1.12.0/24 is directly connected, GigabitEthernet1/0
L      10.1.12.1/32 is directly connected, GigabitEthernet1/0
C      10.1.13.0/24 is directly connected, GigabitEthernet2/0
L      10.1.13.1/32 is directly connected, GigabitEthernet2/0

```

In Example 18-24, the 10.1.1.0/26 network will be advertised because there is an exact match of the network command in the routing table.

Now review Example 18-25. Will the **network mask** command successfully advertise the route indicated?

Example 18-25 *Determining Whether the Network Will Be Advertised*

```

R1#config t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#router bgp 65501
R1(config-router)#network 10.1.1.0 mask 255.255.255.0
R1(config-router)#end
R1#show ip route
...output omitted...

Gateway of last resort is not set

      1.0.0.0/32 is subnetted, 1 subnets
C      1.1.1.1 is directly connected, Loopback0
      2.0.0.0/32 is subnetted, 1 subnets
S      2.2.2.2 [1/0] via 10.1.12.2
      10.0.0.0/8 is variably subnetted, 12 subnets, 3 masks
C      10.1.1.0/26 is directly connected, GigabitEthernet0/0.1
L      10.1.1.1/32 is directly connected, GigabitEthernet0/0.1
C      10.1.1.64/26 is directly connected, GigabitEthernet0/0.2
L      10.1.1.65/32 is directly connected, GigabitEthernet0/0.2
C      10.1.1.128/26 is directly connected, GigabitEthernet0/0.3
L      10.1.1.129/32 is directly connected, GigabitEthernet0/0.3
C      10.1.1.192/26 is directly connected, GigabitEthernet0/0.4
L      10.1.1.193/32 is directly connected, GigabitEthernet0/0.4
C      10.1.12.0/24 is directly connected, GigabitEthernet1/0
L      10.1.12.1/32 is directly connected, GigabitEthernet1/0
C      10.1.13.0/24 is directly connected, GigabitEthernet2/0
L      10.1.13.1/32 is directly connected, GigabitEthernet2/0

```

The **network mask** command in this case is 10.1.1.0/24. Although 10.1.1.0/24 as a summary would include 10.1.1.0/26, 10.1.1.64/26, 10.1.1.128/26, and 10.1.1.192/26, the **network mask** command states advertise this network (10.1.1.0/24). Because 10.1.1.0/24 is not in the routing table, nothing is advertised.

It is important that you are able to recognize a bad or missing **network mask** command as being the reason for missing routes. If a router is not learning a BGP route that it

should and you trace it all the way back to the source, review the running configuration to see whether there is a **network mask** command advertising the network and whether there is a matching route in the routing table.

Next-Hop Router Not Reachable

If you are seeing BGP routes in the BGP table, but they are not appearing in the routing table, the router might not be able to reach the next hop. For a BGP router to install a BGP route in the routing table, it must be able to reach the next-hop address listed for the network. Example 18-26 shows the output of **show bgp ipv4 unicast** on R5. Let's focus on network 10.1.1.0/26. Notice how there is no > symbol after the *. The * > symbols indicate that it is a valid best path to reach that network and has been installed in the routing table. In this case, the path is valid but not the best, and as a result, not placed in the routing table.

Example 18-26 Identifying BGP Next-Hop Issues

```
R5#show bgp ipv4 unicast
BGP table version is 2, local router ID is 5.5.5.5
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

Network	Next Hop	Metric	LocPrf	Weight	Path
* i 1.1.1.1/32	1.1.1.1	0	100	0	65501 ?
* i 10.1.1.0/26	1.1.1.1	0	100	0	65501 i
* i 10.1.1.0/24	1.1.1.1	0	100	0	65501 i
* i 10.1.1.64/26	1.1.1.1	0	100	0	65501 i
* i 10.1.1.128/26	1.1.1.1	0	100	0	65501 i
* i 10.1.1.192/26	1.1.1.1	0	100	0	65501 i
r>i 10.1.5.0/24	10.1.24.4	3328	100	0	i
* i 10.1.12.0/24	1.1.1.1	0	100	0	65501 ?
* i 10.1.13.0/24	1.1.1.1	0	100	0	65501 ?



The reason why it is not being used is because the next-hop address is not reachable. In Example 18-27, the **ping 1.1.1.1** command fails proving that the next hop is not reachable.

Example 18-27 Verifying Next-Hop Reachability

```
R5#ping 1.1.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

Refer to Figure 18-3. Notice where the next-hop address 1.1.1.1 is compared to R5. The next hop for BGP routes outside an autonomous system is the IP address of the router advertising the route to the local autonomous system. The router receiving the advertisement (R2 in this case) does not change the next hop by default because BGP is based on autonomous system-by-autonomous system hops, not on router-by-router hops. Therefore, the next hop is the IP address of the router advertising the network from the next-hop autonomous system.

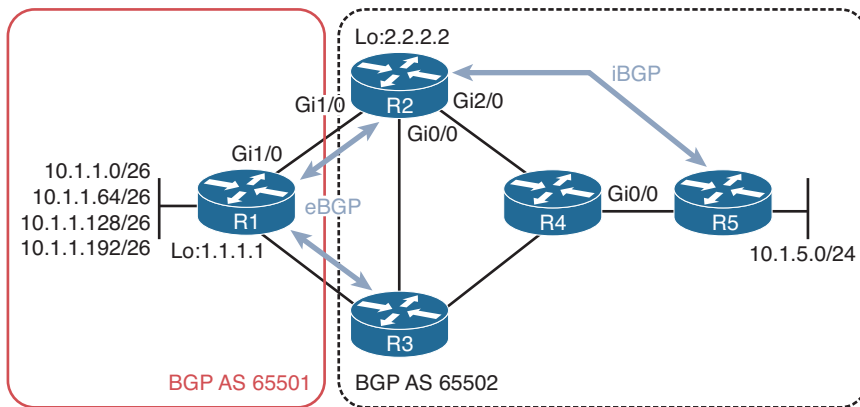


Figure 18-3 Troubleshooting Next-Hop Address Behavior

There are many different ways to solve this problem. The key is to train R5 about how to get to the next hop. The following list contains a few examples:

- Create a static default route on R2 and R3; advertise it into the IGP routing protocol
- Create a static default route on R5
- Create a static route on R5
- Advertise the next-hop address into the Interior Gateway Protocol (IGP) routing protocol

In addition, BGP has a built-in option you can take advantage of. It is the **neighbor ip_address next-hop-self** command. This command allows, for example, R2 to change the next-hop address to its own address before advertising the route to the peer. In Example 18-28, R2 has been configured with the **neighbor 5.5.5.5 next-hop-self** command that changes the next hop to 2.2.2.2 when R2 advertises routes to R5. Example 18-29 displays the BGP table on R5, which now has 2.2.2.2 as the next hop for 10.1.1.0/26, and it now has a > symbol, so it is the best and installed in the routing table.

Example 18-28 Modifying Next-Hop Address

```
R2#config t
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#router bgp 65502
R2(config-router)#neighbor 5.5.5.5 next-hop-self
```


Example 18-29 *Verifying Next-Hop Address in BGP Table*

```

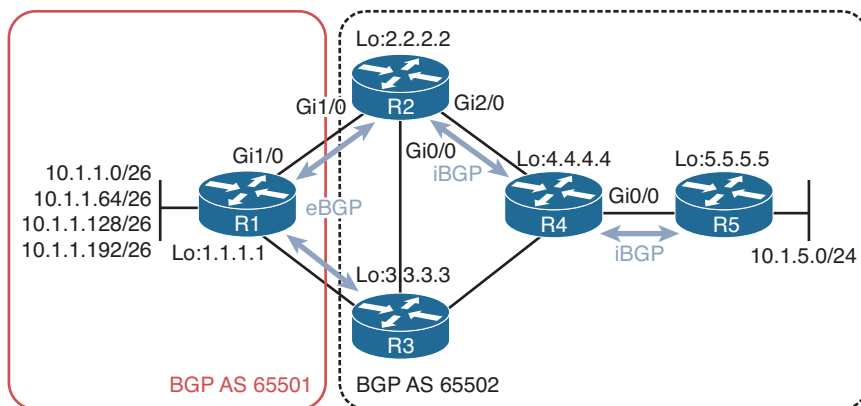
R5#show bgp ipv4 unicast
BGP table version is 13, local router ID is 5.5.5.5
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

```

Network	Next Hop	Metric	LocPrf	Weight	Path
*>i 1.1.1.1/32	2.2.2.2	0	100	0	65501 ?
*>i 10.1.1.0/26	2.2.2.2	0	100	0	65501 i
*>i 10.1.1.0/24	2.2.2.2	0	100	0	65501 i
*>i 10.1.1.64/26	2.2.2.2	0	100	0	65501 i
*>i 10.1.1.128/26	2.2.2.2	0	100	0	65501 i
*>i 10.1.1.192/26	2.2.2.2	0	100	0	65501 i
r>i 10.1.5.0/24	2.2.2.2	3328	100	0	i
r>i 10.1.12.0/24	2.2.2.2	0	100	0	65501 ?
r>i 10.1.13.0/24	2.2.2.2	0	100	0	65501 ?

BGP Split-Horizon Rule

The BGP split-horizon rule states that a BGP router that receives a BGP route via an iBGP peering shall not advertise that route to another router that is an iBGP peer. It is important that you commit this rule to memory. By doing so, you will be able to recognize when this is the reason for missing routes. Figure 18-4 shows the current BGP peerings. Notice that R2 has an iBGP peering with R4 and that R4 has an iBGP peering with R5. When R2 advertises the 10.1.1.0/26 network (as an example) to R4, it is via an iBGP peering. Because R4 and R5 are iBGP peers, R4 will not advertise the 10.1.1.0/26 network to R5 because of the BGP split-horizon rule.

**Figure 18-4** *BGP Peerings Enforcing the BGP Split-Horizon Rule*

For R5 to learn about the 10.1.1.0/26 network, it has to be an iBGP peer with the router that learned about the route from an eBGP peer or it has to be a peer with a route reflector, which is beyond the scope of this book. Figure 18-5 indicates how the iBGP peerings should be to ensure both R4 and R5 learn about 10.1.1.0/26 (as well as the other networks). It also ensures redundancy is optimized in the BGP AS.

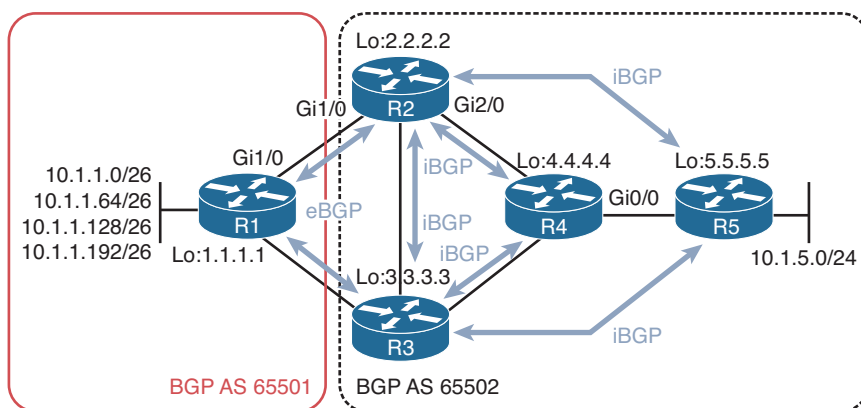


Figure 18-5 Proper BGP Peerings to Avoid the BGP Split-Horizon Rule

Key Topic

Using `show bgp ipv4 unicast summary` on all the routers to identify peerings and then drawing your peerings on paper will give you an idea if the BGP split-horizon rule is causing the missing routes, as long as you remember this: *A BGP router that receives a BGP route via an iBGP peering shall not advertise that route to another router that is an iBGP peer.*

Better Source of Information

Routes learned from eBGP peers have an administrative distance of 20, and routes learned from iBGP peers have an administrative distance of 200. Why the huge difference? BGP is designed to share routes between different autonomous systems. Therefore, if you learn a route from another autonomous system via eBGP, iBGP, or EIGRP sources, you want the eBGP-learned route to be the best source of information over all the other dynamic routing protocols. For example, refer to Figure 18-5 again. R1 advertises 10.1.1.0/26 to R2 using eBGP and R3 using eBGP. R3, because it has an iBGP peering with R2, advertises it to R2 using iBGP. In addition, let's say on R3 we redistribute the 10.1.1.0/26 eBGP-learned route into EIGRP and that R2 learns it via an EIGRP update. Now, R2 knows about the same network from three different sources: eBGP(20), iBGP(200), and EIGRP(170). As a result, the eBGP path is chosen because it has the lower AD. If it was not for eBGP having the lower AD, we would end up with suboptimal routing as a different source is used, and traffic would have to go to R3 first before it leaves the network, instead of directly from R2 to R1.

Example 18-30 displays the output of the IPv4 unicast BGP table on R5 using the `show bgp ipv4 unicast` command. In the table, you will notice that the 10.1.5.0/24, 10.1.12.0/24, and 10.1.13.0/24 networks are *best* (installed in routing table), as indicated by the > symbol; however, they are not *valid*. They are listed as having a RIB failure, as indicated by

the *r*. A RIB failure means that the BGP route was not able to be installed in the routing table; however, you can clearly see that the route is in the routing table because of the *>* symbol. Be careful here. In this case, the route in the routing table is from a better source.

Example 18-30 Verifying BGP Routes

```
R5#show bgp ipv4 unicast
BGP table version is 10, local router ID is 5.5.5.5
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

Network	Next Hop	Metric	LocPrf	Weight	Path
* i 1.1.1.1/32	3.3.3.3	0	100	0	65501 ?
*>i	2.2.2.2	0	100	0	65501 ?
* i 10.1.1.0/26	3.3.3.3	0	100	0	65501 i
*>i	2.2.2.2	0	100	0	65501 i
* i 10.1.1.0/24	3.3.3.3	0	100	0	65501 i
*>i	2.2.2.2	0	100	0	65501 i
* i 10.1.1.64/26	3.3.3.3	0	100	0	65501 i
*>i	2.2.2.2	0	100	0	65501 i
* i 10.1.1.128/26	3.3.3.3	0	100	0	65501 i
*>i	2.2.2.2	0	100	0	65501 i
* i 10.1.1.192/26	3.3.3.3	0	100	0	65501 i
*>i	2.2.2.2	0	100	0	65501 i
r i 10.1.5.0/24	3.3.3.3	3328	100	0	i
r>i	2.2.2.2	3328	100	0	i
r i 10.1.12.0/24	3.3.3.3	0	100	0	65501 ?
r>i	2.2.2.2	0	100	0	65501 ?
r i 10.1.13.0/24	3.3.3.3	0	100	0	65501 ?
r>i	2.2.2.2	0	100	0	65501 ?

Using the command **show ip route 10.1.5.0 255.255.255.0**, as shown in Example 18-31, indicates that 10.1.5.0/24 is learned via connected. In the same example, you can also see the output of **show ip route 10.1.12.0 255.255.255.0**, which indicates that it was learned via EIGRP. Connected is always the most trustworthy; therefore, it is always used over other routing information. With regard to the 10.1.12.0/24 network, the output of **show bgp ipv4 unicast 10.1.12.0** in Example 18-32 indicates that it was learned from R2 and R3 using iBGP (*internal*), which has an AD of 200, much higher than EIGRP.

Example 18-31 Verifying AD of Routes in Routing Table

```
R5#show ip route 10.1.5.0 255.255.255.0
Routing entry for 10.1.5.0/24
  Known via "connected", distance 0, metric 0 (connected, via interface)
  ...output omitted...
```

```
R5#show ip route 10.1.12.0 255.255.255.0
Routing entry for 10.1.12.0/24
  Known via "eigrp 100", distance 90, metric 3328, type internal
...output omitted...
```

Example 18-32 Verifying Details of BGP Routes

```
R5#show bgp ipv4 unicast 10.1.12.0
BGP routing table entry for 10.1.12.0/24, version 50
Paths: (2 available, best #2, table default, RIB-failure(17))
  Not advertised to any peer
  Refresh Epoch 2
  65501
    3.3.3.3 (metric 131072) from 3.3.3.3 (3.3.3.3)
      Origin incomplete, metric 0, localpref 100, valid, internal
      rx pathid: 0, tx pathid: 0
  Refresh Epoch 2
  65501
    2.2.2.2 (metric 131072) from 2.2.2.2 (2.2.2.2)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      rx pathid: 0, tx pathid: 0x0
```

You can verify why a route is experiencing a RIB failure with the `show bgp ipv4 unicast rib-failure` command, as shown in Example 18-33. In this example, all three RIB failures are due to the BGP route having a higher AD.

Example 18-33 Verifying RIB Failures

```
R5#show bgp ipv4 unicast rib-failure
```

Network	Next Hop	RIB-failure	RIB-NH Matches
10.1.5.0/24	2.2.2.2	Higher admin distance	n/a
10.1.12.0/24	2.2.2.2	Higher admin distance	n/a
10.1.13.0/24	2.2.2.2	Higher admin distance	n/a

Route Filtering

The amount of control you have over routes in BGP is incredible—so much so, that we could dedicate an entire chapter to controlling BGP routes. However, that would be beyond the scope of the book and the TSHOOT exam. What we want to be able to do while troubleshooting missing routes is determine whether there is a route filter applied and if it is the cause of the missing routes. Example 18-34 displays the BGP table on R5 using the `show bgp ipv4 unicast` command. Notice that there is no entry for 10.1.13.0/24.

Example 18-34 Verifying Missing Routes on R5

```
R5#show bgp ipv4 unicast
BGP table version is 10, local router ID is 5.5.5.5
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
```

```

        r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
        x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

```

	Network	Next Hop	Metric	LocPrf	Weight	Path
* i	1.1.1.1/32	3.3.3.3	0	100	0	65501 ?
*>i		2.2.2.2	0	100	0	65501 ?
* i	10.1.1.0/26	3.3.3.3	0	100	0	65501 i
*>i		2.2.2.2	0	100	0	65501 i
* i	10.1.1.0/24	3.3.3.3	0	100	0	65501 i
*>i		2.2.2.2	0	100	0	65501 i
* i	10.1.1.64/26	3.3.3.3	0	100	0	65501 i
*>i		2.2.2.2	0	100	0	65501 i
* i	10.1.1.128/26	3.3.3.3	0	100	0	65501 i
*>i		2.2.2.2	0	100	0	65501 i
* i	10.1.1.192/26	3.3.3.3	0	100	0	65501 i
*>i		2.2.2.2	0	100	0	65501 i
r i	10.1.5.0/24	3.3.3.3	3328	100	0	i
r>i		2.2.2.2	3328	100	0	i
r i	10.1.12.0/24	3.3.3.3	0	100	0	65501 ?
r>i		2.2.2.2	0	100	0	65501 ?

However, let's see whether we are receiving the route from R2 or R3 using the **show bgp ipv4 unicast neighbors ip_address routes** command, as shown in Example 18-35. The output clearly shows that we are not learning 10.1.13.0/24. But wait, this command displays routes learned after local filters have been applied. Therefore, let's check to see whether R2 or R3 are advertising the 10.1.13.0/24 route before we check for filters. As shown in Example 18-36, which displays the output of the **show bgp ipv4 unicast neighbors ip_address advertised-routes** command, R2 and R3 are advertising the 10.1.13.0/24 network to R5.

Example 18-35 Verifying Whether Routes Are Being Received on R5

```

R5#show bgp ipv4 unicast neighbors 2.2.2.2 routes
BGP table version is 9, local router ID is 5.5.5.5
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

```

	Network	Next Hop	Metric	LocPrf	Weight	Path
*>i	1.1.1.1/32	2.2.2.2	0	100	0	65501 ?
*>i	10.1.1.0/26	2.2.2.2	0	100	0	65501 i
*>i	10.1.1.0/24	2.2.2.2	0	100	0	65501 i
*>i	10.1.1.64/26	2.2.2.2	0	100	0	65501 i

```
*>i 10.1.1.128/26    2.2.2.2            0    100    0 65501 i
*>i 10.1.1.192/26    2.2.2.2            0    100    0 65501 i
r>i 10.1.5.0/24      2.2.2.2            3328  100    0 i
r>i 10.1.12.0/24     2.2.2.2            0    100    0 65501 ?
```

Total number of prefixes 8

R5#show bgp ipv4 unicast neighbors 3.3.3.3 routes

BGP table version is 9, local router ID is 5.5.5.5

Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
x best-external, a additional-path, c RIB-compressed,

Origin codes: i - IGP, e - EGP, ? - incomplete

RPKI validation codes: V valid, I invalid, N Not found

	Network	Next Hop	Metric	LocPrf	Weight	Path
* i	1.1.1.1/32	3.3.3.3	0	100	0	65501 ?
* i	10.1.1.0/26	3.3.3.3	0	100	0	65501 i
* i	10.1.1.0/24	3.3.3.3	0	100	0	65501 i
* i	10.1.1.64/26	3.3.3.3	0	100	0	65501 i
* i	10.1.1.128/26	3.3.3.3	0	100	0	65501 i
* i	10.1.1.192/26	3.3.3.3	0	100	0	65501 i
r i	10.1.5.0/24	3.3.3.3	3328	100	0	i
r i	10.1.12.0/24	3.3.3.3	0	100	0	65501 ?

Total number of prefixes 8

Example 18-36 Verifying Whether Routes Are Being Sent to R5

R2#show bgp ipv4 unicast neighbors 5.5.5.5 advertised-routes

BGP table version is 10, local router ID is 2.2.2.2

Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
x best-external, a additional-path, c RIB-compressed,

Origin codes: i - IGP, e - EGP, ? - incomplete

RPKI validation codes: V valid, I invalid, N Not found

	Network	Next Hop	Metric	LocPrf	Weight	Path
r>	1.1.1.1/32	1.1.1.1	0		0	65501 ?
*>	10.1.1.0/26	1.1.1.1	0		0	65501 i
*>	10.1.1.0/24	1.1.1.1	0		0	65501 i
*>	10.1.1.64/26	1.1.1.1	0		0	65501 i
*>	10.1.1.128/26	1.1.1.1	0		0	65501 i
*>	10.1.1.192/26	1.1.1.1	0		0	65501 i
*>	10.1.5.0/24	10.1.24.4	3328		32768	i
r>	10.1.12.0/24	1.1.1.1	0		0	65501 ?
*>	10.1.13.0/24	1.1.1.1	0		0	65501 ?

Total number of prefixes 9

R3#show bgp ipv4 unicast neighbors 5.5.5.5 advertised-routes

BGP table version is 10, local router ID is 3.3.3.3

Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
x best-external, a additional-path, c RIB-compressed,

Origin codes: i - IGP, e - EGP, ? - incomplete

RPKI validation codes: V valid, I invalid, N Not found

	Network	Next Hop	Metric	LocPrf	Weight	Path
*>	1.1.1.1/32	10.1.13.1	0		0	65501 ?
*>	10.1.1.0/26	10.1.13.1	0		0	65501 i
*>	10.1.1.0/24	10.1.13.1	0		0	65501 i
*>	10.1.1.64/26	10.1.13.1	0		0	65501 i
*>	10.1.1.128/26	10.1.13.1	0		0	65501 i
*>	10.1.1.192/26	10.1.13.1	0		0	65501 i
*>	10.1.5.0/24	10.1.34.4	3328		32768	i
*>	10.1.12.0/24	10.1.13.1	0		0	65501 ?
r>	10.1.13.0/24	10.1.13.1	0		0	65501 ?

Total number of prefixes 9



Issuing the **show ip protocols** command as shown in Example 18-37 displays the incoming filter applied to the BGP autonomous system. It is a distribute list using the prefix list called **FILTER_10.1.13.0/24**, as shown in Example 18-37. The prefix list, as also shown in Example 18-37, is denying 10.1.13.0/24 and permitting all other routes.

Example 18-37 Verifying Whether Filters Are Applied to R5

R5#show ip protocols

...output omitted...

Routing Protocol is "bgp 65502"

Outgoing update filter list for all interfaces is not set

Incoming update filter list for all interfaces is (prefix-list) FILTER_10.1.13.0/24

IGP synchronization is disabled

Automatic route summarization is disabled

Neighbor(s):

Address	FiltIn	FiltOut	DistIn	DistOut	Weight	RouteMap
2.2.2.2						
3.3.3.3						

Maximum path: 1

Routing Information Sources:...output omitted...

R5#show ip prefix-list

ip prefix-list FILTER_10.1.13.0/24: 2 entries

```
seq 5 deny 10.1.13.0/24
seq 10 permit 0.0.0.0/0 le 32

R5#show run | include bgp 65502|distribute-list
router bgp 65502
distribute-list prefix FILTER_10.1.13.0/24 in
```

The example we just reviewed focused on a filter that applied to the entire BGP process. Therefore, no matter which router we receive the route 10.1.13.0/24 from, it would be denied. However, you could apply a filter directly to a neighbor using any one of the following commands:

- `neighbor ip_address distribute-list access_list_number {in | out}`
- `neighbor ip_address prefix-list prefix_list_name {in | out}`
- `neighbor ip_address route-map map_name {in | out}`
- `neighbor ip_address filter-list access_list_number {in | out}`

How do you verify whether a route filter is applied specifically to a neighbor? You would verify the route filters with the same **show** commands as before. You just have to look in a different spot in the output. Refer to Example 18-38. In this example, an inbound distribute list is applied directly to the neighbor 2.2.2.2, as shown in the **show ip protocols** output. Notice that only the first six characters of the ACL are identified. We then review the running configuration and see that the distribute list is using the ACL named `FILTER_10.1.13.0/24`. Using the **show ip access-list** command confirms that the router is denying the 10.1.13.0/24 network from 2.2.2.2 but allowing all other networks.

Example 18-38 Verifying a Distribute List Applied to a Neighbor

```
R5#show ip protocols
...output omitted...

Routing Protocol is "bgp 65502"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  IGP synchronization is disabled
  Automatic route summarization is disabled
  Neighbor(s):
    Address          FiltIn FiltOut DistIn DistOut Weight RouteMap
    2.2.2.2          FILTER
    3.3.3.3
  Maximum path: 1
  Routing Information Sources:
  ...output omitted...

R5#show run | include bgp 65502|distribute-list
router bgp 65502
```



```
neighbor 2.2.2.2 distribute-list FILTER_10.1.13.0/24 in
```

```
R5#show ip access-lists
```

```
Standard IP access list FILTER_10.1.13.0/24
 10 deny 10.1.13.0, wildcard bits 0.0.0.255
 20 permit any
```

As noted earlier, you can also apply a route map, a prefix list, and a filter list directly to the **neighbor** command. The filter list will appear under the **FiltIn** and **FiltOut** column in **show ip protocols**, and the route map will appear under the **RouteMap** column in **show ip protocols** output. If the prefix list is applied directly to a **neighbor** statement, it does not appear in the output of **show ip protocols**. You will need to review the output of **show bgp ipv4 unicast neighbors**. However, as you recall, it is an extremely verbose output. Therefore, here is a shortcut that you might want to remember for troubleshooting route filters:

show bgp ipv4 unicast neighbors ip_address | include prefix|filter|Route map

Example 18-39 shows a sample of what would appear in the output of **show bgp ipv4 unicast neighbors** on R5 based on different filters applied to and from neighbors R2 and R3. In the output, you can see that there is an inbound prefix list applied directly to neighbor 3.3.3.3 called **FILTER_10.1.13.0/24**; there is also an outbound route map called **FILTER_10.1.5.0/24** for routes sent to neighbor 3.3.3.3. With regard to neighbor 2.2.2.2, there is an inbound “network filter” (distribute list) applied to the **neighbor** statement that is using the ACL called **FILTER_10.1.13.0/24**, and also an inbound autonomous system path ACL called 25.

Example 18-39 Verifying Filters Applied to the neighbor Statements

```
R5#show bgp ipv4 unicast neighbors 3.3.3.3 | include prefix|filter|Route map
Incoming update prefix filter list is FILTER_10.1.13.0/24
Route map for outgoing advertisements is FILTER_10.1.5.0/24
R5#show bgp ipv4 unicast neighbors 2.2.2.2 | include prefix|filter|Route map
Incoming update network filter list is FILTER_10.1.13.0/24
Incoming update AS path filter list is 25
```

Troubleshooting BGP Path Selection

Unlike OSPF and EIGRP, BGP does not consider a link’s bandwidth when making a route decision. Instead, BGP uses various attributes when deciding which path is the best. When troubleshooting BGP paths, you need to have a solid understanding of all the attributes to fully comprehend why BGP made the decision it made. This section discusses the BGP best path decision-making process. In addition, we examine private autonomous system numbers.

Understanding the Best Path Decision-Making Process

The following list presents the order that BGP reviews the attributes when deciding which path is the best:

**Key
Topic**

1. Prefer highest *weight*
2. Prefer highest *local preference*
3. Prefer route *originated* by the local router
4. Prefer shortest *autonomous system path*
5. Prefer lowest *origin* code
6. Prefer lowest *MED* (metric)
7. Prefer *external* over *internal* path
8. Prefer path through *closest IGP neighbor*
9. Prefer *oldest route* for eBGP paths
10. Prefer path with the lowest *neighbor BGP RID*
11. Prefer path with the lowest *neighbor IP address*

As you go through the BGP best path decision-making process, refer to Figure 18-6 and the output of `show bgp ipv4 unicast 10.1.1.0` on R5 in Example 18-40.

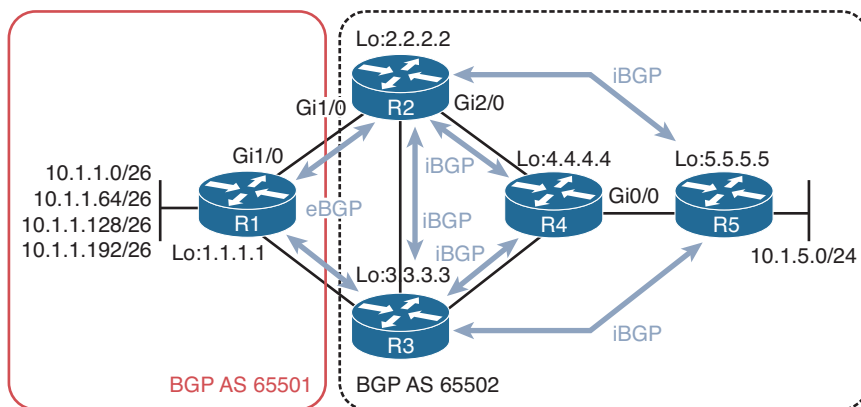


Figure 18-6 Understanding the BGP Best Path Decision Process Topology

Example 18-40 Verifying the BGP Table on Router R5 for Network 10.1.1.0

```
R5#show bgp ipv4 unicast 10.1.1.0
BGP routing table entry for 10.1.1.0/26, version 46
Paths: (2 available, best #1, table default)
  Not advertised to any peer
  Refresh Epoch 4
  65501
    2.2.2.2 (metric 131072) from 2.2.2.2 (2.2.2.2)
```

```

Origin IGP, metric 0, localpref 100, valid, internal, best
rx pathid: 0, tx pathid: 0x0
Refresh Epoch 1
65501
3.3.3.3 (metric 131072) from 3.3.3.3 (3.3.3.3)
Origin IGP, metric 0, localpref 100, valid, internal
rx pathid: 0, tx pathid: 0

```

Once BGP finds a match, it stops and uses that attribute as the reason for choosing the path as the best. It looks no further. In addition, if the next-hop IP address is not reachable, the router does not even go through this process because it considers the next-hop inaccessible:

1. BGP first looks at weight. Higher is better. In Example 18-40, no weight is listed because both paths are using the default value of 0. Therefore, weight is tied and the next attribute is checked.
2. Local preference is checked next. Higher is better. In Example 18-40, localpref is 100 (default) for both paths; therefore, local preference is tied and the next attribute is checked.
3. The router checks whether it generated the BGP route (has a next hop of 0.0.0.0). If it did, it is preferred. In Example 18-40, the next hops are 2.2.2.2 and 3.3.3.3 on the far left of the output. Therefore, R5 did not generate any of the routes, and the next attribute is checked.
4. The autonomous system path is checked next. The shortest path is preferred. In Example 18-40, the autonomous system path is 65501 for both. Therefore, the autonomous system path is tied, and the next attribute is checked.
5. The origin code is checked next. IGP is better than EGP, which is better than incomplete. Note that this is not related to iBGP versus eBGP, which is covered later. IGP means the route was generated with the **network mask** or **summary-address** command, and incomplete means the route was redistributed into BGP. EGP means it was generated from EGP, the predecessor to BGP. In Example 18-40, the origin is IGP for both which means that the next attribute will be checked.
6. MED (metric) is next. Lower is better. In Example 18-40, the MED (metric) is the same for both (0). Therefore, the next attribute has to be checked.
7. Now eBGP is preferred over iBGP. In Example 18-40, they are both learned via iBGP (internal). Therefore, this attribute is tied as well and the next will have to be checked.
8. The IGP path to the neighbor is compared now. In Example 18-40, the IGP path to 2.2.2.2 has a metric of 131072, and the IGP path to 3.3.3.3 has a metric of 131072. They are tied. Therefore, the next attribute has to be checked.
9. If they are eBGP paths, the age of the routes are checked. In Example 18-40, both paths are iBGP paths. Therefore, we skip this attribute and move on to the next attribute.

- 10.** The BGP RIDs are now compared. Lower is better. In Example 18-40, neighbor 2.2.2.2 has a RID of 2.2.2.2 (as displayed in the brackets), and neighbor 3.3.3.3 has a RID of 3.3.3.3 (as displayed in the brackets). Which RID is lower? 2.2.2.2 Therefore, the route provided by the neighbor with the RID of 2.2.2.2 is considered the best path. If the RID happens to be tied, the neighbor IP address is used to break the tie.

Now it is your turn! Try the following on your own, and then we will walk you through it. Refer to Figure 18-7 and Example 18-41 and determine which attribute R2 is using to choose the best path to reach 10.1.1.128.

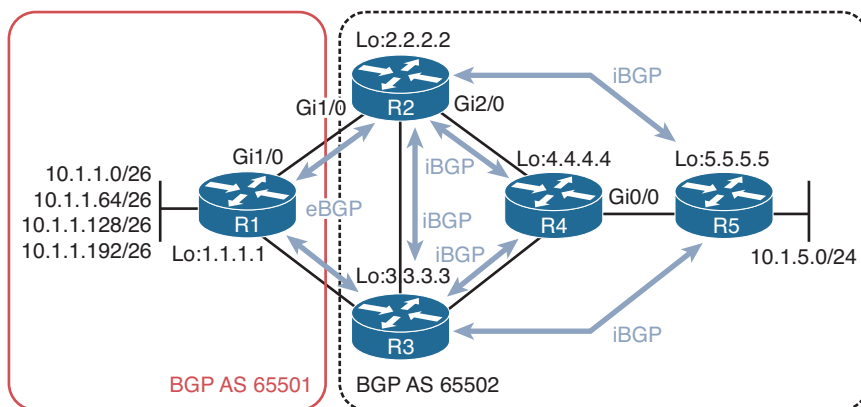


Figure 18-7 Practicing the BGP Best Path Decision Process Topology

Example 18-41 Practicing the BGP Best Path Decision Process

```
R2#show bgp ipv4 unicast 10.1.1.128
BGP routing table entry for 10.1.1.128/26, version 6
Paths: (2 available, best #2, table default)
  Advertised to update-groups:
    2
  Refresh Epoch 2
  65501
    3.3.3.3 (metric 131072) from 3.3.3.3 (3.3.3.3)
      Origin IGP, metric 0, localpref 100, valid, internal
      rx pathid: 0, tx pathid: 0
  Refresh Epoch 3
  65501
    1.1.1.1 from 1.1.1.1 (1.1.1.1)
      Origin IGP, metric 0, localpref 100, valid, external, best
      rx pathid: 0, tx pathid: 0x0
```

Alright, let's walk through it together:

1. Prefer highest *weight* Tied
2. Prefer highest *local preference* Tied

3. Prefer route *originated* by the local router **None**
4. Prefer shortest *autonomous system path* **Same at 65501**
5. Prefer lowest *origin* code **Same**
6. Prefer lowest *MED* (Metric) **Tied at 0**
7. Prefer *external* (eBGP) over *internal* (iBGP) path **Not Tied Stop**

The path learned from neighbor 1.1.1.1 is external (eBGP) and the path learned from neighbor 3.3.3.3 is internal (iBGP). Therefore, the path learned from neighbor 1.1.1.1 is preferred because external is preferred over internal.

If you are not getting desired paths, or the paths you expect to be used as *best*, you need to be able to walk through this process while troubleshooting to figure out why the current best path was chosen as such. There may have been an attribute that was modified locally or remotely at some point that is influencing the decision that is being made. You need to be able to recognize this and then manipulate the paths in your favor by modifying the necessary attributes.

Private Autonomous System Numbers

Like IPv4 addresses, BGP autonomous system numbers also have a private range. In the 2-byte autonomous system range it is 64,512 to 65,534, and for the 4-byte autonomous system range, it is 4,200,000,000 to 4,294,967,294. These autonomous system numbers can be used for networks that are single-homed or dual-homed to the same ISP, thereby preserving the public autonomous system numbers for those networks that are multi-homed to multiple ISPs.

Although the private autonomous system numbers can be used in the customer's network, it is imperative that the autonomous system number is not in the AS_PATH attribute when the routes are advertised to the Internet (global BGP table) because multiple autonomous systems could be using the same private autonomous system number which would then cause issues on the Internet.

If private autonomous system numbers are being sent into the global BGP table, they need to be stopped. You can accomplish this with the **neighbor ip_address remove-private-as** command.

Using debug Commands

The majority of changes that occur with BGP will generate syslog messages in real time. Therefore, you will be notified via syslog if any neighbor issues occur. So, unless you really need to, avoid using the large number of debugs that are available because they place a large amount of pressure on the routers' resources. Only use as a last resort! Following you will find a few **debug** commands that might be useful. However, up to this point, all the **show** commands we have covered and your knowledge can determine the same thing.

Example 18-42 provides sample output from the **debug ip routing** command. The output from this command shows updates to a router's IP routing table. In this example, the Loopback 0 interface (with an IP address of 10.3.3.3) of a neighboring router was administratively shut down and then administratively brought back up. As the 10.3.3.3/32 network became unavailable and then once again became available, you can see that the 10.3.3.3/32 route was deleted and then added to this router's IP routing table. Notice that this output is not specific to BGP. Therefore, you can use the **debug ip routing** command with routing processes other than BGP.

Example 18-42 debug ip routing Command Output

```
R2#debug ip routing
IP routing debugging is on
RT: 10.3.3.3/32 gateway changed from 172.16.1.1 to 172.16.2.2
RT: NET-RED 10.3.3.3/32
RT: del 10.3.3.3/32 via 172.16.2.2, bgp metric [20/0]
RT: delete subnet route to 10.3.3.3/32
RT: NET-RED 10.3.3.3/32
RT: SET_LAST_RDB for 10.3.3.3/32
    NEW rdb: via 172.16.1.1

RT: add 10.3.3.3/32 via 172.16.1.1, bgp metric [20/0]
RT: NET-RED 10.3.3.3/32
```

Example 18-43 provides sample output from the **debug ip bgp** command. The output of this command does not show the contents of BGP updates; however, this command can be useful in watching real-time state changes for IPv4 BGP peering relationships. In this example, you can see a peering session being closed for the neighbor with an IP address of 172.16.1.1.

Example 18-43 debug ip bgp Command Output

```
R2#debug ip bgp
BGP debugging is on for address family: IPv4 Unicast
*Mar 1 00:23:26.535: BGP: 172.16.1.1 remote close, state CLOSEWAIT
*Mar 1 00:23:26.535: BGP: 172.16.1.1 -reset the session
*Mar 1 00:23:26.543: BGPNSF state: 172.16.1.1 went from nsf_not_active to
    nsf_not_active
*Mar 1 00:23:26.547: BGP: 172.16.1.1 went from Established to Idle
*Mar 1 00:23:26.547: %BGP-5-ADJCHANGE: neighbor 172.16.1.1 Down Peer closed the
    session
*Mar 1 00:23:26.547: BGP: 172.16.1.1 closing
*Mar 1 00:23:26.651: BGP: 172.16.1.1 went from Idle to Active
*Mar 1 00:23:26.663: BGP: 172.16.1.1 open active delayed 30162ms (35000ms max,
    28% jitter)
```

Example 18-44 provides sample output from the **debug ip bgp updates** command. This command produces more detailed output than the **debug ip bgp** command. Specifically, you can see the content of IPv4 BGP updates. In this example, you see a route of 10.3.3.3/32 being added to a router's IP routing table.

Example 18-44 `debug ip bgp updates` Command Output

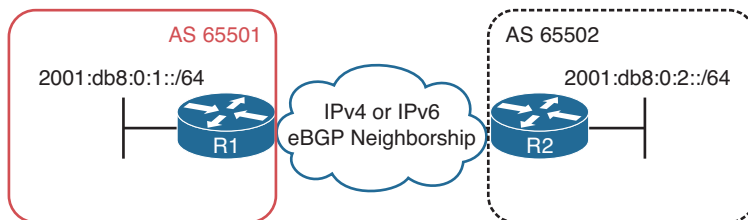
```

R2#debug ip bgp updates
BGP updates debugging is on for address family: IPv4 Unicast
*Mar  1 00:24:27.455: BGP(0): 172.16.1.1 NEXT_HOP part 1 net 10.3.3.3/32, next
    172.16.1.1
*Mar  1 00:24:27.455: BGP(0): 172.16.1.1 send UPDATE (format) 10.3.3.3/32, next
    172.16.1.1, metric 0, path 65002
*Mar  1 00:24:27.507: BGP(0): 172.16.1.1 rcv UPDATE about 10.3.3.3/32 - withdrawn
*Mar  1 00:24:27.515: BGP(0): Revise route installing 1 of 1 routes for
    10.3.3.3/32 -> 172.16.2.2(main) to main IP table
*Mar  1 00:24:27.519: BGP(0): updgrp 1 - 172.16.1.1 updates replicated for
    neighbors: 172.16.2.2
*Mar  1 00:24:27.523: BGP(0): 172.16.1.1 send UPDATE (format) 10.3.3.3/32, next
    172.16.1.2, metric 0, path 65003 65002
*Mar  1 00:24:27.547: BGP(0): 172.16.2.2 rcvd UPDATE w/ attr: nexthop 172.16.2.2,
    origin i, path 65003 65002
*Mar  1 00:24:27.551: BGP(0): 172.16.2.2 rcvd 10.3.3.3/32...duplicate ignored
*Mar  1 00:24:27.555: BGP(0): updgrp 1 - 172.16.1.1 updates replicated for
    neighbors: 172.16.2.2
*Mar  1 00:24:27.675: BGP(0): 172.16.2.2 rcv UPDATE w/ attr: nexthop 172.16.2.2,
    origin i, originator 0.0.0.0, path 65003 65001 65002, community, extended
    community
*Mar  1 00:24:27.683: BGP(0): 172.16.2.2 rcv UPDATE about 10.3.3.3/32 - DENIED
    due to: AS-PATH contains our own AS;
...OUTPUT OMITTED...

```

Troubleshooting BGP for IPv6

BGP for IPv4 and BGP for IPv6 are configured in the same BGP autonomous system configuration mode. This is known as Multiprotocol BGP, or MP-BGP for short. Implementing BGP for IPv4 and IPv6 on the same router requires the use of address families and the activation of neighbors for those address families. This section examines the additional issues (on top of what was already covered thus far in the chapter) that you may encounter when using MP-BGP with IPv4 and IPv6 unicast routes. Refer to Figure 18-8 while reviewing this section.

**Figure 18-8** MP-BGP Topology

There are two different ways to exchange IPv6 routes with BGP. You can exchange them over IPv4 TCP sessions or IPv6 TCP sessions. Example 18-45 displays a sample BGP configuration where IPv6 routes are exchanged over an IPv4 TCP session.

Notice how there are two address families: one for IPv4 unicast, and one for IPv6 unicast. The neighbors and remote autonomous system numbers are identified outside of the address family (AF) configuration. You then activate the neighbor within the AF with the **neighbor ip_address activate** command. In this example, the IPv6 AF is using an IPv4 neighbor address to establish the TCP session. Therefore, the TCP session will be IPv4 based. Reviewing the output of **show bgp ipv6 unicast summary**, as shown in Example 18-46, shows the IPv6 unicast AF neighbor adjacency that has been formed with router 2.2.2.2. Notice that the adjacency has been formed with an IPv4 unicast address. It also states that one IPv6 prefix has been learned from the neighbor.

Example 18-45 *MP-BGP Configuration for IPv6 Routes Over IPv4 TCP Session*

```
R1#show run | s router bgp
router bgp 65501
  bgp log-neighbor-changes
  neighbor 2.2.2.2 remote-as 65502
  neighbor 2.2.2.2 ebgp-multihop 2
  neighbor 2.2.2.2 password CISCO
  neighbor 2.2.2.2 update-source Loopback0
!
address-family ipv4
  network 10.1.1.0 mask 255.255.255.192
  network 10.1.1.64 mask 255.255.255.192
  network 10.1.1.128 mask 255.255.255.192
  network 10.1.1.192 mask 255.255.255.192
  aggregate-address 10.1.1.0 255.255.255.0
  redistribute connected
  neighbor 2.2.2.2 activate
  exit-address-family
!
address-family ipv6
  network 2001:DB8:1::/64
  neighbor 2.2.2.2 activate
  exit-address-family
```

Example 18-46 *Verifying MP-BGP IPv6 Unicast Neighbor Adjacencies*

```
R1#show bgp ipv6 unicast summary
BGP router identifier 1.1.1.1, local AS number 65501
BGP table version is 2, main routing table version 2
2 network entries using 336 bytes of memory
2 path entries using 208 bytes of memory
2/1 BGP path/bestpath attribute entries using 272 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
```



```

0 BGP filter-list cache entries using 0 bytes of memory
BGP using 840 total bytes of memory
BGP activity 11/0 prefixes, 18/6 paths, scan interval 60 secs

Neighbor      V      AS MsgRcvd MsgSent  TblVer  InQ OutQ Up/Down  State/PfxRcd
2.2.2.2        4      65502    25    25      2    0   0 00:12:02      1

```

Key Topic

To verify the IPv6 unicast routes that have been learned from all neighbors, you can issue the **show bgp ipv6 unicast** command, as shown in Example 18-47. This displays the IPv6 BGP table. The route 2001:db8:1::/64 is locally originated because of the next hop ::, and it is in the routing table as indicated by the *> at the beginning of the entry. Examine the 2001:db8:2::/64 route. This is the route that was learned from R2 (the 2.2.2.2 neighbor). It is not installed in the routing table as indicated by the absence of the *>. The reason is because the next hop is not reachable. The address ::FFFF:2.2.2.2 is a dynamically generated next hop that was created to replace the original next hop of 2.2.2.2. This occurs because an IPv6 route cannot have an IPv4 next-hop address. Why was the next hop an IPv4 address? This is because the adjacency is an IPv4 adjacency for the IPv6 AF.

Example 18-47 Verifying MP-BGP IPv6 Unicast Routes in the IPv6 BGP Table

```

R1#show bgp ipv6 unicast
BGP table version is 2, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network          Next Hop           Metric LocPrf Weight Path
*>  2001:DB8:1::/64  ::                0           32768 i
*   2001:DB8:2::/64  ::FFFF:2.2.2.2    0           0 65502 i

```

Key Topic

To solve this issue, you need to create a route map that will change the next hop to a valid IPv6 address and attach it to the **neighbor** statement. Now, be very careful with this. It has to be *done on the router advertising the route*, not receiving the route. In Example 18-48, a route map is configured on R2 that changes the next-hop address to 2001:db8:12::2. The route map is then attached to the neighbor 1.1.1.1 outbound.

Example 18-48 Modifying the BGP Next Hop

```

R2#config t
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#route-map CHANGE_NH permit 10
R2(config-route-map)#set ipv6 next-hop 2001:db8:12::2
R2(config-route-map)#exit
R2(config)#router bgp 65502
R2(config-router)#address-family ipv6 unicast
R2(config-router-af)#neighbor 1.1.1.1 route-map CHANGE_NH out

```

When you examine the output of **show bgp ipv6 unicast** again in Example 18-49, the next hop is now a valid hop, and the route is installed in the table.

Example 18-49 Verifying the BGP Next Hop

```
R1#show bgp ipv6 unicast
BGP table version is 3, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 2001:DB8:1::/64	::	0		32768	i
*> 2001:DB8:2::/64	2001:DB8:12::2	0		0	65502 i

When forming IPv6 TCP sessions and neighbor relationships, you do not have to worry about the issue just described. However, you have to make sure that you define the IPv6 neighbor and activate it. Take a look at Example 18-50. To form the IPv6 TCP session, you define the neighbor with the **neighbor ipv6_address remote-as autonomous_system_number** command outside of the AF configuration, and then you activate the neighbor in the IPv6 AF configuration with the **neighbor ipv6_address activate** command.

Example 18-50 MP-BGP Configuration for IPv6 Routes over IPv6 TCP Session

```
R1#show run | section router bgp
router bgp 65501
  bgp log-neighbor-changes
  neighbor 2.2.2.2 remote-as 65502
  neighbor 2.2.2.2 ebgp-multihop 2
  neighbor 2.2.2.2 password CISCO
  neighbor 2.2.2.2 update-source Loopback0
  neighbor 10.1.13.3 remote-as 65502
  neighbor 2001:DB8:12::2 remote-as 65502
!
address-family ipv4
  network 10.1.1.0 mask 255.255.255.192
  network 10.1.1.64 mask 255.255.255.192
  network 10.1.1.128 mask 255.255.255.192
  network 10.1.1.192 mask 255.255.255.192
  aggregate-address 10.1.1.0 255.255.255.0
  redistribute connected
  neighbor 2.2.2.2 activate
  neighbor 10.1.13.3 activate
  no neighbor 2001:DB8:12::2 activate
exit-address-family
!
```

```

address-family ipv6
  network 2001:DB8:1::/64
  neighbor 2001:DB8:12::2 activate
exit-address-family

```

The output of **show bgp ipv6 unicast summary** as shown in Example 18-51 shows that R1 has formed an IPv6 BGP neighbor adjacency with the device at 2001:db8:12::2 using an IPv6 TCP session, and one prefix has been received. The IPv6 BGP table, as displayed in the output of **show bgp ipv6 unicast** command in Example 18-52, indicates that 2001:DB8:2::/64 can be reached with a next hop of 2001:DB8:12::2 and that it is installed in the routing table, as indicated by the *>.

Example 18-51 MP-BGP Adjacencies with IPv6 TCP Sessions

```

R1#show bgp ipv6 unicast summary
BGP router identifier 1.1.1.1, local AS number 65501
BGP table version is 5, main routing table version 5
2 network entries using 336 bytes of memory
2 path entries using 208 bytes of memory
2/2 BGP path/bestpath attribute entries using 272 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 840 total bytes of memory
BGP activity 12/1 prefixes, 22/10 paths, scan interval 60 secs

```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
2001:DB8:12::2	4	65502	5	5	4	0	0	00:00:05	1

Example 18-52 Verifying IPv6 BGP Table

```

R1#show bgp ipv6 unicast
BGP table version is 5, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 2001:DB8:1::/64	::	0		32768	i
*> 2001:DB8:2::/64	2001:DB8:12::2	0		0	65502 i

BGP Trouble Tickets

This section presents various trouble tickets relating to the topics discussed earlier in the chapter. The purpose of these trouble tickets is to give a process that you can follow

when troubleshooting in the real world or in an exam environment. All trouble tickets in this section are based on the topology depicted in Figure 18-9.

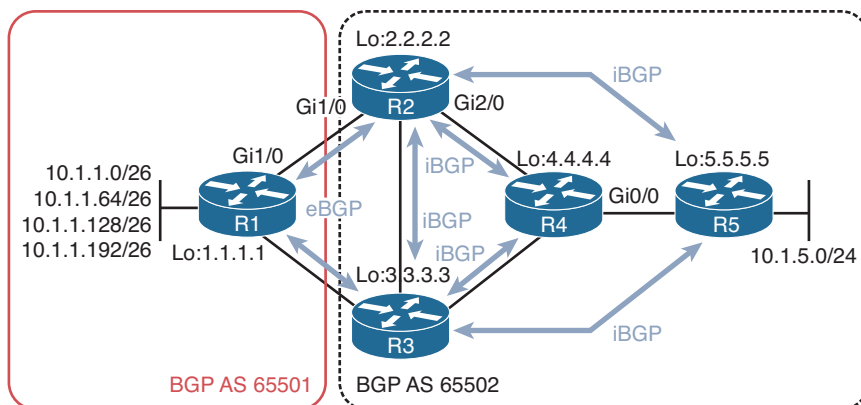


Figure 18-9 BGP Trouble Tickets Topology

Trouble Ticket 18-1

Problem: You are the administrator for BGP autonomous system 65502. While you were away on vacation, the link between R1 and R2 failed. When the link between R1 and R2 fails, the link between R1 and R3 is supposed to forward traffic to BGP autonomous system 65501. However, that did not occur while you were away. Your co-worker had to restore connectivity between R1 and R2 while complaints kept flowing in from the users in 10.1.5.0/24 about connectivity to the 10.1.1.0/24 networks being down.

At this point, connectivity is fine. You confirm this by pinging from a PC in 10.1.5.0/24 to 10.1.1.10. In Example 18-53, the ping is successful. Because it is the middle of the day, you cannot bring down the link between R1 and R2 to re-create the issue because it will disrupt the network users. Therefore, you need to be creative with your troubleshooting efforts.

Example 18-53 Verifying Connectivity

```
C:\>ping 10.1.1.10

Pinging 10.1.1.10 with 32 bytes of data:

Reply from 10.1.1.10: bytes=32 time 1ms TTL=128
Reply from 10.1.1.10: bytes=32 time 1ms TTL=128
Reply from 10.1.1.10: bytes=32 time 1ms TTL=128
Reply from 10.1.1.10: bytes=32 time 1ms TTL=128

Ping statistics for 10.1.1.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

For router R5 to know about the networks in autonomous system 65501, they have to be advertised to it. The best place to see whether R5 is learning about the routes is R5's BGP table. Based on the network topology, R5 should be learning about the networks from R2 and R3. In Example 18-54, the output of **show bgp ipv4 unicast** is displayed. As you can see from the next-hop column, all valid routes to the 10.1.1.x/26 networks are via the next hop of 2.2.2.2, which is R2. There are no entries for R3 at 3.3.3.3 that are valid for those networks.

Example 18-54 Examining R5's BGP Table

```
R5#show bgp ipv4 unicast
BGP table version is 56, local router ID is 5.5.5.5
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*>i 1.1.1.1/32	2.2.2.2	0	100	0	65501 ?
*>i 10.1.1.0/26	2.2.2.2	0	100	0	65501 i
*>i 10.1.1.64/26	2.2.2.2	0	100	0	65501 i
*>i 10.1.1.128/26	2.2.2.2	0	100	0	65501 i
*>i 10.1.1.192/26	2.2.2.2	0	100	0	65501 i
r>i 10.1.5.0/24	2.2.2.2	3328	100	0	i
r i	3.3.3.3	3328	100	0	i
r>i 10.1.12.0/24	2.2.2.2	0	100	0	65501 ?
r>i 10.1.13.0/24	2.2.2.2	0	100	0	65501 ?

Next you want to confirm whether R5 is even receiving the routes from R3. Therefore, you issue the command **show bgp ipv4 unicast neighbors 2.2.2.2 routes** and **show bgp ipv4 unicast neighbors 3.3.3.3 routes** to determine which routes are being received and to compare what is being advertised from R2 versus R3. The output in Example 18-55 clearly shows that R5 is not receiving any routes about the 10.1.1.x/26 networks from R3. This is the reason why network connectivity was lost when the link between R1 and R2 went down. R5 does not have any route information from R3.

Example 18-55 Examining Routes Received from R2 and R3

```
R5#show bgp ipv4 unicast neighbors 2.2.2.2 routes
BGP table version is 56, local router ID is 5.5.5.5
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

Network	Next Hop	Metric	LocPrf	Weight	Path

```

*>i 1.1.1.1/32      2.2.2.2      0      100      0 65501 ?
*>i 10.1.1.0/26     2.2.2.2      0      100      0 65501 i
*>i 10.1.1.64/26    2.2.2.2      0      100      0 65501 i
*>i 10.1.1.128/26   2.2.2.2      0      100      0 65501 i
*>i 10.1.1.192/26   2.2.2.2      0      100      0 65501 i
r>i 10.1.5.0/24     2.2.2.2      3328    100      0 i
r>i 10.1.12.0/24     2.2.2.2      0      100      0 65501 ?
r>i 10.1.13.0/24     2.2.2.2      0      100      0 65501 ?

Total number of prefixes 8
R5#show bgp ipv4 unicast neighbors 3.3.3.3 routes
BGP table version is 56, local router ID is 5.5.5.5
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

      Network      Next Hop      Metric LocPrf Weight Path
r i 10.1.5.0/24    3.3.3.3      3328     100      0 i

Total number of prefixes 1

```

You access R3 and issue the **show bgp ipv4 unicast neighbors 5.5.5.5 advertised-routes** command to determine which routes, if any, R3 is sending to R5. In Example 18-56 you can see that there are no routes related to the 10.1.1.x/26 networks being advertised to R5. So, that raises the question, does R3 even know about the networks?

Example 18-56 Examining Routes Sent from R3 to R5

```

R3#show bgp ipv4 unicast neighbors 5.5.5.5 advertised-routes
BGP table version is 108, local router ID is 3.3.3.3
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

      Network      Next Hop      Metric LocPrf Weight Path
*> 10.1.5.0/24     10.1.34.4     3328      32768 i

Total number of prefixes 1

```

On R3, you issue the command **show ip route 10.1.1.0 255.255.255.0 longer-prefixes**, as shown in Example 18-57, and confirm that the networks are learned via BGP. However, you also notice something else that is strange. The AD is 200, which is the value associated with iBGP-learned routes and the next hop is via 2.2.2.2, which is R2. The AD should be 20 for eBGP, and the next hop should be R1's IP in this case.

Example 18-57 *Examining BGP Routes in R3's Routing Table*

```
R3#show ip route 10.1.1.0 255.255.255.0 longer-prefixes
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 14 subnets, 3 masks
B       10.1.1.0/26 [200/0] via 2.2.2.2, 00:09:07
B       10.1.1.64/26 [200/0] via 2.2.2.2, 00:09:07
B       10.1.1.128/26 [200/0] via 2.2.2.2, 00:09:07
B       10.1.1.192/26 [200/0] via 2.2.2.2, 00:09:07
```

You issue the command **show bgp ipv4 unicast** on R3 to check the BGP table, as shown in Example 18-58. Based on the output, only R2 and R4 are next hops for routes. R1 is not a next hop for any of them.

Example 18-58 *Examining BGP Routes in R3's BGP Table*

```
R3#show bgp ipv4 unicast
BGP table version is 108, local router ID is 3.3.3.3
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*>i 1.1.1.1/32	2.2.2.2	0	100	0	65501 ?
*>i 10.1.1.0/26	2.2.2.2	0	100	0	65501 i
*>i 10.1.1.0/24	2.2.2.2	0	100	0	65501 i
*>i 10.1.1.64/26	2.2.2.2	0	100	0	65501 i
*>i 10.1.1.128/26	2.2.2.2	0	100	0	65501 i
*>i 10.1.1.192/26	2.2.2.2	0	100	0	65501 i
* i 10.1.5.0/24	2.2.2.2	3328	100	0	i
*>	10.1.34.4	3328		32768	i
r>i 10.1.12.0/24	2.2.2.2	0	100	0	65501 ?
r>i 10.1.13.0/24	2.2.2.2	0	100	0	65501 ?

Issuing the **show bgp ipv4 unicast neighbors 10.1.13.1 routes** command on R3 confirms that no routes are being received from R1, as shown in Example 18-59.

Example 18-59 *Verifying Routes Learned from R1*

```
R3#show bgp ipv4 unicast neighbors 10.1.13.1 routes
```

```
Total number of prefixes 0
```

Because R1 is not in your autonomous system, you cannot access it for troubleshooting purposes. Therefore, you will need to call the admin in autonomous system 65501. However, do not do that just yet. We can check many more items on R3. For example, to learn BGP routes, you need a BGP adjacency. To confirm that R3 is a neighbor with R1, you issue the **show bgp ipv4 unicast summary** command, as shown in Example 18-60. Based on the output, R1 and R3 are not neighbors because the state is listed as idle. You think you have found the issue.

Example 18-60 *Verifying Neighbor Adjacency Between R1 and R3*

```
R3#show bgp ipv4 unicast summary
```

```
BGP router identifier 3.3.3.3, local AS number 65502
BGP table version is 108, main routing table version 108
9 network entries using 1296 bytes of memory
10 path entries using 800 bytes of memory
5/4 BGP path/bestpath attribute entries using 680 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 2800 total bytes of memory
BGP activity 17/8 prefixes, 71/61 paths, scan interval 60 secs
```

Neighbor PfxRcd	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/
2.2.2.2	4	65502	34	34	108	0	0	00:24:29	9
4.4.4.4	4	65502	47	48	108	0	0	00:39:00	0
5.5.5.5	4	65502	5	6	108	0	0	00:00:18	0
10.1.13.1	4	65510	0	0	1	0	0	never	Idle

Comparing the output in Example 18-60 to your network documentation (Figure 18-9), you notice that the autonomous system number is incorrect for 10.1.13.1. It is listed as 65510 when it should be 65501. To fix the issue, you remove the current **neighbor remote-as** statement and add the correct one, as shown in Example 18-61. Once the changes are made, the neighbor relationship is up.

Example 18-61 *Modifying the neighbor remote-as Statement*

```
R3#config t
```

```
Enter configuration commands, one per line. End with CNTL/Z.
```

```
R3(config)#router bgp 65502
```

```
R3(config-router)#no neighbor 10.1.13.1 remote-as 65510
```

```
R3(config-router)#neighbor 10.1.13.1 remote-as 65501
```

```
%BGP-5-ADJCHANGE: neighbor 10.1.13.1 Up
```

```
R3(config-router)#
```


To confirm that everything is fine, you access R5 and issue the **show bgp ipv4 unicast** command and confirm that routes from R2 and R3 are now listed in the BGP table, as shown in Example 18-62. Issue solved. After hours, you will bring down the link between R1 and R2 and confirm that traffic successfully flows between R3 and R1.

Example 18-62 *Confirming R5 Knows Routes from R2 and R3*

```
R5#show bgp ipv4 unicast
BGP table version is 56, local router ID is 5.5.5.5
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

Network	Next Hop	Metric	LocPrf	Weight	Path
* i 1.1.1.1/32	3.3.3.3	0	100	0	65501 ?
*>i	2.2.2.2	0	100	0	65501 ?
* i 10.1.1.0/26	3.3.3.3	0	100	0	65501 i
*>i	2.2.2.2	0	100	0	65501 i
* i 10.1.1.64/26	3.3.3.3	0	100	0	65501 i
*>i	2.2.2.2	0	100	0	65501 i
* i 10.1.1.128/26	3.3.3.3	0	100	0	65501 i
*>i	2.2.2.2	0	100	0	65501 i
* i 10.1.1.192/26	3.3.3.3	0	100	0	65501 i
*>i	2.2.2.2	0	100	0	65501 i
r i 10.1.5.0/24	3.3.3.3	3328	100	0	i
r>i	2.2.2.2	3328	100	0	i

Network	Next Hop	Metric	LocPrf	Weight	Path
r i 10.1.12.0/24	3.3.3.3	0	100	0	65501 ?
r>i	2.2.2.2	0	100	0	65501 ?
r i 10.1.13.0/24	3.3.3.3	0	100	0	65501 ?
r>i	2.2.2.2	0	100	0	65501 ?

With a little bit of spare time on your hands, you decide to check the log files from R3. You notice the following BGP message listed many times:

```
%BGP-3-NOTIFICATION: sent to neighbor 10.1.13.1 passive 2/2 (peer in wrong AS)
2 bytes FFDD
```

The syslog message clearly states that the peer is in the wrong autonomous system. Never forget to check your log files before you troubleshoot. It can save you valuable time.

Trouble Ticket 18-2

Problem: You are the administrator for BGP autonomous system 65501. Users in the 10.1.1.0/26 and 10.1.1.64/26 networks have indicated that they are not able to access resources located at 10.1.5.5. However, they can access resources locally.

You begin troubleshooting by issuing two pings on R1 to 10.1.5.5 and sourcing them from 10.1.1.1 and 10.1.1.65. As shown in Example 18-63, the pings fail.

Example 18-63 *Verifying Issue with a Ping*

```
R1#ping 10.1.5.5 source 10.1.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.5.5, timeout is 2 seconds:
Packet sent with a source address of 10.1.1.1
.....
Success rate is 0 percent (0/5)
R1#ping 10.1.5.5 source 10.1.1.65
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.5.5, timeout is 2 seconds:
Packet sent with a source address of 10.1.1.65
.....
Success rate is 0 percent (0/5)
```

You confirm with the command **show ip route 10.1.5.5** on R1, as shown in Example 18-64, that there is a route to 10.1.5.5 via R2 learned via BGP.

Example 18-64 *Confirming R1 Has a Route to 10.1.5.5*

```
R1#show ip route 10.1.5.5
Routing entry for 10.1.5.0/24
  Known via "bgp 65501", distance 20, metric 3328
  Tag 65502, type external
  Last update from 2.2.2.2 00:12:35 ago
  Routing Descriptor Blocks:
  * 2.2.2.2, from 2.2.2.2, 00:12:35 ago
    Route metric is 3328, traffic share count is 1
    AS Hops 1
    Route tag 65502
    MPLS label: none
```

You would like to see how far the packets are traveling to get a rough idea of where they might be failing. Therefore, you decide to issue an extended traceroute to hopefully gather some additional information. In Example 18-65, you can see that the trace is failing at the next hop router (R2).

Example 18-65 *Identifying How Far Packets Are Traveling Before They Fail*

```
R1#traceroute 10.1.5.5 source 10.1.1.1
Type escape sequence to abort.
Tracing the route to 10.1.5.5
VRF info: (vrf in name/id, vrf out name/id)
 1 10.1.12.2 40 msec 44 msec 28 msec
 2 * * *
 3 * * *
```

```

4 * * *
...output omitted...
R1#traceroute 10.1.5.5 source 10.1.1.65
Type escape sequence to abort.
Tracing the route to 10.1.5.5
VRF info: (vrf in name/id, vrf out name/id)
 1 10.1.12.2 44 msec 48 msec 36 msec
 2 * * *
 3 * * *
 4 * * *
...output omitted...

```

You are a bit confused, so you sit back and review what you know. You have confirmed that R1 knows about 10.1.5.5 via R2. Therefore, R1 can route packets toward that address. However, the trace that was executed is failing at R2. Is it possible that R2 does not know how to reach 10.1.1.0/26 or 10.1.1.64/26 to respond to the trace? Is it possible that 10.1.5.5 does not know about the networks either and cannot respond to the ping? You decide to focus on your thoughts about R2. R2 needs to know about the routes 10.1.1.0/26 and 10.1.1.64/26 to successfully respond to the trace. Therefore, R1 needs to be advertising the networks with the **BGP network mask** command. On R1, you issue the command **show bgp ipv4 unicast** to verify whether 10.1.1.0/26 and 10.1.1.64/26 are in the BGP table. As shown in Example 18-66, they are. Because they are in the BGP table and they are listed as valid and best, they can be advertised to the neighbors.

Example 18-66 Verifying R1's BGP Table

```

R1#show bgp ipv4 unicast
BGP table version is 10, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network          Next Hop           Metric LocPrf Weight Path
*>  1.1.1.1/32      0.0.0.0              0           32768 ?
*>  10.1.1.0/26     0.0.0.0              0           32768 i
*>  10.1.1.64/26    0.0.0.0              0           32768 i
*>  10.1.1.128/26   0.0.0.0              0           32768 i
*>  10.1.1.192/26   0.0.0.0              0           32768 i
*   10.1.5.0/24     10.1.13.3            3328                0 65502 i
*>                2.2.2.2              3328                0 65502 i
*>  10.1.12.0/24    0.0.0.0              0           32768 ?
*>  10.1.13.0/24    0.0.0.0              0           32768 ?

```

You issue the command **show bgp ipv4 unicast summary** to verify the BGP neighbors. Based on the output in Example 18-67, you confirm that both R2 and R3 are BGP neighbors because there is a number in the PfxRcd column.

Example 18-67 *Verifying R1's BGP Neighbors*

```
R1#show bgp ipv4 unicast summary
BGP router identifier 1.1.1.1, local AS number 65501
BGP table version is 10, main routing table version 10
9 network entries using 1296 bytes of memory
10 path entries using 800 bytes of memory
4/4 BGP path/bestpath attribute entries using 544 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 2664 total bytes of memory
BGP activity 19/10 prefixes, 54/44 paths, scan interval 60 secs
```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
2.2.2.2	4	65502	38	39	10	0	0	00:30:05	1
10.1.13.3	4	65502	7	6	10	0	0	00:02:06	1

Next you issue the **show bgp ipv4 unicast neighbors 2.2.2.2 advertised-routes** command and the **show bgp ipv4 unicast neighbors 10.1.13.3 advertised-routes** command to verify which routes are being advertised to R2 and R3. As verified in Example 18-68, no routes are being advertised to the neighbors.

Example 18-68 *Verifying R1's Advertised Routes*

```
R1#show bgp ipv4 unicast neighbors 2.2.2.2 advertised-routes

Total number of prefixes 0

R1#show bgp ipv4 unicast neighbors 10.1.13.3 advertised-routes

Total number of prefixes 0
```

What could prevent a route that is valid and best in the BGP table from being advertised to an eBGP neighbor? A filter? You decide to check the output of **show ip protocols** to determine whether a filter is applied to the BGP autonomous system. As shown in Example 18-69, no filter is applied.

Example 18-69 *Verifying Whether R1 Has Any BGP Filters.*

```
R1#show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "bgp 65501"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  IGP synchronization is disabled
  Automatic route summarization is disabled
  Redistributing: connected
  Unicast Aggregate Generation:
```

```

10.1.1.0/24
Neighbor(s):
  Address      FiltIn FiltOut DistIn DistOut Weight RouteMap
  2.2.2.2
  10.1.13.3
Maximum path: 1
Routing Information Sources:
  Gateway      Distance      Last Update
  2.2.2.2      20            00:37:02
  10.1.13.3    20            21:12:13
Distance: external 20 internal 200 local 200

```

But wait, you remember from your TSHOOT studies that a prefix list filter does not show up in the output of **show ip protocols**. It shows up only in the BGP neighbor output. Therefore, you issue the command **show bgp ipv4 unicast neighbors | i prefix** to see whether there is any prefix list applied at all. In the output of Example 18-70, you can see the same prefix list called **BGP_FILTER** applied twice in the outbound direction.

Example 18-70 Verifying Whether R1 Has Any BGP Prefix List Filters

```

R1#show bgp ipv4 unicast neighbors | i prefix
Outgoing update prefix filter list is BGP_FILTER
  prefix-list                27          0
Outgoing update prefix filter list is BGP_FILTER
  prefix-list                27          0

```

Now you feel like you are on the right track. Therefore, you issue the **show run | section router bgp** command, as shown in Example 18-71, to examine the BGP configuration on R1 and look for the culprit. You immediately notice that the prefix list **BGP_FILTER** is applied to neighbor 2.2.2.2 and 10.1.13.3 in the outbound direction.

Example 18-71 Verifying BGP Configuration on R1

```

R1#show run | section router bgp
router bgp 65501
  bgp log-neighbor-changes
  network 10.1.1.0 mask 255.255.255.192
  network 10.1.1.64 mask 255.255.255.192
  network 10.1.1.128 mask 255.255.255.192
  network 10.1.1.192 mask 255.255.255.192
  aggregate-address 10.1.1.0 255.255.255.0
  redistribute connected
  neighbor 2.2.2.2 remote-as 65502
  neighbor 2.2.2.2 password CISCO
  neighbor 2.2.2.2 ebgp-multihop 2
  neighbor 2.2.2.2 update-source Loopback0
  neighbor 2.2.2.2 prefix-list BGP_FILTER out
  neighbor 10.1.13.3 remote-as 65502
  neighbor 10.1.13.3 prefix-list BGP_FILTER out

```

Now you want to examine the prefix list, so you issue the command **show ip prefix-list BGP_FILTER**, as shown in Example 18-72. You immediately notice that 10.1.1.128/26 and 10.1.1.192/26 are being denied. Therefore, they are not being advertised to R2 or R3. You check your documentation, and it states that 10.1.1.128/26 and 10.1.1.192/26 should not be advertised to BGP autonomous system 65502, which this prefix list accomplishes.

Example 18-72 *Verifying a Prefix List on R1*

```
R1#show ip prefix-list BGP_FILTER
ip prefix-list BGP_FILTER: 2 entries
  seq 5 deny 10.1.1.128/26
  seq 10 deny 10.1.1.192/26
```

You think about this issue a bit more, and then it hits you. The implicit deny all at the end of the prefix list is denying all other routes. You propose that by adding the entry **ip prefix-list BGP_FILTER permit 0.0.0.0/0 le 32**, as shown in Example 18-73, to R1 will permit all other routes, which in this case are 10.1.1.0/26 and 10.1.1.64/26. The command **show ip prefix-list BGP_FILTER** confirms that it has been added.

Example 18-73 *Modifying a Prefix List on R1*

```
R1#config t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#ip prefix-list BGP_FILTER permit 0.0.0.0/0 le 32
R1(config)#end
%SYS-5-CONFIG_I: Configured from console by console
R1#show ip prefix-list BGP_FILTER
ip prefix-list BGP_FILTER: 3 entries
  seq 5 deny 10.1.1.128/26
  seq 10 deny 10.1.1.192/26
  seq 15 permit 0.0.0.0/0 le 32
```

To force a refresh of the BGP information being sent to R1's neighbors, you issue the **clear bgp ipv4 unicast * soft out** command. You then issue the commands **show bgp ipv4 unicast neighbors 2.2.2.2 advertised-routes** and **show bgp ipv4 unicast neighbors 10.1.13.3 advertised-routes** to confirm that routes are now being advertised to R1's neighbors. The output of Example 18-74 confirms that 10.1.1.0/26 and 10.1.1.64/26 are now being advertised.

Example 18-74 *Verifying Routes Advertised to R1's Neighbors*

```
R1#show bgp ipv4 unicast neighbors 2.2.2.2 advertised-routes
BGP table version is 10, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

```

      Network      Next Hop      Metric LocPrf Weight Path
*>  1.1.1.1/32      0.0.0.0          0          32768 ?
*>  10.1.1.0/26     0.0.0.0          0          32768 i
*>  10.1.1.64/26    0.0.0.0          0          32768 i
...output omitted...

R1#show bgp ipv4 unicast neighbors 10.1.13.3 advertised-routes
BGP table version is 10, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

      Network      Next Hop      Metric LocPrf Weight Path
*>  1.1.1.1/32      0.0.0.0          0          32768 ?
*>  10.1.1.0/26     0.0.0.0          0          32768 i
*>  10.1.1.64/26    0.0.0.0          0          32768 i
...output omitted...

```

However, you still want to confirm the problem is solved. Can users in 10.1.10/26 and 10.1.164/26 reach 10.1.5.5? To confirm the problem is solved, you ping 10.1.5.5 from 10.1.1.1 and 10.1.1.65 again. As shown in Example 18-75, it is solved.

Example 18-75 *Verifying That the Problem Is Solved*

```

R1#ping 10.1.5.5 source 10.1.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.5.5, timeout is 2 seconds:
Packet sent with a source address of 10.1.1.1
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 44/58/68 ms
R1#ping 10.1.5.5 source 10.1.1.65
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.5.5, timeout is 2 seconds:
Packet sent with a source address of 10.1.1.65
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/48/80 ms

```

Trouble Ticket 18-3

Problem: You are the administrator for BGP autonomous system 65502. Traffic reports indicate that all traffic out of the autonomous system is flowing through R3 and across the backup link. This is undesirable unless the link between R2 and R1 fails.

To verify the issue, you use traceroute from R5. As shown in Example 18-76, the trace to 10.1.1.1 and 10.1.1.65 goes through R3 to get to autonomous system 65501.

Example 18-76 *Verifying the Issue*

```

R5#traceroute 10.1.1.1 source 10.1.5.5
Type escape sequence to abort.
Tracing the route to 10.1.1.1
VRF info: (vrf in name/id, vrf out name/id)
  1 10.1.45.4 48 msec 40 msec 28 msec
  2 10.1.34.3 64 msec 32 msec 60 msec
  3 10.1.13.1 [AS 65501] 72 msec 52 msec 48 msec
R5#traceroute 10.1.1.65 source 10.1.5.5
Type escape sequence to abort.
Tracing the route to 10.1.1.65
VRF info: (vrf in name/id, vrf out name/id)
  1 10.1.45.4 48 msec 40 msec 28 msec
  2 10.1.34.3 64 msec 32 msec 60 msec
  3 10.1.13.1 [AS 65501] 72 msec 52 msec 48 msec

```

On R5, you issue the **show ip route 10.1.1.1** command and **show ip route 10.1.1.65** command to verify the routes. As shown in Example 18-77, the routes were learned via iBGP and are reachable via 3.3.3.3, which is R3.

Example 18-77 *Verifying the Routes on R5*

```

R5#show ip route 10.1.1.1
Routing entry for 10.1.1.0/26
  Known via "bgp 65502", distance 200, metric 0
  Tag 65501, type internal
  Last update from 3.3.3.3 00:01:09 ago
  Routing Descriptor Blocks:
  * 3.3.3.3, from 3.3.3.3, 00:01:09 ago
    Route metric is 0, traffic share count is 1
    AS Hops 1
    Route tag 65501
    MPLS label: none
R5#show ip route 10.1.1.65
Routing entry for 10.1.1.64/26
  Known via "bgp 65502", distance 200, metric 0
  Tag 65501, type internal
  Last update from 3.3.3.3 00:02:10 ago
  Routing Descriptor Blocks:
  * 3.3.3.3, from 3.3.3.3, 00:02:10 ago
    Route metric is 0, traffic share count is 1
    AS Hops 1
    Route tag 65501
    MPLS label: none

```

Are the routes being learned from R2? You issue the **show bgp ipv4 unicast** command to examine the BGP table. According to the BGP table in Example 18-78 10.1.1.0/26 and

10.1.1.64/26 are both learned via R2 as well. So, why is R5 preferring R3 as the best path? You must now examine the BGP path selection process between the next hops 2.2.2.2 and 3.3.3.3.

First of all, can R5 reach 2.2.2.2 and 3.3.3.3? Obviously, 3.3.3.3 is reachable because R5 is using it at the moment. However, using the command **show ip route 2.2.2.2**, as shown in Example 18-79, confirms that 2.2.2.2 is reachable as well. This is important because a path will never be used if the next hop is not reachable.

Next you examine weight as shown in Example 18-78. It is 0 for both the path via 2.2.2.2 and 3.3.3.3. Therefore, a tie means check the next attribute, which is local preference. In this case, the path via 2.2.2.2 is 50, and the path via 3.3.3.3 is 100. Local preference has a default value of 100, and higher is better. That is why 3.3.3.3 is preferred. It has the higher local preference. It appears the path via 2.2.2.2 had its local preference modified either when it was advertised by R2 or when it was received by R5.

Example 18-78 Examining R5's BGP Table

```
R5#show bgp ipv4 unicast
BGP table version is 613, local router ID is 5.5.5.5
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*>i 1.1.1.1/32	3.3.3.3	0	100	0	65501 ?
* i	2.2.2.2	0	50	0	65501 ?
*>i 10.1.1.0/26	3.3.3.3	0	100	0	65501 i
* i	2.2.2.2	0	50	0	65501 i
*>i 10.1.1.64/26	3.3.3.3	0	100	0	65501 i
* i	2.2.2.2	0	50	0	65501 i
r>i 10.1.5.0/24	3.3.3.3	3328	100	0	i
r i	2.2.2.2	3328	50	0	i
r>i 10.1.12.0/24	3.3.3.3	0	100	0	65501 ?
r i	2.2.2.2	0	50	0	65501 ?
r>i 10.1.13.0/24	3.3.3.3	0	100	0	65501 ?
r i	2.2.2.2	0	50	0	65501 ?

Example 18-79 Confirming That 2.2.2.2 Is Reachable

```
R5#show ip route 2.2.2.2
Routing entry for 2.2.2.2/32
  Known via "eigrp 100", distance 90, metric 131072, type internal
  Redistributing via eigrp 100
  Last update from 10.1.45.4 on GigabitEthernet1/0, 22:33:44 ago
  Routing Descriptor Blocks:
    * 10.1.45.4, from 10.1.45.4, 22:33:44 ago, via GigabitEthernet1/0
```

```

Route metric is 131072, traffic share count is 1
Total delay is 5020 microseconds, minimum bandwidth is 1000000 Kbit
Reliability 255/255, minimum MTU 1500 bytes
Loading 1/255, Hops 2

```

You examine R5's BGP configuration with the **show run | section router bgp** command. As shown in Example 18-80, there is no indication that the local preference is being modified. If there were, we would see a route map applied to the **neighbor** statement of 2.2.2.2.

Example 18-80 Examining R5's BGP Configuration

```

R5#show run | section router bgp
router bgp 65502
  bgp log-neighbor-changes
  neighbor 2.2.2.2 remote-as 65502
  neighbor 2.2.2.2 update-source Loopback0
  neighbor 3.3.3.3 remote-as 65502
  neighbor 3.3.3.3 update-source Loopback0

```

Next you move to R2 and issue the **show run | section router bgp** command. Immediately you notice a route map called **TSHOOT_BGP_FILTER** applied in the outbound direction for the peer group called **TSHOOT_IBGP_NEIGHBORS**, as shown in Example 18-81. You also notice that R5 is part of the peer group. Therefore, the route map applies to R5. You need to dig into the route map now, so you issue the command **show route-map TSHOOT_BGP_FILTER**. As shown in Example 18-82, the route map is setting the local preference to 50. You examine the network documentation, and it states that the local preference should be 150.

Example 18-81 Examining R2's BGP Configuration

```

R2#show run | section router bgp
router bgp 65502
  bgp log-neighbor-changes
  network 10.1.5.0 mask 255.255.255.0
  neighbor TSHOOT_IBGP_NEIGHBORS peer-group
  neighbor TSHOOT_IBGP_NEIGHBORS transport connection-mode passive
  neighbor TSHOOT_IBGP_NEIGHBORS update-source Loopback0
  neighbor TSHOOT_IBGP_NEIGHBORS next-hop-self
  neighbor TSHOOT_IBGP_NEIGHBORS route-map TSHOOT_BGP_FILTER out
  neighbor 1.1.1.1 remote-as 65501
  neighbor 1.1.1.1 password CISCO
  neighbor 1.1.1.1 ebgp-multihop 2
  neighbor 1.1.1.1 update-source Loopback0
  neighbor 3.3.3.3 remote-as 65502
  neighbor 3.3.3.3 peer-group TSHOOT_IBGP_NEIGHBORS
  neighbor 4.4.4.4 remote-as 65502
  neighbor 4.4.4.4 peer-group TSHOOT_IBGP_NEIGHBORS
  neighbor 5.5.5.5 remote-as 65502
  neighbor 5.5.5.5 peer-group TSHOOT_IBGP_NEIGHBORS

```

Example 18-82 *Examining R2's Route Map*

```
R2#show route-map TSHOOT_BGP_FILTER
route-map TSHOOT_BGP_FILTER, permit, sequence 10
  Match clauses:
  Set clauses:
    local-preference 50
  Policy routing matches: 0 packets, 0 bytes
```

You modify the route map on R2, as shown in Example 18-83, to solve the issue. You confirm the changes were applied by using the command **show route-map TSHOOT_BGP_FILTER**. The local preference has been successfully modified to 150. To speed up the BGP changes, you issue the **clear bgp ipv4 unicast * soft out** command.

Example 18-83 *Modifying the Local Preference Value in the Route Map*

```
R2#config t
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#route-map TSHOOT_BGP_FILTER 10
R2(config-route-map)#set local-preference 150
R2(config-route-map)#end
%SYS-5-CONFIG_I: Configured from console by console
R2#show route-map TSHOOT_BGP_FILTER
route-map TSHOOT_BGP_FILTER, permit, sequence 10
  Match clauses:
  Set clauses:
    local-preference 150
  Policy routing matches: 0 packets, 0 bytes
```

You go back to R5 and issue a trace and confirm that the path through R2 is now being used, as shown in Example 18-84.

Example 18-84 *Confirming That the Issue Is Solved*

```
R5#traceroute 10.1.1.1 source 10.1.5.5
Type escape sequence to abort.
Tracing the route to 10.1.1.1
VRF info: (vrf in name/id, vrf out name/id)
 1 10.1.45.4 28 msec 44 msec 8 msec
 2 10.1.24.2 40 msec 40 msec 40 msec
 3 10.1.12.1 [AS 65501] 64 msec 56 msec 100 msec
R5#traceroute 10.1.1.65 source 10.1.5.5
Type escape sequence to abort.
Tracing the route to 10.1.1.65
VRF info: (vrf in name/id, vrf out name/id)
 1 10.1.45.4 28 msec 44 msec 24 msec
 2 10.1.24.2 32 msec 56 msec 48 msec
 3 10.1.12.1 [AS 65501] 68 msec 36 msec 56 msec
```

MP-BGP Trouble Tickets

This section presents various trouble tickets relating to the topics discussed earlier in the chapter. The purpose of these trouble tickets is to give a process that you can follow when troubleshooting in the real world or in an exam environment. All trouble tickets in this section are based on the topology depicted in Figure 18-10.

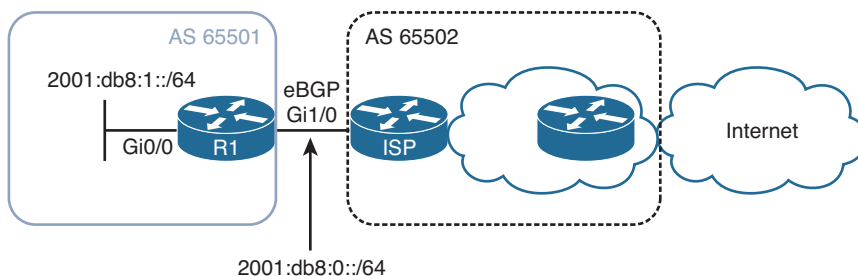


Figure 18-10 *MP-BGP Trouble Tickets Topology*

Trouble Ticket 18-4

Problem: You are an administrator of BGP autonomous system 65501. You have been asked by another administrator in your autonomous system for help. The default route from your ISP is not being learned by your router (R1) via BGP. As a result of this, no one in your autonomous system is able to reach the Internet.

You start by confirming the issue by using the **show ipv6 route** command on R1. In Example 18-85, no default route is present. The default route is supposed to be learned from the ISP router via MP-eBGP.

Example 18-85 *Verifying the Problem*

```
R1#show ipv6 route
IPv6 Routing Table - default - 5 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
        B - BGP, R - RIP, H - NHRP, I1 - ISIS L1
        I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary, D - EIGRP
        EX - EIGRP external, ND - ND Default, NDp - ND Prefix, DCE - Destination
        NDr - Redirect, O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1
        OE2 - OSPF ext 2, ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2, l - LISP
C   2001:DB8::/64 [0/0]
    via GigabitEthernet1/0, directly connected
L   2001:DB8::1/128 [0/0]
    via GigabitEthernet1/0, receive
C   2001:DB8:1::/64 [0/0]
    via GigabitEthernet0/0, directly connected
L   2001:DB8:1::1/128 [0/0]
    via GigabitEthernet0/0, receive
L   FF00::/8 [0/0]
    via Null0, receive
```

You issue the command **show bgp ipv6 unicast** to verify the contents of the IPv6 BGP table, as shown in Example 18-86. There is nothing in the IPv6 BGP table.

Example 18-86 *Viewing the IPv6 BGP Table*

```
R1#show bgp ipv6 unicast
R1#
```

Next you verify whether there are any IPv6 unicast BGP neighbors on R1. The output of **show bgp ipv6 unicast summary** indicates that there are no neighbors, as shown in Example 18-87.

Example 18-87 *Viewing the IPv6 Unicast BGP Neighbors*

```
R1#show bgp ipv6 unicast summary
R1#
```

You have a feeling that there is an error in the BGP configuration on R1. Therefore, you issue the **show run | section router bgp** command to verify R1's BGP configuration. As shown in Example 18-88, the **neighbor 2001:DB8::2 remote-as 65502** command is specified. The address is correct, and the remote autonomous system is correct. However, you notice the command **no neighbor 2001:DB8::2 activate**, which means that the neighbor is not activated in the AF. However, be careful here. This is the IPv4 AF, and we are dealing with IPv6. Therefore, we need to activate the neighbor in the IPv6 AF. Upon closer look, there is no IPv6 AF specified, and as a result, the neighbor 2001:DB8::2 is not activated.

Example 18-88 *Viewing the BGP Configuration on R1*

```
R1#show run | section router bgp
router bgp 65501
  bgp router-id 1.1.1.1
  bgp log-neighbor-changes
  neighbor 2001:DB8::2 remote-as 65502
  !
  address-family ipv4
    no neighbor 2001:DB8::2 activate
  exit-address-family
```

To solve this issue, you need to activate the neighbor with the **neighbor 2001:DB8::2 activate** command in IPv6 AF configuration mode, as shown in Example 18-89. After you activate the neighbor, the adjacency comes up.

Example 18-89 *Activating the Neighbor in Address Family Configuration Mode*

```
R1#config t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#router bgp 65501
R1(config-router)#address-family ipv6 unicast
```

```
R1(config-router-af)#neighbor 2001:db8::2 activate
R1(config-router-af)#
%BGP-5-ADJCHANGE: neighbor 2001:DB8::2 Up
```

You examine the IPv6 BGP table on R1 again with the **show bgp ipv6 unicast** command and notice that the default route is now listed in Example 18-90. The routing table, as shown in Example 18-91, also shows the default route. Problem solved!

Example 18-90 Verifying That the Default Route Is in the IPv6 BGP Table on R1

```
R1#show bgp ipv6 unicast
BGP table version is 4, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

	Network	Next Hop	Metric	LocPrf	Weight	Path
*>	::/0	2001:DB8::2	0		0	65502 i

Example 18-91 Verifying That the Default Route Is in the IPv6 Routing Table on R1

```
R1#show ipv6 route
IPv6 Routing Table - default - 6 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, R - RIP, H - NHRP, I1 - ISIS L1
       I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary, D - EIGRP
       EX - EIGRP external, ND - ND Default, NDp - ND Prefix, DCE - Destination
       NDr - Redirect, O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1
       OE2 - OSPF ext 2, ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2, l - LISP
B    ::/0 [20/0]
    via FE80::C836:17FF:FEE8:1C, GigabitEthernet1/0
C    2001:DB8::/64 [0/0]
    via GigabitEthernet1/0, directly connected
L    2001:DB8::1/128 [0/0]
    via GigabitEthernet1/0, receive
C    2001:DB8:1::/64 [0/0]
    via GigabitEthernet0/0, directly connected
L    2001:DB8:1::1/128 [0/0]
    via GigabitEthernet0/0, receive
L    FF00::/8 [0/0]
    via Null0, receive
```

Exam Preparation Tasks

As mentioned in the section “How to Use This Book” in the Introduction, you have a couple of choices for exam preparation: the exercises here; Chapter 22, “Final Preparation;” and the exam simulation questions on the CD-ROM.

Review All Key Topics

Review the most important topics in this chapter, noted with the Key Topic icon in the outer margin of the page. Table 18-2 lists a reference of these key topics and the page numbers on which each is found.



Table 18-2 *Key Topics for Chapter 18*

Key Topic Element	Description	Page Number
Example 18-1	Verifying BGP neighbors with show bgp ipv4 unicast summary	753
List	Outlines the issues you should consider when troubleshooting BGP neighbor relationships	753
Section	Path to neighbor is via default route	755
Section	Incorrect neighbor remote-as statement	757
Paragraph	Discusses how to control the source address of BGP packets	758
Paragraph	Describes how BGP TCP sessions are formed and how you can control the server and client for the TCP session	760
Paragraph	Explains how to manipulate the TTL of an eBGP packet	763
Paragraph	Describes how the minimum hold-time parameter can prevent BGP neighbor relationships	765
List	Outlines the reasons why a BGP route might be missing from the BGP table or the routing table	766
Example 18-22	Examining the BGP table	767
List	Identifies the requirements of the BGP network mask command	768
Paragraph	Discusses the BGP next-hop issue	770
Paragraph	Describes how to identify BGP split-horizon issues	773
Paragraph	Outlines how to troubleshoot filters that may be preventing BGP routes from being advertised or learned	778

Key Topic Element	Description	Page Number
List	Provides the steps that BGP uses to successfully determine the best path to reach a given network	781
Paragraph	Describes the next-hop issue that occurs when exchanging IPv6 BGP routes over IPv4 BGP TCP sessions	788
Paragraph	Describes how to solve the next-hop issue that occurs when exchanging IPv6 BGP routes over IPv4 BGP TCP sessions	788

Define Key Terms

Define the following key terms from this chapter and check your answers in the glossary:

BGP, EGP, eBGP, iBGP, MP-BGP, ISP, address family, TTL, peer group, split-horizon rule (iBGP), weight, local preference, autonomous system path, MED

Command Reference to Check Your Memory

This section includes the most important **show** and **debug** commands covered in this chapter. It might not be necessary to memorize the complete syntax of every command, but you should be able to remember the basic keywords that are needed.

To test your memory of the commands, cover the right side of Table 18-3 with a piece of paper, read the description on the left side, and then see how much of the command you can remember.

The 300-135 TSHOOT exam focuses on practical, hands-on skills that are used by a networking professional. Therefore, you should be able to identify the commands needed to successfully identify and troubleshoot the issues presented in this chapter.

Table 18-3 *show and debug Commands*

Task	Command Syntax
Displays a router's BGP RID, autonomous system number, information about the BGP's memory usage, and summary information about IPv4/IPv6 unicast BGP neighbors.	show bgp {ipv4 ipv6} unicast summary
Displays detailed information about all the IPv4/IPv6 BGP neighbors of a router.	show bgp {ipv4 ipv6} unicast neighbors
Displays the IPv4/IPv6 network prefixes present in the IPv4/IPv6 BGP table.	show bgp {ipv4 ipv6} unicast

Task	Command Syntax
Shows routes known to a router's IPv4/IPv6 routing table that were learned via BGP.	<code>show {ipv4 ipv6}route bgp</code>
Provides real-time information about BGP events, such as the establishment of a peering relationship.	<code>debug ip bgp</code>
Shows real-time information about BGP updates sent and received by a BGP router.	<code>debug ip bgp updates</code>
Displays updates that occur in a router's IP routing table. Therefore, this command is not specific to BGP.	<code>debug ip routing</code>

Note The command `show ip bgp` will display the same output as `show bgp ipv4 unicast`. The command `show ip bgp summary` will display the same output as `show bgp ipv4 unicast summary`. The command `show ip bgp neighbors` will display the same output as `show bgp ipv4 unicast neighbors`.

This page intentionally left blank



This chapter covers the following topics:

- **Management Protocols Troubleshooting:** This section examines how to recognize and troubleshoot issues related to management protocols such as NTP, syslog, and SNMP.
- **Management Tools Troubleshooting:** This section examines how to recognize and troubleshoot issues related to management tools such as Cisco IP SLA, object tracking, SPAN, and RSPAN.
- **Management Protocols and Tools Trouble Tickets:** This section provides trouble tickets that demonstrate how you can use a structured troubleshooting process to solve a reported problem.

Troubleshooting Management Protocols and Tools

During your troubleshooting endeavors, you will rely on various protocols and tools to help you solve the problems that are being presented. Some tools will be used to notify you of issues, some will be used to gather additional information, and some will even be used to help you monitor and maintain the health of the network.

This chapter covers how to identify and troubleshoot issues related to management protocols such as Network Time Protocol (NTP), which is used to keep accurate time in the network; syslog, which will notify you of changes on a device; and Simple Network Management Protocol (SNMP), which is used to monitor the health of a device.

In addition, this chapter explains how to identify and troubleshoot issues related to management tools such as Cisco IP SLA (service level agreement), which can measure the health of your network; object tracking, which can keep track of the status of an object; and Switched Port Analyzer / Remote Switched Port Analyzer (SPAN/RSPAN), which enables you to copy frames from one switchport on a switch to a port on the same switch or a different switch.

Usually, you will be spending your time troubleshooting issues using these management protocols and tools. However, you might sometimes troubleshoot issues related to the protocols and tools that help you. When that time comes, you need to be ready.

“Do I Know This Already?” Quiz

The “Do I Know This Already?” quiz allows you to assess whether you should read this entire chapter thoroughly or jump to the “Exam Preparation Tasks” section. If you are in doubt about your answers to these questions or your own assessment of your knowledge of the topics, read the entire chapter. Table 19-1 lists the major headings in this chapter and their corresponding “Do I Know This Already?” quiz questions. You can find the answers in Appendix A, “Answers to the ‘Do I Know This Already?’ Quizzes.”

Table 19-1 “Do I Know This Already?” Section-to-Question Mapping

Foundation Topics Section	Questions
Management Protocols Troubleshooting	1–6
Management Tools Troubleshooting	7–10

Caution The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you do not know the answer to a question or are only partially sure of the answer, you should mark that question as wrong for purposes of the self-assessment. Giving yourself credit for an answer that you correctly guess skews your self-assessment results and might provide you with a false sense of security.

1. Which port does NTP use?
 - a. 22
 - b. 23
 - c. 123
 - d. 514
2. Which stratum level indicates that an NTP server is not reachable?
 - a. 1
 - b. 5
 - c. 10
 - d. 16
3. Which port is used by syslog?
 - a. 22
 - b. 23
 - c. 110
 - d. 514
4. You have accessed a router via telnet and issued the **debug ntp packets** command. No debugs are being displayed in the terminal window even though the logging level to the vty lines is set to debugging. Why?
 - a. Debugs are not sent to the vty lines.
 - b. You need to issue the **terminal no monitor** command.
 - c. You need to issue the **terminal monitor** command.
 - d. Debugs need to be enabled.
5. Which command enables you to verify which SNMP group a user belongs to?
 - a. **show snmp user**
 - b. **show snmp group**
 - c. **show snmp host**
 - d. **show snmp view**

6. Which two commands are used to verify the OIDs that a particular group is able to access on the local device?
 - a. `show snmp user`
 - b. `show snmp group`
 - c. `show snmp host`
 - d. `show snmp view`
7. Which command enables you to verify the number of successes and failures for an IP SLA instance?
 - a. `show ip sla configuration`
 - b. `show ip sla statistics`
 - c. `show ip sla responder`
 - d. `show ip sla summary`
8. Which of the following statements are true? (Choose two answers.)
 - a. A SPAN session copies packets from a switchport on one device to a switchport on the same device.
 - b. A SPAN session copies packets from a switchport on one device to a switchport on a different device.
 - c. An RSPAN session copies packets from a switchport on one device to a switchport on the same device.
 - d. An RSPAN session copies packets from a switchport on one device to a switchport on a different device.
9. Which two commands can be used to verify that a switchport is a destination SPAN or RSPAN monitoring port?
 - a. `show ip interface brief`
 - b. `show ip interfaces`
 - c. `show interfaces status`
 - d. `show monitor`
10. Which three commands enable you to verify that a VLAN is an RSPAN VLAN?
 - a. `show vlan`
 - b. `show vlan brief`
 - c. `show monitor`
 - d. `show vlan remote-span`

Foundation Topics

Management Protocols Troubleshooting

Tools such as syslog and SNMP help you monitor the health of your network devices. They are very valuable tools if they are working properly. If they have been misconfigured, you will not be able to gather the information you need while troubleshooting specific events, or be notified that an event has occurred. In addition, it is important that you know what time the events occurred. Therefore, you need accurate time using a protocol such as NTP. However, if there is an issue with NTP, you need to be able to solve it quickly so that log messages have the appropriate time.

This section explains how to identify and troubleshoot issues related to NTP, syslog, and SNMP.

NTP Troubleshooting

Network Time Protocol is used to synchronize clocks among the various network devices. It is a client/server protocol where NTP servers provide time to NTP clients. There are many reasons as to why a device configured as an NTP client might not be able to synchronize with an NTP server. The following list details many of these reasons:



- **The time server is not reachable:** To synchronize with the NTP server, you have to be able to reach it. Use the **ping** command to verify connectivity from the client to the server. However, be careful to test using the correct destination IP addresses and source IP addresses. For example, if your client is configured to source NTP packets from Loopback 0, you should **ping** with a source of Loopback 0.
- **ACL blocking NTP packets:** NTP uses UDP port 123. Therefore, it is important that no access control list (ACL) exists between the NTP client and server that is configured to deny NTP packets either on purpose or by accident. You will need to verify whether any ACLs exist on interfaces with the **show ip interface *interface_type* *interface_number*** command and, if you find one, verify the ACL entries with the **show access-list** command.
- **NTP authentication mismatch:** Authentication is not required, but if implemented both the server and the client need to be configured with the correct authentication key and key string. To verify the NTP authentication configuration, use the **show run | section ntp** command.
- **The wrong server is being used:** You can configure a client with multiple NTP servers; by default, the protocol will choose the best server. However, this may not be the one you want it to use. Therefore, you can force a preferred NTP server with the **ntp server *ip_address* prefer** command. To verify which server is being used, use the **show ntp status** command.

- **High CPU utilization:** The CPU is responsible for processing NTP packets. If the CPU is under high load it will fail to process packets and synchronization will fail. You can verify CPU load with the **show processes cpu** command.
- **Time offset is too high:** If the offset between the clock on the server and the client is extreme, it can take a significant amount of time for the clock to synchronize, or it may not synchronize at all. Therefore, you should manually set the clock with the **clock set [hh:mm:ss] [day] [month] [year]** command and then allow NTP to fine-tune the clocks. To verify the clock that is set on a device, issue the **show clock** command.
- **Stratum level is too high:** The NTP hierarchy is based on stratum levels from 1 to 15. 1 is considered the best (most reliable), and 15 is considered the worst (least reliable). A stratum level of 16 is unreachable. Therefore, if a device is synchronizing with another device that has a stratum of 15, the synchronization will fail.
- **Server is configured to accept NTP packets from specific IP addresses:** You can configure an NTP access group on the NTP server to control which NTP packets will be responded to. If the NTP clients are sourcing NTP packets from the wrong IP address, the server will not respond to the packets as the source address of the packet does not match the ACL.

Example 19-1 displays the output of **show ntp status**. With this command, you can verify whether the clock is synchronized, the stratum level, and the IP address of the time server the local device is synchronized with. You can also verify clock statistics if necessary.

Key Topic

Example 19-1 Verifying the Status of NTP on a Client

```
SW1#show ntp status
Clock is synchronized, stratum 2, reference is 192.168.1.3
nominal freq is 119.2092 Hz, actual freq is 119.2116 Hz, precision is 2**17
reference time is D77BFCDB.2A77CE72 (21:44:59.165 UTC Thu Jul 24 2014)
clock offset is -85.7435 msec, root delay is 43.18 msec
root dispersion is 105.32 msec, peer dispersion is 3.73 msec
loopfilter state is 'CTRL' (Normal Controlled Loop), drift is -0.000020157 s/s
system poll interval is 64, last update was 215 sec ago.
```

Example 19-2 displays the output of **show ntp associations**, which you can use to check the status of the configured NTP servers. You can also verify which server is currently being used for time synchronization and which servers are candidate time servers. Therefore, if there are multiple time servers configured on the device, all of them will be listed here. The * beside 192.168.1.3 indicates that the local device is synchronized with that server. A + beside it means that it is a candidate server for synchronization.

Example 19-2 *Verifying NTP Time Server Associations on the Client*

```
SW1#show ntp associations

address          ref clock      st  when  poll reach  delay  offset  disp
*~192.168.1.3    .LOCL.         1   55    64   377 44.591 -97.671 4.366
* sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured
```

To obtain detailed output of the NTP server associations (including if the server is authenticated), you can issue the **show ntp associations detail** command, as shown in Example 19-3.

**Example 19-3** *Verifying Details of the NTP Time Servers Associated with the Client*

```
SW1#show ntp associations detail
192.168.1.3 configured, authenticated, our_master, sane, valid, stratum 1
ref ID .LOCL., time D77C0219.89D511B9 (22:07:21.538 UTC Thu Jul 24 2014)
our mode client, peer mode server, our poll intvl 64, peer poll intvl 64
root delay 0.00 msec, root disp 0.27, reach 377, sync dist 30.35
delay 30.37 msec, offset -101.7186 msec, dispersion 3.11
precision 2**18, version 4
org time D77C021C.E4F5FC74 (22:07:24.894 UTC Thu Jul 24 2014)
rec time D77C021D.0A706F23 (22:07:25.040 UTC Thu Jul 24 2014)
xmt time D77C021C.FC13B398 (22:07:24.984 UTC Thu Jul 24 2014)
filtdelay =    56.03   34.00   37.55   39.02   30.37   36.97   48.39   63.59
filtoffset = -118.38 -102.96  -98.68  -99.64 -101.71 -125.02 -105.19 -102.24
filtererror =    0.01    0.95    1.94    2.89    3.88    4.84    5.80    6.74
minpoll = 6, maxpoll = 10
```

If your client is not synchronizing with the server, you can use the **debug ntp all** command, as shown in Example 19-4, which will debug NTP, events, core messages, clock adjustments, reference clocks, and packets. As seen on SW1, the **debug** output shows that an NTP message is sent to the NTP server at 192.168.1.3. If this message does not get a response, synchronization cannot occur. In this case, an NTP message has been received from the server at 192.168.1.3 and is being processed.

Example 19-4 *Using Debugs to Troubleshoot NTP Issues*

```
SW1#debug ntp all
NTP events debugging is on
NTP core messages debugging is on
NTP clock adjustments debugging is on
NTP reference clocks debugging is on
NTP packets debugging is on
SW1#
NTP message sent to 192.168.1.3, from interface 'Loopback0' (192.168.1.10).
NTP message received from 192.168.1.3 on interface 'Loopback0' (192.168.1.10).
NTP Core(DEBUG): ntp_receive: message received
```

```
NTP Core(DEBUG): ntp_receive: peer is 0x041657A8, next action is 1.
NTP Core(DEBUG): receive: packet given to process_packet
```

In Example 19-5 the **debug ntp all** command is displaying that an NTP message is sent to the server at 192.168.1.3 and that the server has sent a response back. However, the NTP message is being dropped because of a crypto-NAK. This means that the authentication parameters do not match between the client and the server. You will need to compare the configurations between the server and the client and make sure the authentication commands match.

Example 19-5 Using debug to Verify NTP Authentication Issues

```
SW1#debug ntp all
NTP message sent to 192.168.1.3, from interface 'Loopback0' (192.168.1.10).
NTP message received from 192.168.1.3 on interface 'Loopback0' (192.168.1.10).
NTP Core(DEBUG): ntp_receive: message received
NTP Core(DEBUG): ntp_receive: peer is 0x041657A8, next action is 1.
NTP Core(NOTICE): ntp_receive: dropping message: crypto-NAK.
```

Syslog Troubleshooting

To verify your syslog configuration, confirm logging is enabled, and view the syslog messages stored in the buffer, you use the command **show logging**, as shown in Example 19-6. When troubleshooting, you need syslog to generate the right type of messages at the right time. By default, console, monitor, and buffer logging display messages with a severity level of debugging (7) and lower. Logging to a server is disabled by default, but once enabled, all severity levels will be sent to the server. Therefore, in all cases if you are not receiving the syslog messages you expect, verify that the correct level is configured. In this example, console and monitor are configured with a level of informational, buffer is configured with a level of debugging, and the trap logging (server) is configured with a level of warnings.



When logging to a server the correct server IP address needs to be specified and the server needs to be reachable. In addition, because syslog uses UDP port 514, it is important to make sure that no ACLs are blocking traffic destined to UDP port 514.

The buffer will have a default size of 8192 bytes. Once the buffer fills up, the older entries are overwritten. Therefore, if you are using the buffer and experiencing a loss of syslog messages, consider increasing the size of the buffer with the **logging buffered size** command or sending the messages to a syslog server instead.

Finally, if you have remotely connected to a device via Telnet or SSH, and no syslog messages are appearing, it is because the **terminal monitor** command has not been issued.

Example 19-6 *Verifying Syslog Configuration*

```

R4#show logging
Syslog logging: enabled (0 messages dropped, 0 messages rate-limited, 0 flushes, 0
overruns, xml disabled, filtering disabled)

No Active Message Discriminator.

Inactive Message Discriminator:
OSPF          severity group drops    4

  Console logging: level informational, 116 messages logged, xml disabled,
                    filtering disabled
  Monitor logging: level informational, 0 messages logged, xml disabled,
                    filtering disabled
  Buffer logging:  level debugging, 175 messages logged, xml disabled,
                    filtering disabled
  Exception Logging: size (8192 bytes)
  Count and timestamp logging messages: disabled
  Persistent logging: disabled

No active filter modules.

  Trap logging: level warnings, 108 message lines logged
    Logging to 10.1.100.100 (udp port 514, audit disabled,
        link up),
        2 message lines logged,
        0 message lines rate-limited,
        0 message lines dropped-by-MD,
        xml disabled, sequence number disabled
        filtering disabled
    Logging Source-Interface:          VRF Name:

  Log Buffer (8192 bytes):
Jul 24 21:54:50.422: %SYS-5-CONFIG_I: Configured from console by console
Jul 24 21:57:16.070: %OSPFv3-4-ERRRCV: OSPFv3-10-IPv6 Received invalid packet: Bad
Checksum from FE80::C829:FFF:FE50:54, GigabitEthernet2/0
Jul 24 21:58:20.014: NTP message received from 192.168.1.10 on interface
'GigabitEthernet2/0' (10.1.34.4).
Jul 24 21:58:20.018: NTP Core(DEBUG): ntp_receive: message received
Jul 24 21:58:20.022: NTP Core(DEBUG): ntp_receive: peer is 0x00000000, next action
is 3.
Jul 24 21:58:20.030: NTP message sent to 192.168.1.10, from interface
'GigabitEthernet2/0' (10.1.34.4).
Jul 24 21:59:25.014: NTP message received from 192.168.1.10 on interface
'GigabitEthernet2/0' (10.1.34.4).
Jul 24 21:59:25.018: NTP Core(DEBUG): ntp_receive: message received
Jul 24 21:59:25.022: NTP Core(DEBUG): ntp_receive: peer is 0x00000000, next action

```

```
is 3.
```

```
Jul 24 21:59:25.026: NTP message sent to 192.168.1.10, from interface 'GigabitEther-  
net2/0' (10.1.34.4).
```

Having log messages and debug messages stamped with a time is critical for troubleshooting. If no time stamps are included with either, it is because the **no service time-stamps** command has been executed. To configure time stamps, use the **service time-stamps [debug | log] [datetime | uptime]** command. The **datetime** option will include the date and time the log or debug message occurred. Therefore, it is important to have an accurate calendar and time set. Use NTP for this. The **uptime** option provides a time stamp based on the amount of time that has passed since the last reboot.

SNMP Troubleshooting

Regardless of whether you are using SNMPv2c or SNMPv3, you need to be able to ping the server from the agent. If Layer 3 connectivity does not exist, the SNMP Network Management Server cannot access the information in the Management Information Base (MIB) on the agent. In addition, SNMP uses UDP port 161 for general messages and UDP port 162 for traps and informs. Therefore, if an ACL is denying these ports, SNMP communication will not occur between the NMS and the agent.

Keep the following few things in mind as you troubleshoot SNMPv2c. Refer to Example 19-7 when reviewing the following list:



- **Community strings must match:** For the NMS to read from or write to the agent, the read community string or the read/write community string must match between the NMS and the agent. In Example 19-7, the read-only community string specified is CISCO.
- **ACLs classifying servers must be correct:** If you are using ACLs to define which NMS (based on IP address) is allowed to retrieve objects from the MIB, the ACL has to accurately define the server addresses. In Example 19-7, ACL 10 is only permitting the NMS server with the IP address 10.1.100.100 to read from the MIB using the read-only community string CISCO.
- **Correct configuration for notifications:** If your agent is configured to send traps or informs, you should verify the following:
 1. That traps are enabled.
 2. The correct host (NMS) IP address is specified.
 3. The correct SNMP version is specified.
 4. The correct community string is specified.
 5. You specified **traps** or **informs** (default is **traps**).
 6. If you did not want all traps to be sent, it is imperative you specified the correct ones you want to send. In Example 19-7, the **snmp-server host** command indicates that SNMPv2c informs will be sent to the NMS at 10.1.100.100 with a community string of CISCO.

- **Indexes keep shuffling:** To prevent index shuffling and guarantee index persistence during reboots or minor software upgrades, use the **snmp-server ifindex persist** command, which shows up as **snmp ifmib ifindex persist** in the running configuration.

Example 19-7 SNMPv2c Configuration Sample

```
R4#show run | section snmp
snmp-server community CISCO RO 10
snmp-server enable traps cpu threshold
snmp-server host 10.1.100.100 informs version 2c CISCO
snmp ifmib ifindex persist
R4#show ip access-lists
Standard IP access list 10
  10 permit 10.1.100.100
```

SNMPv3 offers major improvements over SNMPv2c when it comes to security. It offers improved authentication and encryption. Keep the following few things in mind as you troubleshoot SNMPv3. Refer to Example 19-8 when reviewing the following list:



- **Nesting of users, views, and groups:** With SNMPv3, you create users with authentication and encryption parameters that are nested into groups that define the servers that are allowed to read from or write to the objects within the MIB on the agent. If you fail to nest the users, views, and groups, SNMPv3 will not function as expected. In Example 19-8, the user NMSERVER is nested into the group NMSREADONLY, which allows read-only access to the object identifiers (OIDs) listed in the view MIBACCESS to the NMS with the IP address 10.1.100.100.
- **Wrong security level specified:** SNMPv3 supports three security levels: noAuthNoPriv, authNoPriv, and authPriv. The security level specified for the group, the users, and for the sending of traps has to match what is used on the server. In Example 19-8, authPriv is being used extensively (with the **priv** parameter in the commands), which means that authentication and encryption will be used.
- **Wrong hashing algorithm, encryption algorithm, or passwords defined:** When authenticating, the hashing algorithm has to match along with the password; otherwise, authentication will fail. When performing encryption, the encryption algorithm and password have to match; otherwise, the NMS will not be able to decrypt the data it receives. In Example 19-8, SHA is being used as the hashing algorithm, AES256 as the encryption algorithm, and MYPASSWORD is the password.
- **Wrong OIDs specified in the view:** The views identify the objects within the MIB that the NMS will be able to access. If the wrong objects are defined, SNMPv3 will not produce the desired results. In Example 19-8, the objects sysUpTime, ifAdminStatus, and ifOperStatus are defined in the MIBACCESS view.
- **Correct configuration for notifications:** If your agent is configured to send traps or informs you should verify that traps are enabled, the correct host (NMS) IP address is specified, the correct SNMP version is specified, the correct security level is

specified, and you specified **traps** or **informs** (default is **traps**). If you do not want all traps to be sent, it is imperative that you specify the correct ones. You also need to specify the correct SNMPv3 username for the authentication/encryption process. In Example 19-8, the **snmp-server host** command indicates that SNMPv3 will send traps related to the CPU to the NMS at 10.1.100.100, with the authentication and encryption provided by the username NMSERVER.

- **Indexes keep shuffling:** To prevent index shuffling and guarantee index persistence during reboots or minor software upgrades, use the **snmp-server ifindex persist** command, which shows up as **snmp ifmib ifindex persist** in the running configuration.

Example 19-8 SNMPv3 Configuration Sample

```
SW2#show run | section snmp
snmp-server group NMSREADONLY v3 priv read MIBACCESS access 99
snmp-server view MIBACCESS sysUpTime included
snmp-server view MIBACCESS ifAdminStatus included
snmp-server view MIBACCESS ifOperStatus included
snmp-server user NMSERVER NMSREADONLY v3 auth sha MYPASSWORD priv aes 256 MYPASSWORD
snmp-server host 10.1.100.100 version 3 priv NMSERVER cpu
snmp ifmib ifindex persist
SW2#show ip access-lists
Standard IP access list 99
 10 permit 10.1.100.100
```

You can verify the configured snmp groups with the **show snmp group** command. In Example 19-9 the group is NMSREADONLY, the security model is v3 priv (authPriv), the associated read-only view is MIBACCESS, and only servers in access list 99 will be permitted to read the OIDs in the view..

Key Topic

Example 19-9 Verifying SNMP Groups

```
SW2#show snmp group
groupname: NMSREADONLY          security model:v3 priv
contextname: <no context specified> storage-type: nonvolatile
readview : MIBACCESS            writeview: <no writeview specified>
notifyview: *tv.00000000.00000000.10000000.0
row status: active              access-list: 99
```

You can verify the configured SNMP users with the **show snmp user** command. Example 19-10 shows a user named NMSERVER that is using the SHA authentication protocol and the AES256 privacy (encryption) protocol. The user is also associated with the group NMSREADONLY.

**Key
Topic****Example 19-10** *Verifying SNMP Users*

```
SW2#show snmp user

User name: NMSERVER
Engine ID: 800000090300001C57FEF601
storage-type: nonvolatile          active
Authentication Protocol: SHA
Privacy Protocol: AES256
Group-name: NMSREADONLY
```

To verify where traps or informs (notifications) are being sent, use the **show snmp host** command. In Example 19-11, the notifications are being sent to the NMS at 10.1.100.100 using UDP port 162. The specific notifications are traps, and the username that will be used for authentication and encryption is NMSERVER using the security model v3 priv.

**Key
Topic****Example 19-11** *Verifying SNMP Hosts*

```
SW2#show snmp host

Notification host: 10.1.100.100  udp-port: 162    type: trap
user: NMSERVER    security model: v3 priv
```

You can use the **show snmp view** command to view the OIDs that are included in each of the views. In Example 19-12, the MIBACCESS view has the OIDs sysUpTime, ifAdminStatus, and ifOperStatus included.

**Key
Topic****Example 19-12** *Verifying SNMP Views*

```
SW2#show snmp view

...output omitted...
cac_view lifEntry.20 - included read-only active
cac_view cciDescriptionEntry.1 - included read-only active
MIBACCESS sysUpTime - included nonvolatile active
MIBACCESS ifAdminStatus - included nonvolatile active
MIBACCESS ifOperStatus - included nonvolatile active
vldefault iso - included permanent active
vldefault internet - included permanent active
...output omitted...
```

Management Tools Troubleshooting

The performance of your network is critical. Being able to accurately measure the performance and have statistics that can be used to identify potential issues is the key to having a healthy network. One of the options that Cisco IOS IP SLA offers is the ability to monitor network performance. You can also use it to test reachability, and when it is attached to a tracking object, it can help maintain network availability. Being able to troubleshoot issues related to IP SLA and object tracking is essential.

Another set of tools at your disposal is SPAN and RSPAN. These tools enable you to capture frames as they traverse a switch and send copies to packet-capturing devices for analysis. These tools are valuable, and being able to troubleshoot issues related to them is important because you will likely be using them to troubleshoot other issues.

This section explains how to troubleshoot issues related to IP SLA, object tracking, SPAN, and RSPAN.

Cisco IOS IPSLA Troubleshooting

Cisco IOS IP SLA enables you to measure network performance and test network availability by generating a continuous, reliable probe (simulated traffic) in a predictable manner. The data you can collect varies greatly depending on how you set up the probe. You can collect information about packet loss, one-way latency, response times, jitter, network resource availability, application performance, server response times, and even voice quality.

IP SLA consists of an IP SLA source (sends the probes) and IP SLA responder (replies to the probes). However, both are not needed in all cases. Only the IP SLA source is required all the time. The IP SLA responder is needed only when gathering highly accurate statistics for services that are not offered by any specific destination device. The responder has the ability to respond back to the source with accurate measurements taking into account its own processing time of the probe. Figure 19-1 shows a scenario with just the IP SLA source sending a ping to test connectivity. Figure 19-2 shows a scenario with an IP SLA source and IP SLA responder that is measuring jitter (interpacket delay variance).

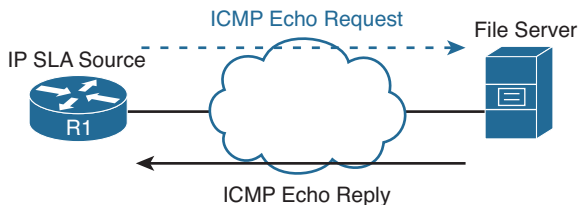


Figure 19-1 IP SLA Source Topology

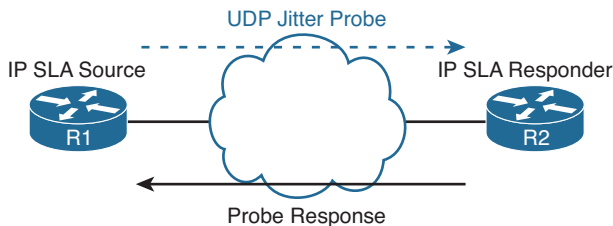


Figure 19-2 IP SLA Source and Responder Topology

Example 19-13 shows a sample configuration based on Figure 19-1. In this example, R1 is configured as an IP SLA source. The probe it is sending is an Internet Control

Message Protocol (ICMP) echo (ping) to 10.1.100.100 using the local source address of 192.168.1.11. This probe is being sent every 15 seconds and it will never expire.

Example 19-13 *IP SLA ICMP-ECHO Probe Configuration Sample*

```
R1#show run | section sla
ip sla 2
  icmp-echo 10.1.100.100 source-ip 192.168.1.11
  frequency 15
ip sla schedule 2 life forever start-time now
```

Example 19-14 shows a sample configuration based on Figure 19-2. In this example, R1 is configured as an IP SLA source. The probe it is sending is testing UDP jitter from the source address 192.168.1.11 to 10.1.34.4 using port 65051. It will send 20 probe packets for each test with a size of 160 bytes each and repeat this every 30 seconds. The probe is started and will never expire. To get measurements related to jitter, you need to have a device that can process the probes and respond accordingly. Therefore, the destination device needs to be able to support Cisco IOS IP SLA and be configured as a responder. R2 is configured as the IP SLA responder.

Example 19-14 *IP SLA UDP-JITTER Probe Configuration Sample*

```
R1#show run | section sla
ip sla 1
  udp-jitter 10.1.34.4 65051 source-ip 192.168.1.11 num-packets 20
  request-data-size 160
  frequency 30
ip sla schedule 1 life forever start-time now

R2#show run | section sla
ip sla responder
```

When troubleshooting Cisco IOS IP SLA, consider the following:



- The correct operation needs to be chosen based on the metrics you intend to measure.
- The destination IP address needs to be reachable and correctly defined.
- The source IP address needs to be reachable from the destination and correctly defined.
- Any necessary port numbers need to be correctly identified.
- The SLA instance needs to be started for it to work.
- If the operation needs an IP SLA responder, one has to be configured and reachable.

To verify which operations are supported on the platform in addition to how many operations are configured and how many are currently active, use the **show ip sla application** command, as shown in Example 19-15.

Example 19-15 *Output of show ip sla application*

```

R1#show ip sla application
      IP Service Level Agreements
Version: Round Trip Time MIB 2.2.0, Infrastructure Engine-III

Supported Operation Types:
      icmpEcho, path-echo, path-jitter, udpEcho, tcpConnect, http
      dns, udpJitter, dhcp, ftp, lsp Group, lspPing, lspTrace
      802.lagEcho VLAN, EVC, Port, 802.lagJitter VLAN, EVC, Port
      pseudowirePing, udpApp, wspApp

Supported Features:
      IPSLAs Event Publisher

IP SLAs low memory water mark: 30919230
Estimated system max number of entries: 22645

Estimated number of configurable operations: 22643
Number of Entries configured      : 2
Number of active Entries          : 2
Number of pending Entries         : 0
Number of inactive Entries        : 0
Time of last change in whole IP SLAs: 09:29:04.789 UTC Sat Jul 26 2014

```

To verify the configuration values for each IP SLA instance as well as the default values that you did not modify, use the **show ip sla configuration** command, as shown in Example 19-16. In this example, there are two entries (instances): number 1 and number 2. You can verify for each entry the type of operation that is being performed, the operation timeout, the source and destination address, the source and destination port, type of service values, packet size, packet interval (if operation supports it), and the schedule that has been configured for the operation. In this case, both entry 1 and 2 are started, and they will never expire.

Example 19-16 *Output of show ip sla configuration*

```

R1#show ip sla configuration
IP SLAs Infrastructure Engine-III
Entry number: 1
Owner:
Tag:
Operation timeout (milliseconds): 5000
Type of operation to perform: udp-jitter
Target address/Source address: 10.1.34.4/192.168.1.11
Target port/Source port: 65051/0
Type Of Service parameter: 0x0
Request size (ARR data portion): 160

```

```

Packet Interval (milliseconds)/Number of packets: 20/20
Verify data: No
Vrf Name:
Control Packets: enabled
Schedule:
    Operation frequency (seconds): 30 (not considered if randomly scheduled)
    Next Scheduled Start Time: Start Time already passed
    Group Scheduled : FALSE
    Randomly Scheduled : FALSE
    Life (seconds): Forever
    Entry Ageout (seconds): never
    Recurring (Starting Everyday): FALSE
    Status of entry (SNMP RowStatus): Active
Threshold (milliseconds): 5000
Distribution Statistics:
    Number of statistic hours kept: 2
    Number of statistic distribution buckets kept: 1
    Statistic distribution interval (milliseconds): 20
Enhanced History:

Entry number: 2
Owner:
Tag:
Operation timeout (milliseconds): 5000
Type of operation to perform: icmp-echo
Target address/Source address: 10.1.100.100/192.168.1.11
Type Of Service parameter: 0x0
Request size (ARR data portion): 28
Verify data: No
Vrf Name:
Schedule:
    Operation frequency (seconds): 15 (not considered if randomly scheduled)
    Next Scheduled Start Time: Start Time already passed
    Group Scheduled : FALSE
    Randomly Scheduled : FALSE
    Life (seconds): Forever
    Entry Ageout (seconds): never
    Recurring (Starting Everyday): FALSE
    Status of entry (SNMP RowStatus): Active
Threshold (milliseconds): 5000
Distribution Statistics:
    Number of statistic hours kept: 2
    Number of statistic distribution buckets kept: 1
    Statistic distribution interval (milliseconds): 20
Enhanced History:
History Statistics:

```

```

Number of history Lives kept: 0
Number of history Buckets kept: 15
History Filter Type: None

```

To display the results of the IP SLA operations and the statistics collected, use the **show ip sla statistics** command, as shown in Example 19-17. In the output, you can verify the type of operation, when it last started, the latest return code, the values returned (depending on the operation), and the number of successes and failures.



Example 19-17 *Output of show ip sla statistics*

```

R1#show ip sla statistics
IPSLAs Latest Operation Statistics

IPSLA operation id: 1
Type of operation: udp-jitter
    Latest RTT: 53 milliseconds
Latest operation start time: 09:52:23 UTC Sat Jul 26 2014
Latest operation return code: OK
RTT Values:
    Number Of RTT: 17                RTT Min/Avg/Max: 46/53/66 milliseconds
Latency one-way time:
    Number of Latency one-way Samples: 0
    Source to Destination Latency one way Min/Avg/Max: 0/0/0 milliseconds
    Destination to Source Latency one way Min/Avg/Max: 0/0/0 milliseconds
Jitter Time:
    Number of SD Jitter Samples: 14
    Number of DS Jitter Samples: 14
    Source to Destination Jitter Min/Avg/Max: 1/7/13 milliseconds
    Destination to Source Jitter Min/Avg/Max: 1/6/13 milliseconds
Packet Loss Values:
    Loss Source to Destination: 0
    Source to Destination Loss Periods Number: 0
    Source to Destination Loss Period Length Min/Max: 0/0
    Source to Destination Inter Loss Period Length Min/Max: 0/0
    Loss Destination to Source: 3
    Destination to Source Loss Periods Number: 2
    Destination to Source Loss Period Length Min/Max: 1/2
    Destination to Source Inter Loss Period Length Min/Max: 1/9
    Out Of Sequence: 0      Tail Drop: 0
    Packet Late Arrival: 0  Packet Skipped: 0
Voice Score Values:
    Calculated Planning Impairment Factor (ICPIF): 0
    Mean Opinion Score (MOS): 0
Number of successes: 61
Number of failures: 0

```

```

Operation time to live: Forever

IPSLA operation id: 2
    Latest RTT: 1 milliseconds
Latest operation start time: 09:52:49 UTC Sat Jul 26 2014
Latest operation return code: OK
Number of successes: 95
Number of failures: 1
Operation time to live: Forever

```

To verify the operation of the IP SLA responder, use the command **show ip sla responder**, as shown in Example 19-18, on the Cisco IOS device acting as the responder. You can verify the general control port number, the total number of probes received, the number of errors, and the recent sources of IP SLA probes.



Example 19-18 *Output of show ip sla responder*

```

R2#show ip sla responder
    General IP SLA Responder on Control port 1967
General IP SLA Responder is: Enabled
Number of control message received: 2333 Number of errors: 0
Recent sources:
    192.168.1.11 [09:53:52.001 UTC Sat Jul 26 2014]
    192.168.1.11 [09:53:22.033 UTC Sat Jul 26 2014]
    192.168.1.11 [09:52:52.029 UTC Sat Jul 26 2014]
    192.168.1.11 [09:52:22.049 UTC Sat Jul 26 2014]
    192.168.1.11 [09:51:52.029 UTC Sat Jul 26 2014]
Recent error sources:

    Permanent Port IP SLA Responder
Permanent Port IP SLA Responder is: Disabled

udpEcho Responder:
    IP Address          Port

```

Example 19-19 shows real-time output of an SLA operation with the **debug ip sla trace 2** command. The debug is displaying a successful trace of the IP SLA instance 2. The operation is waking up, starting, sending the probe, receiving a response, and then the statistics are updated accordingly.

Example 19-19 *Debug Displaying a Successful IP SLA Operation*

```

R1#debug ip sla trace 2
IPSLA-INFRA_TRACE:OPER:2 slaSchedulerEventWakeup

IPSLA-INFRA_TRACE:OPER:2 Starting an operation

IPSLA-OPER_TRACE:OPER:2 source IP:192.168.1.11

IPSLA-OPER_TRACE:OPER:2 Starting icmpEcho operation - destAddr=10.1.100.100,
sAddr=192.168.1.11

IPSLA-OPER_TRACE:OPER:2 Sending ID: 113

IPSLA-OPER_TRACE:OPER:2 ID:113, RTT=1

IPSLA-INFRA_TRACE:OPER:2 Updating result

```

Example 19-20 shows real-time output of an SLA operation with the **debug ip sla trace 2** command. The debug is displaying an unsuccessful trace of the IP SLA instance 2. You can see that the operation timed out between the source IP 192.168.1.11 and the destination IP 10.1.100.100. The results are then updated accordingly in the SLA statistics. This confirms that the IP SLA operation was not successful.

Example 19-20 *Debug Displaying an Unsuccessful IP SLA Operation*

```

R1#debug ip sla trace 2
IPSLA-INFRA_TRACE:OPER:2 slaSchedulerEventWakeup

IPSLA-INFRA_TRACE:OPER:2 Starting an operation

IPSLA-OPER_TRACE:OPER:2 source IP:192.168.1.11

IPSLA-OPER_TRACE:OPER:2 Starting icmpEcho operation - destAddr=10.1.100.100,
sAddr=192.168.1.11

IPSLA-OPER_TRACE:OPER:2 Sending ID: 205

IPSLA-OPER_TRACE:OPER:2 Timeout - destAddr=10.1.100.100, sAddr=192.168.1.11

IPSLA-INFRA_TRACE:OPER:2 Updating result

```

Object Tracking Troubleshooting

Object tracking enables you to dynamically control what will occur if the result of the tracking object is up or down. For example, you can attach an object to a static route; if the object is up, the route is installed in the routing table. If the object is down, the route

will not be installed in the routing table. With first-hop redundancy protocols (FHRPs), you can decrement or increment the priority based on the status of the object. For example, if the status of the tracking object is down, the FHRP priority is decremented.

With object tracking, you can track IP routes, IP SLA instances, interfaces, and groups of objects. For example, you can track an IP SLA instance that is using ICMP echoes. If the echo fails, the IP SLA instance fails, which brings the tracking object down. If the tracking object is tied to an FHRP, the priority is decremented, if the tracking object is tied to a static route, the static route is removed from the routing table.

To verify the configuration of a tracking object and the status of the tracking object, use the **show track** command. In Example 19-21 tracking object 1 exists on SW1. It is tracking the reachability of an IP route, 10.1.43.0/24. If the route is in the routing table, the object is up. If the route is not in the routing table, the object is down. The object is attached to (Tracked by:) HSRP Group 10.

Example 19-21 *Verifying the Configuration and Status of a Tracking Object (Up)*

```
SW1#show track
Track 1
  IP route 10.1.43.0 255.255.255.0 reachability
  Reachability is Up (EIGRP)
    1 change, last change 00:01:55
  First-hop interface is GigabitEthernet1/0/10
  Tracked by:
    HSRP Vlan10 10
```

In Example 19-22, the tracking object is down because the route to 10.1.43.0/24 is no longer in the routing table. Because it is attached to HSRP Group 10, an action based on the configuration of HSRP Group 10 would occur, such as decrementing the local HSRP priority.

Example 19-22 *Verifying the Configuration and Status of a Tracking Object (Down)*

```
SW1#
%TRACKING-5-STATE: 1 ip route 10.1.43.0/24 reachability Up->Down
SW1#show track
Track 1
  IP route 10.1.43.0 255.255.255.0 reachability
  Reachability is Down (no route)
    2 changes, last change 00:00:04
  First-hop interface is unknown
  Tracked by:
    HSRP Vlan10 10
```

SPAN and RSPAN Troubleshooting

SPAN and RSPAN enable you to take ingress/egress frames on a switchport, copy them, and send them to another port that has a management station running packet-capturing software attached.

With SPAN, you copy traffic from a source port on one switch to a destination port on the same switch, as shown in Figure 19-3. With RSPAN, you copy traffic from a source port on one switch to the destination port on a different switch, as shown in Figure 19-4.

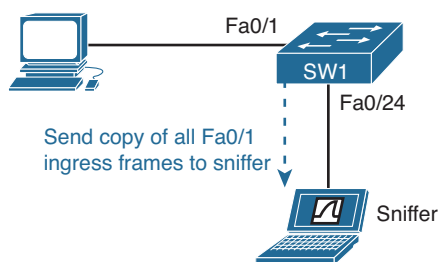


Figure 19-3 SPAN Topology

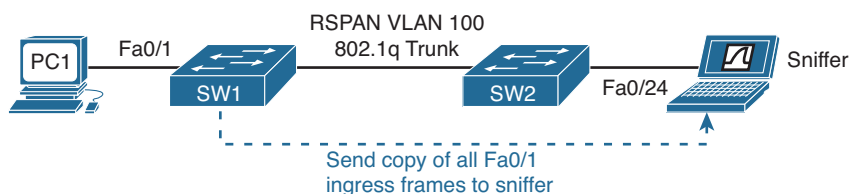


Figure 19-4 RSPAN Topology

Example 19-23 displays the configuration needed to successfully configure SPAN on SW1 in Figure 19-3. Troubleshooting issues will be minor for SPAN. Consider the following while troubleshooting SPAN issues:



- The source and destination session numbers must match to be part of the same SPAN session.
- The source interface/VLAN and destination interfaces have to be correctly identified.
- The direction of captured packets has to be correctly defined. (Default is both ingress and egress.)
- The interfaces have to be up/up.

Example 19-23 *Sample SPAN Configuration*

```
SW1#show run | section monitor
monitor session 1 source interface Fa0/1 rx
monitor session 1 destination interface Fa0/24
```

Example 19-24 displays the configuration needed to successfully configure RSPAN on SW1 and SW2 in Figure 19-4. Troubleshooting issues are more difficult for RSPAN. Consider the following while troubleshooting RSPAN issues:



- The source and destination session numbers must match locally to be part of the same RSPAN session. However, they do not have to match with the session numbers used on the remote switch.
- The source interface/VLAN and destination interface/vlan have to be correctly identified.
- The direction of captured packets has to be correctly defined. (Default is both ingress and egress.)
- The interfaces have to be up/up.
- The RSPAN VLAN must be configured and identified as an RSPAN VLAN.
- The RSPAN VLAN must be allowed across the trunk link (not pruned).
- STP cannot be blocking the RSPAN VLAN.

Example 19-24 *Sample RSPAN Configuration*

```
SW1#show run | section monitor
vlan 100
 name REMOTESPAN
 remote-span
monitor session 1 source interface fa0/1 rx
monitor session 1 destination remote vlan 100
```

```
SW2#show run | section monitor
vlan 100
 name REMOTESPAN
 remote-span
monitor session 1 source remote vlan 100
monitor session 1 destination interface fa0/24
```

To verify the SPAN or RSPAN sessions, use the command **show monitor**, as shown in Example 19-25. In this example, SW1 has an RSPAN session with an ID of 1 capturing frames ingress only on Fa0/1 and copying them to the RSPAN VLAN 100. SW2 has an RSPAN session with an ID of 1 capturing frames on the RSPAN VLAN 100 and sending them out Fa0/24. Using the **show monitor detail** command will display all the configured and nonconfigured parameters.

Example 19-25 *Verifying SPAN and RSPAN Sessions with show monitor Command*

```

SW1#show monitor
Session 1
-----
Type                : Remote Source Session
Source Ports        :
    RX Only         : Fa0/1
Dest RSPAN VLAN     : 100

SW2#show monitor
Session 1
-----
Type                : Remote Destination Session
Source RSPAN VLAN   : 100
Destination Ports    : Fa0/24
    Encapsulation    : Native
    Ingress          : Disabled

```

To verify RSPAN VLANs, use the command **show vlan remote-span**, as shown in Example 19-26. In this case, the remote span VLAN is 100.

Example 19-26 *Verifying RSPAN VLANs*

```

SW1#show vlan remote-span

Remote SPAN VLANs
-----
100

```

To verify whether an interface is configured as a destination SPAN port, use the command **show interfaces status**, as shown in Example 19-27. In this output, you can see that interface Fa0/24 on SW2 is in the monitoring status; therefore, it is no longer a normal switchport, and only monitored traffic will pass through it.

Example 19-27 *Verifying Destination SPAN/RSPAN Ports*

```

SW2#show interfaces status | i Port|Fa0/24

```

Port	Name	Status	Vlan	Duplex	Speed	Type
Fa0/24		monitoring	1	a-full	a-100	10/100BaseTX

Management Protocols and Tools Trouble Tickets

This section presents various trouble tickets relating to the topics discussed earlier in the chapter. The purpose of these trouble tickets is to give a process that you can follow when troubleshooting in the real world or in an exam environment. All trouble tickets in this section are based on the topology depicted in Figure 19-5.

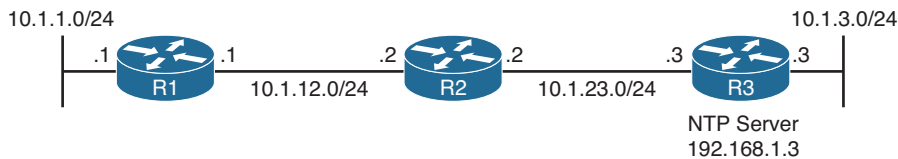


Figure 19-5 *Management Protocols and Services Trouble Tickets Topology*

Trouble Ticket 19-1

Problem: Router R1 is not synchronizing its local time with the NTP server (R3) at 192.168.1.3.

You begin troubleshooting by verifying the problem with the **show ntp status** command. Example 19-28 confirms that the clock is not synchronized. The stratum is also 16, which means unreachable.

Example 19-28 *Verifying NTP Status on R1*

```

R1#show ntp status
Clock is unsynchronized, stratum 16, reference is 65.85.84.72
nominal freq is 250.0000 Hz, actual freq is 250.0000 Hz, precision is 2**18
ntp uptime is 82600 (1/100 of seconds), resolution is 4000
reference time is D7811765.F28B7F98 (18:39:33.947 UTC Mon Jul 28 2014)
clock offset is 0.0000 msec, root delay is 0.00 msec
root dispersion is 2.44 msec, peer dispersion is 15937.50 msec
loopfilter state is 'CTRL' (Normal Controlled Loop), drift is -0.000000007 s/s
system poll interval is 64, last update was 678 sec ago.
  
```

Next you check whether the NTP server is reachable. You use the command **ping 192.168.1.3**, as shown in Example 19-29. In this example, the ping is successful.

Example 19-29 *Testing Connectivity to NTP Server with Ping*

```

R1#ping 192.168.1.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 36/52/64 ms
  
```

You access R3 and issue the **show ntp status** command to determine whether NTP is running. It is running, as shown in the output of Example 19-30. R3 is synchronized and has a stratum level of 1. It is referencing itself.

Example 19-30 *Verifying That NTP Is Operational on R3*

```

R3#show ntp status
Clock is synchronized, stratum 1, reference is .LOCL.
nominal freq is 250.0000 Hz, actual freq is 250.0000 Hz, precision is 2**16
  
```

```

ntp uptime is 1209300 (1/100 of seconds), resolution is 4000
reference time is D7811D8F.7696A1E4 (19:05:51.463 UTC Mon Jul 28 2014)
clock offset is 0.0000 msec, root delay is 0.00 msec
root dispersion is 0.42 msec, peer dispersion is 0.24 msec
loopfilter state is 'CTRL' (Normal Controlled Loop), drift is 0.000000000 s/s
system poll interval is 16, last update was 12 sec ago.

```

You decide to check whether an ACL is blocking NTP port 123. Back on R1, you use an extended traceroute, as shown in Example 19-31. You source the trace from 10.1.12.1 and specify a destination of 192.168.1.3. You also include the port number 123 for NTP. The result of the trace shows that at the hop 10.1.23.3 the trace is being administratively prohibited. In other words, it is being blocked by an ACL.

Example 19-31 Using a Trace to Determine Where Packets Fail

```

R1#traceroute
Protocol [ip]: ip
Target IP address: 192.168.1.3
Source address: 10.1.12.1
Numeric display [n]:
Timeout in seconds [3]:
Probe count [3]:
Minimum Time to Live [1]:
Maximum Time to Live [30]:
Port Number [33434]: 123
Loose, Strict, Record, Timestamp, Verbose[none]:
Type escape sequence to abort.
Tracing the route to 192.168.1.3
VRF info: (vrf in name/id, vrf out name/id)
  1 10.1.12.2 48 msec * 40 msec
  2 10.1.23.3 !A !A !A

```

The IP address 10.1.23.3 belongs to R3 according to Figure 19-5. Therefore, you access R3 and issue the **show ip interface brief** command and note that interface Gig1/0 is using that IP address, as shown in Example 19-32.

Example 19-32 Verifying IP Address Assignment

```

R3#show ip interface brief

```

Interface	IP-Address	OK?	Method	Status	Protocol
GigabitEthernet0/0	10.1.1.3.3	YES	NVRAM	up	up
GigabitEthernet1/0	10.1.23.3	YES	NVRAM	up	up
Loopback0	192.168.1.3	YES	NVRAM	up	up

You issue the command **show ip interface gigabitethernet1/0** and notice that ACL 100 is applied inbound on Gig1/0, as shown in Example 19-33.

Example 19-33 *Verifying ACLs on Gig1/0*

```
R3#show ip interface gigabitEthernet 1/0
GigabitEthernet1/0 is up, line protocol is up
  Internet address is 10.1.23.3/24
...output omitted...
  Multicast reserved groups joined: 224.0.0.10
  Outgoing access list is not set
  Inbound access list is 100
  Proxy ARP is enabled
  Local Proxy ARP is disabled
...output omitted...
```

Examining the output of **show access-list 100** in Example 19-34 shows that ACL 100 is blocking NTP packets and permitting all other packets.

Example 19-34 *Verifying ACL 100 Configuration*

```
R3#show access-lists 100
Extended IP access list 100
  10 deny udp any any eq ntp (23 matches)
  20 permit ip any any (819 matches)
```

Because ACL 100 is only blocking NTP packets while permitting all other packets, and you need NTP packets to be permitted, you decide to remove the ACL from the interface with the **no ip access-group 100 in** command in interface configuration mode on R3, as shown in 19-35.

Example 19-35 *Removing ACL from Interface and Verifying That It Is Removed*

```
R3#config t
Enter configuration commands, one per line. End with CNTL/Z.
R3(config)#int gig 1/0
R3(config-if)#no ip access-group 100 in
R3(config-if)#end
R3#show ip interface gig1/0
GigabitEthernet1/0 is up, line protocol is up
  Internet address is 10.1.23.3/24
...output omitted...
  Multicast reserved groups joined: 224.0.0.10
  Outgoing access list is not set
  Inbound access list is not set
  Proxy ARP is enabled
  Local Proxy ARP is disabled
...output omitted...
```

Now you go back to R1 and issue the **traceroute** command again, as shown in Example 19-36. In this case, it is successful.

Example 19-36 *Using a Trace to Determine Whether Packets Still Fail*

```

R1#traceroute
Protocol [ip]: ip
Target IP address: 192.168.1.3
Source address: 10.1.12.1
Numeric display [n]:
Timeout in seconds [3]:
Probe count [3]:
Minimum Time to Live [1]:
Maximum Time to Live [30]:
Port Number [33434]: 123
Loose, Strict, Record, Timestamp, Verbose[none]:
Type escape sequence to abort.
Tracing the route to 192.168.1.3
VRF info: (vrf in name/id, vrf out name/id)
  1 10.1.12.2 48 msec * 40 msec
  2 10.1.23.3 48 msec * 40 msec

```

You issue the **show ntp status** command again, as shown in Example 19-37, and notice that the problem is not solved. R1 is still not synchronized.

Example 19-37 *Verifying NTP Status on R1 After an ACL Is Removed*

```

R1#show ntp status
Clock is unsynchronized, stratum 16, reference is 65.85.84.72
nominal freq is 250.0000 Hz, actual freq is 250.0000 Hz, precision is 2**18
ntp uptime is 379600 (1/100 of seconds), resolution is 4000
reference time is D7811765.F28B7F98 (18:39:33.947 UTC Mon Jul 28 2014)
clock offset is 0.0000 msec, root delay is 0.00 msec
root dispersion is 46.99 msec, peer dispersion is 15937.50 msec
loopfilter state is 'CTRL' (Normal Controlled Loop), drift is -0.000000007 s/s
system poll interval is 64, last update was 3648 sec ago.

```

You now decide to check the NTP configuration on R1 and R3, as shown in Example 19-38. The first thing you notice is that R1 is configured with the **ntp server** command and that it is pointing to the correct address but the authentication key is incorrect when compared to the **ntp trusted key** command or the authentication key that is being used by R3. Therefore, the key should be 13 in this case, not 12. To fix this issue, you use the command **no ntp server 192.168.1.3 key 12** and issue the command **ntp server 192.168.1.3 key 13**, as shown in Example 19-39.

Example 19-38 *Verifying NTP Configuration*

```

R1#show run | section ntp
ntp authentication-key 13 md5 030752180500 7
ntp authenticate
ntp trusted-key 13

```

```
ntp server 192.168.1.3 key 12

R3#show run | section ntp
ntp authentication-key 13 md5 00071A150754 7
ntp authenticate
ntp trusted-key 13
ntp source Loopback0
ntp access-group serve-only 10
ntp master 1
```

Example 19-39 *Adjusting NTP Configuration*

```
R1#config t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#no ntp server 192.168.1.3 key 12
R1(config)#ntp server 192.168.1.3 key 13
R1(config)#end
```

To verify that the problem is solved you issue the command **show ntp status**. As shown in Example 19-40, the problem is still not solved. R1 is still not synchronized.

Example 19-40 *Verifying That the Problem Is Solved*

```
R1#show ntp status
Clock is unsynchronized, stratum 16, reference is 73.78.73.84
nominal freq is 250.0000 Hz, actual freq is 250.0000 Hz, precision is 2**18
ntp uptime is 447000 (1/100 of seconds), resolution is 4000
reference time is D7811765.F28B7F98 (18:39:33.947 UTC Mon Jul 28 2014)
clock offset is 0.0000 msec, root delay is 0.00 msec
root dispersion is 2.62 msec, peer dispersion is 15937.50 msec
loopfilter state is 'CTRL' (Normal Controlled Loop), drift is -0.000000007 s/s
system poll interval is 64, last update was 4322 sec ago.
```

You decide to enable debugging with the **debug ntp all** command on R1. The **debug** output shows that NTP packets are being sent but not received in Example 19-41. You issue the same command on R3, as shown in Example 19-42. In this case, R3 is receiving them but not responding. It states, *dropping message: RES_DONTSERVE restriction*. This indicates that there is an NTP access group on R3.

Example 19-41 *Debugging NTP Packets on R1*

```
R1#debug ntp all
NTP events debugging is on
NTP core messages debugging is on
NTP clock adjustments debugging is on
NTP reference clocks debugging is on
NTP packets debugging is on
R1#
```

```
.Jul 28 20:12:34.960: NTP message sent to 192.168.1.3, from interface
'GigabitEthernet1/0' (10.1.12.1).
```

Example 19-42 Debugging NTP Packets on R3

```
R3#debug ntp all
NTP message received from 10.1.12.1 on interface 'Loopback0' (192.168.1.3).
NTP Core(DEBUG): ntp_receive: message received
NTP Core(NOTICE): ntp_receive: dropping message: RES_DONTSERVE restriction.
```

Reviewing the NTP configuration on R1 and R3 again, as shown in Example 19-43, you notice that there is an NTP access group configured that will only respond to NTP packets sourced from IP addresses listed in ACL 10. You issue the command **show access-list 10** and note that only NTP packets sourced with an IP from 192.168.1.0 to 192.168.1.255 are permitted. Reviewing the configuration on R1 indicates that packets will be sourced with the IP address of the interface the packets will be sent from. Therefore, you need to include the **ntp source** command on R1 to control the source IP address of the packets.

Example 19-43 Reviewing NTP Configuration

```
R1#show run | section ntp
ntp authentication-key 13 md5 030752180500 7
ntp authenticate
ntp trusted-key 13
ntp server 192.168.1.3 key 13

R3#show run | section ntp
ntp authentication-key 13 md5 00071A150754 7
ntp authenticate
ntp trusted-key 13
ntp source Loopback0
ntp access-group serve-only 10
ntp master 1

R3#show access-lists 10
Standard IP access list 10
 10 permit 192.168.1.0, wildcard bits 0.0.0.255 (289 matches)
```

On R1, you issue the **show ip interface brief** command, as shown in Example 19-44, and notice that interface Loopback 0 is using the IP address 192.168.1.1. Therefore, on R1, you issue the command **ntp source loopback0**, as also seen in Example 19-44.

Example 19-44 Adding the NTP Source Command to R1

```
R1#show ip interface brief
```

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	unassigned	YES	NVRAM	administratively down	down
GigabitEthernet0/0	10.1.1.0	YES	NVRAM	up	up


```
GigabitEthernet1/0    10.1.12.1      YES NVRAM  up          up
Loopback0             192.168.1.1    YES manual up          up
R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#ntp source loopback 0
```

Now you issue the command **show ntp status**, as shown in Example 19-45, on R1, and the clock is synchronized with the NTP server at 192.168.1.3. Using the command **show ntp association detail**, you confirm that authentication was successful as well. (Note that the clocks can take some time to synchronize; it is not immediate.)

Example 19-45 *Verifying the Issue Is Solved*

```
R1#show ntp status
Clock is synchronized, stratum 2, reference is 192.168.1.3
nominal freq is 250.0000 Hz, actual freq is 249.9966 Hz, precision is 2**18
ntp uptime is 549900 (1/100 of seconds), resolution is 4016
reference time is D7812C43.F6589518 (20:08:35.962 UTC Mon Jul 28 2014)
clock offset is 74.6174 msec, root delay is 19.98 msec
root dispersion is 111.38 msec, peer dispersion is 1.49 msec
loopfilter state is 'CTRL' (Normal Controlled Loop), drift is 0.000013584 s/s
system poll interval is 64, last update was 9 sec ago.

R1#show ntp associations detail
192.168.1.3 configured, ipv4, authenticated, our_master, sane, valid, stratum 1
ref ID .LOCL., time D7812C3F.7685D321 (20:08:31.462 UTC Mon Jul 28 2014)
our mode client, peer mode server, our poll intvl 64, peer poll intvl 64
...output omitted...
```

Exam Preparation Tasks

As mentioned in the section “How to Use This Book” in the Introduction, you have a couple of choices for exam preparation: the exercises here; Chapter 22, “Final Preparation;” and the exam simulation questions on the CD-ROM.

Review All Key Topics

Review the most important topics in this chapter, noted with the Key Topic icon in the outer margin of the page. Table 19-2 lists a reference of these key topics and the page numbers on which each is found.



Table 19-2 *Key Topics for Chapter 19*

Key Topic Element	Description	Page Number
List	Outlines the reasons why an NTP client does not synchronize with an NTP server	818
Example 19-1	Verifying the status of NTP on a client	819
Example 19-3	Verifying details of the NTP time servers associated with the client	820
Paragraph	Discusses what to keep in mind when troubleshooting issues related to syslog	821
List	Outlines the reasons why SNMPv2c may not be operating as expected	823
List	Outlines the reasons why SNMPv3 may not be operating as expected	824
Example 19-9	Verifying SNMP groups	825
Example 19-10	Verifying SNMP users	826
Example 19-11	Verifying SNMP hosts	826
Example 19-12	Verifying SNMP views	826
List	Outlines the issues that should be considered while troubleshooting Cisco IOS IP SLA	828
Example 19-17	Output of show ip sla statistics	831
Paragraph	Describes how to verify the statistics on a Cisco IOS IP SLA responder	832
List	Examines the issues that you might experience with SPAN	835
List	Examines the issues that you might experience with RSPAN	836

Define Key Terms

Define the following key terms from this chapter and check your answers in the glossary:

NTP, NTP server, NTP client, stratum, syslog, SNMPv2c, SNMPv3, community string, traps, informs, NMS, noAuthNoPriv, authNoPriv, authPriv, OID, SNMP view, IP SLA, IP SLA source, IP SLA responder, object tracking, SPAN, RSPAN, sniffer

Command Reference to Check Your Memory

This section includes the most important **show** and **debug** commands covered in this chapter. It might not be necessary to memorize the complete syntax of every command, but you should be able to remember the basic keywords that are needed.

To test your memory of the commands, cover the right side of Table 19-3 with a piece of paper, read the description on the left side, and then see how much of the command you can remember.

The 300-135 TSHOOT exam focuses on practical, hands-on skills that are used by a networking professional. Therefore, you should be able to identify the commands needed to successfully troubleshoot the topics and concepts covered in this chapter.

Table 19-3 *show and debug Commands*

Task	Command Syntax
Displays the NTP configuration within the running configuration.	show run section ntp
Displays the status of NTP on a device (including whether the device is synchronized), the stratum level, and the IP address of the NTP server (device synchronized with).	show ntp status
Displays a summary of all the NTP devices the local device is associated with, including which one the local device is currently synchronized with.	show ntp associations
Displays detailed information of all the NTP devices the local device is associated with, including who the local device is currently synchronized with, the stratum levels, and whether authentication was successful.	show ntp associations detail
Displays real-time information about NTP packets sent and received as well as event changes.	debug ntp all

Task	Command Syntax
Displays the status of syslog on a device, such as whether it is enabled or disabled, the severity level for each logging option, and the server syslog messages will be sent to.	show logging
Displays the SNMP configuration in the running configuration.	show run section snmp
Displays the SNMP groups configured on the local device, including the security model associated with the group, the read-only and read-write views associated with the groups, in addition to any access lists associated with the group.	show snmp group
Displays the local views and OIDs associated with each view.	show snmp view
Displays the SNMP users configured locally, including their authentication protocol, privacy protocol, and the group they are a member of.	show snmp user
Displays the parameters necessary to send SNMP traps and informs to an SNMP NMS. It includes the server IP, port number, whether traps or informs are sent, in addition to the user information for authentication and security purposes.	show snmp host
Displays the IP SLA configuration in the running configuration.	show run section sla
Displays the IP SLA supported operation types, number of configured entries, active entries, pending entries, and inactive entries.	show ip sla application
Displays the configuration of the IP SLA entries, including the target and source address, target and source port, ToS parameter, interval, schedule, and threshold.	show ip sla configuration
Displays the results of the IP SLA operation. The type of operation will determine the statistics that are displayed.	show ip sla statistics
Displays the information about the IP SLA responder such as the control port, the number of control messages received, the number of errors, and the recent sources of IP SLA probes.	show ip sla responder

Task	Command Syntax
Displays the sending and reception of IP SLA messages in real time.	debug ip sla trace
Displays the status of tracking objects, including the object being tracked, the status of the object, and the protocols or service it is attached to.	show track
Displays the configuration of SPAN and RSPAN in the running configuration.	show run section monitor
Displays the SPAN and RSPAN sessions configured on the device, including the source and destination ports and VLANs and the direction the packets will be captured.	show monitor
Displays the RSPAN configuration VLANs.	show vlan remote-span
Displays the status of an interface, the VLAN it is associated with, in addition to the duplex, speed, and type. For SPAN/RSPAN, the destination port is listed as Monitoring in the status column.	show interfaces status

This page intentionally left blank



This chapter covers the following topics:

- **Console and vty Access Troubleshooting:** This section explains how to identify and troubleshoot issues relating to console and vty access, including Telnet and SSH.
- **Cisco IOS AAA Troubleshooting:** This section examines the AAA authentication process and the issues that you might face when using local AAA to authenticate remote access.
- **Management Access Trouble Tickets:** This section provides trouble tickets that demonstrate how you can use a structured troubleshooting process to solve a reported problem.

Troubleshooting Management Access

To troubleshoot issues with Cisco routers and switches, you need access to them. You can access them physically using the console port or remotely with the vty lines. If you attempt to access a device for management purposes, and access fails, you will need to troubleshoot why this failure is occurring before you can troubleshoot the other issues.

This chapter covers the different reasons why access to the console and vty lines might fail and how you can identify those reasons. In addition, you will learn the issues that may arise when using Cisco IOS AAA (authentication, authorization, and accounting) authentication.

“Do I Know This Already?” Quiz

The “Do I Know This Already?” quiz allows you to assess whether you should read this entire chapter thoroughly or jump to the “Exam Preparation Tasks” section. If you are in doubt about your answers to these questions or your own assessment of your knowledge of the topics, read the entire chapter. Table 20-1 lists the major headings in this chapter and their corresponding “Do I Know This Already?” quiz questions. You can find the answers in Appendix A, “Answers to the ‘Do I Know This Already?’ Quizzes.”

Table 20-1 “Do I Know This Already?” Section-to-Question Mapping

Foundation Topics Section	Questions
Console and vty Access Troubleshooting	1–8
Cisco IOS AAA Troubleshooting	9–10

Caution The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you do not know the answer to a question or are only partially sure of the answer, you should mark that question as wrong for purposes of the self-assessment. Giving yourself credit for an answer that you correctly guess skews your self-assessment results and might provide you with a false sense of security.

1. Which of the following are the default serial terminal settings for a Cisco router or switch? (Choose two answers.)
 - a. 9600 baud
 - b. 16 data bits
 - c. 1 stop bit
 - d. Parity
2. What type of cable is used to connect to the console port?
 - a. Straight-through
 - b. Crossover
 - c. Rollover
 - d. Coaxial
3. Which command enables you to define which protocols will be used for remote access to the Cisco device via the vty lines?
 - a. `transport input`
 - b. `login`
 - c. `login local`
 - d. `exec`
4. Which command enables you to specify that SSH access will be authenticated using the local database?
 - a. `login`
 - b. `login local`
 - c. `login authentication default`
 - d. `transport input ssh`
5. Which command enables you to filter the users that are allowed to remotely access the device via the vty lines?
 - a. `access-class {acl_name | acl_number} in`
 - b. `access-class {acl_name | acl_number} out`
 - c. `ip access-group {acl_name | acl_number} in`
 - d. `ip access-group {acl_name | acl_number} out`

6. Which port is used by SSH?
 - a. 21
 - b. 22
 - c. 23
 - d. 25
7. What does an SSH version of 1.99 represent?
 - a. SSHv1 is only enabled.
 - b. SSHv1.99 is only enabled.
 - c. SSHv2 is only enabled.
 - d. SSHv1 and v2 are enabled.
8. Which encryption level uses SHA-256?
 - a. 0
 - b. 4
 - c. 5
 - d. 7
9. Which command successfully configures a user-defined method list on a Cisco IOS device that uses the database on the device if the external server is not available for authentication?
 - a. `aaa authentication login default local group radius`
 - b. `aaa authentication login default group radius local`
 - c. `aaa authentication login REMOTE_ACCESS local group radius`
 - d. `aaa authentication login MANAGEMENT_ACCESS group radius local`
10. Your Cisco router is configured with the following command: `aaa authentication login default group radius local`
What will occur during login if the local database does not contain any username and password when it is checked?
 - a. The RADIUS server will be used for authentication.
 - b. Authentication will fail.
 - c. The user will be granted access.
 - d. The line password will be used.

Foundation Topics

Console and vty Access Troubleshooting

You can access a Cisco IOS router or switch for management purposes in various ways. There is the console line, which is used when you have physical access to the device, or when you are using an access server. There are the vty lines, which provide remote connectivity using Telnet or Secure Shell (SSH), so device management can be done from a remote location. Regardless of the method you use for management purposes, at some point you will likely end up having to troubleshoot why you are not able to connect to a device so that you can troubleshoot another issue that has been presented to you. Therefore, you potentially have to solve one issue to get to the next issue.

This section explains the reasons why management access to a Cisco IOS router or switch may fail, how you can troubleshoot why it is occurring and how you can fix it. You will also learn how to troubleshoot issues related to Cisco IOS AAA authentication which can be used during the authentication process for validating management access.

Console Access Troubleshooting

The default out-of-the-box method of accessing Cisco routers and switches is via the console port. Here are some things you should look out for when troubleshooting console access:



- **Has the correct COM port been selected in the terminal program?** Most times, multiple COM ports are displayed in the terminal program; however, the last one listed is usually the correct one to use. If it is not, try a different one. This is really a trial-and-error process.
- **Are the terminal programs settings configured correctly?** Cisco devices use the following default values: 9600 baud, 8 data bits, 1 stop bit, no parity.
- **Is a line password used to authenticate to the console?** If a line password is being used, the **login** command needs to be configured as well. The **login** command and a line password are not configured by default.
- **Is a local username and password used to authenticate to the console?** If local authentication is being used, a username and password need to exist in the local database, and the **login local** command is required.
- **Is an AAA server used to authenticate to the console?** If AAA authentication is being used, a method list needs to be defined with the **login authentication {default | list_name}** command in line console configuration mode.
- **Are the correct cable and drivers being used to connect to the console port?** Check your device's documentation to see what is needed. Newer devices are using a

mini USB port as the console port (drivers required on PC), whereas older devices are using the serial to RJ-45 console (rollover) cable.

vty Access Troubleshooting

Most devices will be administered remotely via the vty lines, which support protocols such as Telnet and SSH for remote access. Telnet is not recommended because all traffic between the management station and the router/switch is sent in plain text. If a malicious user is able to capture the packets, that user will be able to see all the data that was transmitted back and forth. If you use SSH, the packets will be encrypted, ensuring that if they are captured, they will not be readable.

Telnet

Consider the following while troubleshooting Telnet access to a device:



- **Is the IP address of the remote router/switch reachable?** You can test this with the `ping` command.
- **Are the correct transport protocols defined for the line?** By default with IOS 15.0 and later, Telnet and SSH are allowed, and if other protocols are supported, they are typically allowed as well; however, with the `transport input` command, you can change which transport protocols are allowed. You can verify the allowed protocols with the command `show line vty line_number | include Allowed`, as shown in Example 20-1. In this example, Telnet and SSH are allowed for inbound and outbound connections.
- **Is the line configured to ask the user for credentials?** By default, it is. The `login` command tells the line to prompt the user for a password, as shown in Example 20-2. However, if you need to authenticate the user via the local database, the `login local` command is required, and if you need to authenticate the user via AAA, the `login authentication {default | list_name}` command is required.
- **Is a password specified?** Because the `login` command is enabled by default, a password is required. If it is not set, the error message *Password required, but none set* will appear. If you are using the `login local` command or AAA, you will be prompted for a username and password instead. However, if there is none stored in the database of either, your login will be invalid and fail.
- **Is there an ACL defining which management stations based on IP address can access the router/switch?** Example 20-3 shows ACL 1 applied to the vty lines. It only allows access from the IP address 192.168.1.11. Notice the explicit deny that was added so that we could keep track of the number of denied remote access attempts that have occurred (7 in this case). To receive a log message indicating which IP address was denied, you need to add the `log` keyword to the end of the explicit deny entry in the ACL. A log message appears as follows if the `log` keyword is added: `%SEC-6-IPACCESSLOGS: list 1 denied 10.1.12.2 1 packet.`

- **Are all vty lines busy?** By default, there are five vty lines on Cisco routers and switches, numbered 0 to 4. Some devices have more. However, regardless of the number, if all the lines have established connections, a new connection will not be made, as shown in Example 20-4. In this case, the **show users** command on SW1 indicates there is one console connection and five vty connections on lines 0 to 4. The next device that tries to telnet will be refused and receive the message *Password required, but none set*, even though that is not technically the issue. If you need to manually clear the lines, use the **clear line** command followed by the *line number* specified before vty, as shown in Example 20-4, not the actual vty number listed after vty.
- **Is there an ACL in the path between the client and the device blocking port 23?** Telnet uses TCP port 23. If there is an ACL configured on a router or firewall blocking port 23, you will be unable to make a successful Telnet connection.

Example 20-1 Verifying Transport Protocols for a Line

```
SW1#show line vty 0 | include Allowed
Allowed input transports are telnet ssh.
Allowed output transports are telnet ssh.
```

Example 20-2 Verifying the vty login Command

```
SW1#show run | section line vty
line vty 0 4
  login
```

Example 20-3 Verifying ACLs Used to Secure Management Access

```
SW1#show run | section line vty
line vty 0 4
  access-class 1 in
  password cisco
  login
DSW1#show ip access-lists 1
Standard IP access list 1
  10 permit 192.168.1.11 (4 matches)
  20 deny any (7 matches)
```

Example 20-4 Verifying Which Lines Are Being Used

```
SW1#show users
```

	Line	User	Host(s)	Idle	Location
*	0 con 0		idle	00:00:00	
	1 vty 0		idle	00:00:42	10.1.1.2
	2 vty 1		idle	00:00:48	10.1.10.1
	3 vty 2		idle	00:00:55	10.1.20.1
	4 vty 3		idle	00:00:47	10.1.23.3
	5 vty 4		idle	00:00:41	10.1.43.4

SSH

With Secure Shell (SSH), you will experience the same issues as described with Telnet, in addition to the following:



- **Is the correct version of SSH specified?** By default both version 1 and 2 are enabled. However, with the **ip ssh version {1 | 2}** command it can be changed to just 1 or 2. If clients are connecting with v2 and the device is configured for v1, the SSH connection will fail, and the same is true if clients are using v1 and the devices are configured for v2. To check the version of SSH running use the **show ip ssh** command, as shown in Example 20-5. If it states version 1.99 it means version 1 and 2 are running. If it states version 1 then SSHv1 is running, and if it states version 2 then SSHv2 is running.
- **Has the correct login command been specified?** SSH uses a username and password for authentication. Therefore, the **login** command will not work in this case because it only requests a password. You need to use the **login local** command to authenticate with the local database or the **login authentication {default | list_name}** command to authenticate with an AAA server. As shown in Example 20-6, the **login local** command has been specified.
- **Has the correct size key been specified?** SSHv2 uses an RSA key size of 768 or greater. If you were using a smaller key size with SSHv1 and then switched to SSHv2, you would need to create a new key with the correct size; otherwise, SSHv2 would not work. If you are using SSHv2 but accidentally specify a key size less than 768, SSHv2 connections will not be allowed.
- **Is there an ACL in the path between the client and the device blocking port 22?** SSH uses TCP port 22. If an ACL blocking port 22 is configured on a router or firewall, you will be unable to make a successful SSH connection.

Example 20-5 Verifying the SSH Version

```
SW1#show ip ssh
SSH Enabled - version 1.99
Authentication timeout: 120 secs; Authentication retries: 3
Minimum expected Diffie Hellman key size : 1024 bits
IOS Keys in SECSH format (ssh-rsa, base64 encoded):
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQgQDtRqwdcEI+aGEXYmk1h4G6pSJW1th6/Ivg4BCp19tO
BmdoW6NZahL2SxdzjKW8VIBjO1lVeaMfdmvKlpLjU1x7JDAkPs4Q39kzdPHY74MzD1/u+Fwvir8O5AQO
rUMkc5vuVEHFVc4WxQsxH4Q4Df10a6Q3UA0tnL4E0a7ez/imHw==
```

Example 20-6 Verifying the vty Line Configuration

```
SW1#show run | s line vty
line vty 0 4
password cisco
login local
```

To verify the current SSH connections, use the **show ssh** command. In Example 20-7, there is an SSHv2 inbound and outbound connection with the username cisco. The session is using aes128-cbc encryption and the hashed message authentication code (HMAC) hmac-sha1.

Example 20-7 Verifying SSH Connections

```
SW1#show ssh
Connection Version Mode Encryption Hmac State Username
0 2.0 IN aes128-cbc hmac-sha1 Session started cisco
0 2.0 OUT aes128-cbc hmac-sha1 Session started cisco
%No SSHv1 server connections running.
```

Password Encryption Levels



By default, all passwords are stored in clear text within the IOS configuration. It is recommended that passwords either be encrypted or hashed in the configuration for security reasons. Example 20-8 displays a sample output of the passwords stored in the running configuration. A level of 0 indicates no encryption. A level of 4 indicates that SHA256 was used. A level of 5 indicates that message digest 5 (MD5) was used. A level of 7 indicates that Type-7 encryption was used. The levels from strongest to weakest are 4, 5, 7, and then 0. To implement Type-7 encryption, you issue the **service password-encryption** command. To implement level 4 encryption, you use the **secret** keyword when specifying a password. In IOS 15.0 and later, level 4 is the default for the **secret** keyword. If you need to use level 5 (default on 12.4 and earlier), you will have to use the **secret 5** keyword and specify the actual MD5 hash and not the clear-text password.

Example 20-8 Verifying Password Security Levels

```
SW1#show run | section username
username admin password 0 letmein
username administrator password 7 082D495A041C0C19
username cisco secret 4 tnhtc92DXBhelxjYk8LWJrPV36S2i4ntXrpb4RFmfqY
username Raymond secret 5 $1$sHu.$sIjLazYcNokRrgAjhyhxn0
```

Cisco IOS AAA Troubleshooting

AAA is a framework that provides authentication, authorization, and accounting to secure the management plane. The 300-135 TSHOOT exam objectives focus on AAA authentication using the local database; therefore, in this section, the troubleshooting focus centers on this. However, because most organizations use AAA servers, we include a RADIUS server in our example so that you can see what occurs when the RADIUS server is not accessible and the router or switch falls back to local authentication.

Example 20-9 provides a sample Cisco IOS AAA configuration for management access to the console and vty lines. As you review the output, consider the following items you should keep in mind while troubleshooting Cisco IOS AAA authentication:



- **AAA needs to be enabled:** AAA is disabled by default on Cisco routers and switches. To enable AAA, use the **aaa new-model** command. Once you do this, local authentication is immediately applied to all lines except the console line. Therefore, you will not be able to access the device remotely if no username and password exists in the local database. Console access is still capable with no username or password.
- **AAA relies on the local username and password database or an AAA server such as RADIUS or TACACS+:** By default, AAA uses the local username and password database for authentication. If no username and password exists that can be used for remote access, authentication will fail. Therefore, if you are using local authentication, a username and password needs to exist on the local device. However, if you are using an AAA server, you should still configure at least one username and password in the local database that can be used for fallback purposes in case the AAA server is not available. In Example 20-9, the username *admin* with a password of *letmein* exists.
- **A method list defines the authentication methods:** When no method list exists, the vty lines use the local username and password database by default. However, with the method list, you can define what methods of authentication will be used and in what order. In Example 20-9, a user-defined method list for login authentication called `MANAGEMENT_ACCESS` will use RADIUS servers first, and if they are not accessible, local authentication will be used. If there is no username or password in the database, authentication fails.
- **AAA method lists are applied to the lines:** The method list that will be used to define how authentication will occur for the vty lines or console line needs to be applied with the **login authentication {default | list_name}** command. In Example 20-9, the `MANAGEMENT_ACCESS` method list is attached to the vty lines.

Example 20-9 Verifying Cisco IOS AAA Configuration

```
R1#show run | section username|aaa|line vty
username admin password 0 letmein
aaa new-model
aaa authentication login MANAGEMENT_ACCESS group radius local
line vty 0 4
  password cisco
  login authentication MANAGEMENT_ACCESS
```

You can use the **debug aaa authentication** command to verify the authentication process in real time. You can use the **debug radius authentication** command to view the RADIUS authentication processes in real time. You can use the **debug aaa protocol local** command to view local authentication processes in real time.

In Example 20-10, all three **debug** commands have been enabled on R1. When a user attempts to telnet from SW1 to R1, R1 invokes the method list `MANAGEMENT_ACCESS`, which, based on the configuration in Example 20-9, will use `RADIUS` first and then local authentication if the `RADIUS` server is not accessible. R1 asks the user for his credentials, and then sends a `RADIUS` packet to the address 10.1.100.100 on port 1645. Notice how the request to the `RADIUS` server fails because there is no response from the `RADIUS` server in this example. Therefore, R1 resorts to local authentication and checks the username and password against the local database. However, the user provided the wrong username/password combination, and the process starts over again by choosing the method list `MANAGEMENT_ACCESS` and asking the user for his credentials again.

Example 20-10 Debugging Cisco IOS AAA Configuration

```
R1#debug aaa authentication
AAA Authentication debugging is on
R1#debug radius authentication
Radius protocol debugging is on
Radius protocol brief debugging is off
Radius protocol verbose debugging is off
Radius packet hex dump debugging is off
Radius packet protocol (authentication) debugging is on
Radius packet protocol (accounting) debugging is off
Radius elog debugging debugging is off
Radius packet retransmission debugging is off
Radius server fail-over debugging is off
Radius elog debugging debugging is off
R1#debug aaa protocol local
AAA Local debugs debugging is on
R1#
AAA/LOCAL: exec
AAA/BIND(0000004D): Bind i/f
AAA/LOCAL: new_ascii_login: tty 76EA4F4 idb 0
AAA/AUTHEN/LOGIN (0000004D): Pick method list 'MANAGEMENT_ACCESS'
RADIUS/ENCODE(0000004D): ask "Username: "
RADIUS/ENCODE(0000004D): send packet; GET_USER
R1#
RADIUS/ENCODE(0000004D): ask "Password: "
RADIUS/ENCODE(0000004D): send packet; GET_PASSWORD
...output omitted...
RADIUS(0000004D): Sending a IPv4 Radius Packet
RADIUS(0000004D): Send Access-Request to 10.1.100.100:1645 id 1645/11, len 69
RADIUS:  authenticator 09 9E 3E A4 D9 F9 03 87 - 85 02 41 47 BD 72 8F ED
RADIUS:  User-Name           [1]   7  "admin"
RADIUS:  User-Password       [2]  18  *
RADIUS:  NAS-Port            [5]   6   1
RADIUS:  NAS-Port-Id         [87]  6   "tty1"
RADIUS:  NAS-Port-Type       [61]  6   Virtual           [5]
```

```

RADIUS: NAS-IP-Address      [4]   6   10.1.100.1
R1#
RADIUS(0000004D): Started 5 sec timeout
R1#
RADIUS(0000004D): Request timed out
RADIUS: Retransmit to (10.1.100.100:1645,1646) for id 1645/11
RADIUS(0000004D): Started 5 sec timeout
R1#
RADIUS(0000004D): Request timed out
RADIUS: Retransmit to (10.1.100.100:1645,1646) for id 1645/11
RADIUS(0000004D): Started 5 sec timeout
R1#
RADIUS(0000004D): Request timed out
RADIUS: Retransmit to (10.1.100.100:1645,1646) for id 1645/11
RADIUS(0000004D): Started 5 sec timeout
R1#
RADIUS(0000004D): Request timed out
RADIUS: No response from (10.1.100.100:1645,1646) for id 1645/11
RADIUS/DECODE: No response from radius-server; parse response; FAIL
RADIUS/DECODE: Case error(no response/ bad packet/ op decode);parse response; FAIL
AAA/LOCAL/LOGIN(0000004D): check username/password
AAA/LOCAL/LOGIN(0000004D): invalid username/password
R1#
AAA/AUTHEN/LOGIN (0000004D): Pick method list 'MANAGEMENT_ACCESS'
RADIUS/ENCODE(0000004D): ask "Username: "
RADIUS/ENCODE(0000004D): send packet; GET_USER
R1#

```

By default, many Cisco IOS devices use ports 1645 and 1646 for RADIUS and port 49 for TACACS. In Example 20-10, you can see that R1 is attempting to communicate to the RADIUS server at 10.1.100.100 using ports 1645 and 1646. However, RADIUS ports were changed, and the current standard is to use ports 1812 and 1813. Therefore, you need to be aware of which ports are being used on the server and configure your IOS device appropriately. Also, if RADIUS or TACACS+ communication between the authenticator (Cisco IOS device) and the authentication server (RADIUS or TACACS+ server) is not successful, you should verify that any ACLs between these devices are permitting traffic for the RADIUS or TACACS+ ports being used.

Management Access Trouble Tickets

This section presents various trouble tickets relating to the topics discussed earlier in the chapter. The purpose of these trouble tickets is to give a process that you can follow when troubleshooting in the real world or in an exam environment. All trouble tickets in this section are based on the topology depicted in Figure 20-1.

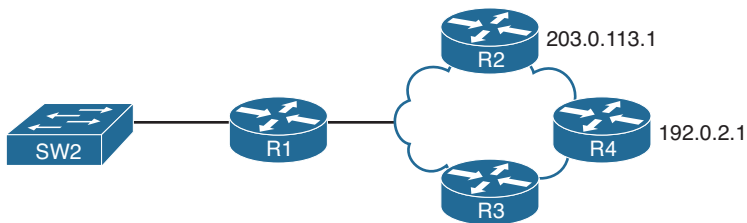


Figure 20-1 *Management Access Trouble Tickets Topology*

Trouble Ticket 20-1

Problem: A security audit has been done, and the report shows that R4 is accessible via Telnet and SSH when it should only be accessible via SSH.

You commence troubleshooting by verifying the problem. In Example 20-11, you telnet to R4's IP address 192.0.2.1, and it is successful. You then use SSHv2, and it is successful as well.

Example 20-11 *Verifying the Problem*

```
SW2#telnet 192.0.2.1
Trying 192.0.2.1 ... Open

User Access Verification

Username: TSHOOT
Password:
R4>exit

[Connection to 192.0.2.1 closed by foreign host]
SW2#ssh -v 2 -l TSHOOT 192.0.2.1
Password:
R4>exit

[Connection to 192.0.2.1 closed by foreign host]
SW2#
```

You access R4 and issue the command **show line vty 0 | i Allowed input transports** to verify which protocols can be used to establish a remote connection with R4. According to the output in Example 20-12, LAT, PAD, Telnet, rlogin, mop, v120, SSH, and NASI are all allowed.

Example 20-12 *Identifying Allowed Protocols*

```
R4#show line vty 0 | i Allowed input transports
Allowed input transports are lat pad telnet rlogin mop v120 ssh nasi.
```

Next you issue the command **show run | section line vty**, as shown in Example 20-13, and notice that there is no **transport input** command controlling which protocols are permitted.

Example 20-13 *Verifying the vty Configuration*

```
R4#show run | section line vty
line vty 0 4
  password cisco
  login local
```

Because you only need to allow SSHv2, you issue the command **transport input ssh** in line vty configuration mode, as shown in Example 20-14, to prevent Telnet access.

Example 20-14 *Modifying the vty Configuration*

```
R4#config t
Enter configuration commands, one per line. End with CNTL/Z.
R4(config)#line vty 0 4
R4(config-line)#transport input ssh
```

To verify that the problem is solved, you attempt to telnet from SW2 to R4 again, but this time the connection is refused by R4. However, SSHv2 still works as expected. The problem is solved, as shown in Example 20-15.

Example 20-15 *Verifying That the Problem Is Solved*

```
SW2#telnet 192.0.2.1
Trying 192.0.2.1 ...
% Connection refused by remote host

DSW2#ssh -v 2 -l TSHOOT 192.0.2.1
Password:
R4>exit

[Connection to 192.0.2.1 closed by foreign host]
SW2#
```

Trouble Ticket 20-2

Problem: For security reasons, when accessing R2 via Telnet, the user should be prompted for a username and password. However, users are only being prompted for a password.

You commence the troubleshooting process by verifying the problem. You attempt to Telnet from SW2 to R2 at the IP address 203.0.113.1. As you can see in the output of Example 20-16, you are only being asked for a password. You have a feeling that the **login local** command is missing.

Example 20-16 *Verifying the Problem*

```
SW2#telnet 203.0.113.1
Trying 203.0.113.1 ... Open

User Access Verification

Password:
R2>
```

On R2, you issue the command **show run | section line vty**, as shown in Example 20-17, to verify that the **login local** command is missing. According to the output, it is.

Example 20-17 *Verifying the vty Configuration on R2*

```
R2#show run | section line vty
line vty 0 4
    password cisco
```

You enter live vty mode, as shown in Example 20-18, and issue the **login local** command, but it fails to execute. The error message indicates that **local** is not a valid option. You then use syntax help, and it indicates that **authentication** is the only valid option. This means that AAA is in use.

Example 20-18 *Configuring Local Authentication on R2*

```
R2#config t
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#line vty 0 4
R2(config-line)#login local
                        ^
% Invalid input detected at '^' marker.

R2(config-line)#
R2(config-line)#login ?
    authentication Authentication parameters.
```

On R4, you issue the command **show run | section aaa** to verify the AAA configuration, as shown in Example 20-19. It appears that the AAA authentication method list was configured incorrectly. It is only using the line for authentication. If you want to use a username and password, you need to use the local database or an AAA server. In this case, you are using the local database; therefore, you need to specify **local** as a method instead of **line**.

Example 20-19 *Verifying AAA Configuration*

```
R2#show run | section aaa
aaa new-model
aaa authentication login default line
```

In Example 20-20, you enter the command **no aaa authentication login default line** and issue the command **aaa authentication login default local**.

Example 20-20 *Modifying AAA Configuration*

```
R2#config t
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#no aaa authentication login default line
R2(config)#aaa authentication login default local
R2(config)#end
```

You then attempt to telnet again, and this time, as shown in Example 20-21, you are prompted for a username and password.

Example 20-21 *Verifying Issue Is Solved*

```
SW2#telnet 203.0.113.1
Trying 203.0.113.1 ... Open

User Access Verification

Username: TSHOOT
Password:

R2>
```

Trouble Ticket 20-3

Problem: A user is trying to manage R4 via an SSHv2 connection, but the connection fails.

You begin by verifying the problem with an attempt to establish an SSHv2 connection to R4 from SW2. It fails, as shown in Example 20-22, confirming the problem. However, you are happy that you verified the problem because the error message is giving you more information. This error usually means that the remote device does not support SSHv2. However, you know without a doubt that R4 does support SSHv2. Therefore, you hypothesize that something is misconfigured on R4.

Example 20-22 *Verifying the Problem*

```
SW2#ssh -v 2 -l TSHOOT 192.0.2.1
[Connection to 192.0.2.1 aborted: error status 0]
```

You access R4 and issue the command **show ip ssh** and confirm that version 1.5 is being used, as shown in Example 20-23. You have a feeling that the SSH RSA key is not large enough for SSHv2. However, it does not list the key size. Therefore, you decide to spot the difference with R2. On R2, as shown in Example 20-24, you issue the **show ip ssh** command and notice that v2 is enabled and that the SSH RSA key is significantly larger.

Example 20-23 *Verifying SSH Configuration on R4*

```
R4#show ip ssh
SSH Enabled - version 1.5
Authentication timeout: 120 secs; Authentication retries: 3
Minimum expected Diffie Hellman key size : 1024 bits
IOS Keys in SECSH format(ssh-rsa, base64 encoded):
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQADIEIKgD03hu48qw9Wy/K5JRB/Gf4YQ8mi0iEo/EKzT
VyR33bQSYBhIsgxo8AAOuU0m3wPlBSwPIdtVV1WHvN9EUDx6x1U6tL/+qEs=
```

Example 20-24 *Verifying SSH Configuration on R5*

```
R2#show ip ssh
SSH Enabled - version 2.0
Authentication timeout: 120 secs; Authentication retries: 3
Minimum expected Diffie Hellman key size : 1024 bits
IOS Keys in SECSH format(ssh-rsa, base64 encoded):
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQAgQCtMAHQGeZaB/uWXiqF17KOWL+LvjsGOSJOCLFEkg7X
carueOLHbfSxhADkThmwFOksN9Sq9jFbd5YVpiRoP4nM8He/yRJszNDmCQAbV47IjhTYVISoznsRPh0P
/rxN/bf5ZsEdk4LVdA1nGnBjLsWTPtMO64PGOf/eVllrCMVYcw==
```

You access your documentation, and it states that all SSHv2 sessions should use a key of 1024. Therefore, on R4 you issue the **crypto key generate rsa modulus 1024** command to generate new cryptographic keys. As you can see in Example 20-25, the old keys are replaced with the new ones, and SSH 1.99 is enabled, which supports v1 and v2. As a result, you issue the **ip ssh version 2** command to enable just SSHv2.

Example 20-25 *Creating a Local Cryptographic Key*

```
R4#config t
Enter configuration commands, one per line. End with CNTL/Z.
R4(config)#crypto key generate rsa modulus 1024
% You already have RSA keys defined named R4.TSHOOT.local.
% They will be replaced.

% The key modulus size is 1024 bits
% Generating 1024 bit RSA keys, keys will be non-exportable...
[OK] (elapsed time was 2 seconds)

R4(config)#
%SSH-5-DISABLED: SSH 1.5 has been disabled
R4(config)#
%SSH-5-ENABLED: SSH 1.99 has been enabled
R4(config)#ip ssh version 2
```

You examine the output of **show ip ssh** on R4 again, as shown in Example 20-26. It shows that SSHv2 is now enabled; and if you compare the new SSH RSA key with the old one, it is much larger.

Example 20-26 *Verifying That SSHv2 Is Enabled*

```

R4#show ip ssh
SSH Enabled - version 2.0
Authentication timeout: 120 secs; Authentication retries: 3
Minimum expected Diffie Hellman key size : 1024 bits
IOS Keys in SECSH format (ssh-rsa, base64 encoded):
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCT6oQo7Ge64ky61+BPOJHOQwnaiUeJCPSbuDSjt610
DB6lRa0nhCjEMRG2W1OznJNtV5kHBdL7E/880ZOoQcSe3DEyh9TD88/CZI/Tr80OrLJYaN+5Y/7ZaZkp
5AUZCBVibtbkuc/z8FokE417607dI1KgP7VsJogKIur8FkciNQ==

```

Back on SW2, you try to establish an SSHv2 session to R4, and it is successful, as shown in Example 20-27.

Example 20-27 *Verify That the Issue Is Solved*

```

SW2#ssh -v 2 -l TSHOOT 192.0.2.1
Password:
R4>

```


Exam Preparation Tasks

As mentioned in the section “How to Use This Book” in the Introduction, you have a couple of choices for exam preparation: the exercises here; Chapter 22, “Final Preparation;” and the exam simulation questions on the CD-ROM.

Review All Key Topics

Review the most important topics in this chapter, noted with the Key Topic icon in the outer margin of the page. Table 20-2 lists a reference of these key topics and the page numbers on which each is found.



Table 20-2 *Key Topics for Chapter 20*

Key Topic Element	Description	Page Number
List	Describes the items you should consider when troubleshooting issues related to console port access	854
List	Outlines the items you should consider when troubleshooting issues related to Telnet	855
List	Outlines the items you should consider when troubleshooting issues related to SSH	857
Paragraph	Reviews the different types of password encryption levels	858
List	Outlines the items you should consider when troubleshooting issues related to Cisco IOS AAA authentication	859

Define Key Terms

Define the following key terms from this chapter and check your answers in the glossary:

login, login local, AAA, method list, rollover cable, Telnet, SSH, line, console, port 23, port 22, level 4 encryption, level 5 encryption, level 7 encryption, RADIUS, TACACS+

Command Reference to Check Your Memory

This section includes the most important **show** and **debug** commands covered in this chapter. It might not be necessary to memorize the complete syntax of every command, but you should be able to remember the basic keywords that are needed.

To test your memory of the commands, cover the right side of Table 20-3 with a piece of paper, read the description on the left side, and then see how much of the command you can remember.

The 300-135 TSHOOT exam focuses on practical, hands-on skills that are used by a networking professional. Therefore, you should be able to identify the commands needed to successfully troubleshoot the topics and concepts covered in this chapter.

Table 20-3 *show and debug Commands*

Task	Command Syntax
Displays the ingress and egress allowed transport protocols on vty line	<code>show line vty <i>line_number</i> include Allowed</code>
Displays only the ingress allowed transport protocols on a vty line	<code>show line vty <i>line_number</i> include Allowed input transports</code>
Displays the vty line configuration in the running configuration	<code>show run section line vty</code>
Displays the lines that are currently being used for management connectivity	<code>show users</code>
Displays whether SSH is enabled or disabled, the version of SSH enabled, and the SSH RSA key	<code>show ip ssh</code>
Displays the SSHv1 and SSHv2 connections to the local device	<code>show ssh</code>
Displays the configuration of the local usernames and passwords on the device, the AAA commands that have been configured, and the vty line configuration (great command for verifying AAA configuration issues)	<code>show run section usernameaaaalline vty</code>
Displays the authentication process in real time	<code>debug aaa authentication</code>
Displays the RADIUS authentication process in real time	<code>debug radius authentication</code>
Displays the local authentication process in real time	<code>debug aaa protocol local</code>



This chapter covers the following topics:

In each Trouble Ticket you are presented with a collection of show and debug commands output and challenged to resolve a series of misconfigurations. Suggested solutions are also provided.

- **Trouble Ticket 1:** This section presents you with a trouble ticket addressing a network experiencing STP issues.
- **Trouble Ticket 2:** This section presents you with a trouble ticket addressing a network experiencing HSRP issues.
- **Trouble Ticket 3:** This section presents you with a trouble ticket addressing a network experiencing EIGRP issues.
- **Trouble Ticket 4:** This section presents you with a trouble ticket addressing a network experiencing OSPF issues.
- **Trouble Ticket 5:** This section presents you with a trouble ticket addressing a network experiencing redistribution issues.
- **Trouble Ticket 6:** This section presents you with a trouble ticket addressing a network experiencing BGP issues.
- **Trouble Ticket 7:** This section presents you with a trouble ticket addressing a network experiencing management access issues.
- **Trouble Ticket 8:** This section presents you with a trouble ticket addressing a network experiencing NAT issues.
- **Trouble Ticket 9:** This section presents you with a trouble ticket addressing a network experiencing OSPFv3 issues.
- **Trouble Ticket 10:** This section presents you with a trouble ticket addressing a network experiencing RIPng issues.

Additional Trouble Tickets

Troubleshooting routed and switched networks is an art. The more time you spend troubleshooting, the better you will become. However, many of us do not have the opportunity to troubleshoot on a regular basis or experience many of the issues that may arise in routed and switched networks. Therefore, the more issues you can see samples of, the better.

This chapter is dedicated to showing you additional trouble tickets and the various approaches that you can take to solve the problems that are presented. Always remember that the right way to troubleshoot is the way that solves the problem for you. You and I and the person next to you will all have different methods and approaches to troubleshooting. What works for one might not work for the other. Someone with years of experience will have a vast knowledge base in their head that they can call upon for help while the novice will have to do more research or ask for assistance at times. However, no matter what, we all have the same goal. Solve the problem! Let's see how the following issues presented in this chapter could be solved.

Introduction

All trouble tickets begin with a problem report and a network topology diagram. Some of the trouble tickets provide you with baseline data, and all the trouble tickets offer output from appropriate verification commands (for example, **show** or **debug** commands) that you can examine.

After you hypothesize the underlying cause of the network issue and formulate a solution, you can check the *suggested solution* comments to confirm your hypothesis. Realize, however, that some trouble tickets might be resolvable by more than one method. Therefore, your solution might be different from the suggested solution.

Trouble Ticket 1

You receive the following trouble ticket:

Users on network 192.168.1.0/24 are experiencing latency or no connectivity when attempting to reach network 10.1.2.0/24. It appears to be STP related.

This trouble ticket references the topology shown in Figure 21-1.

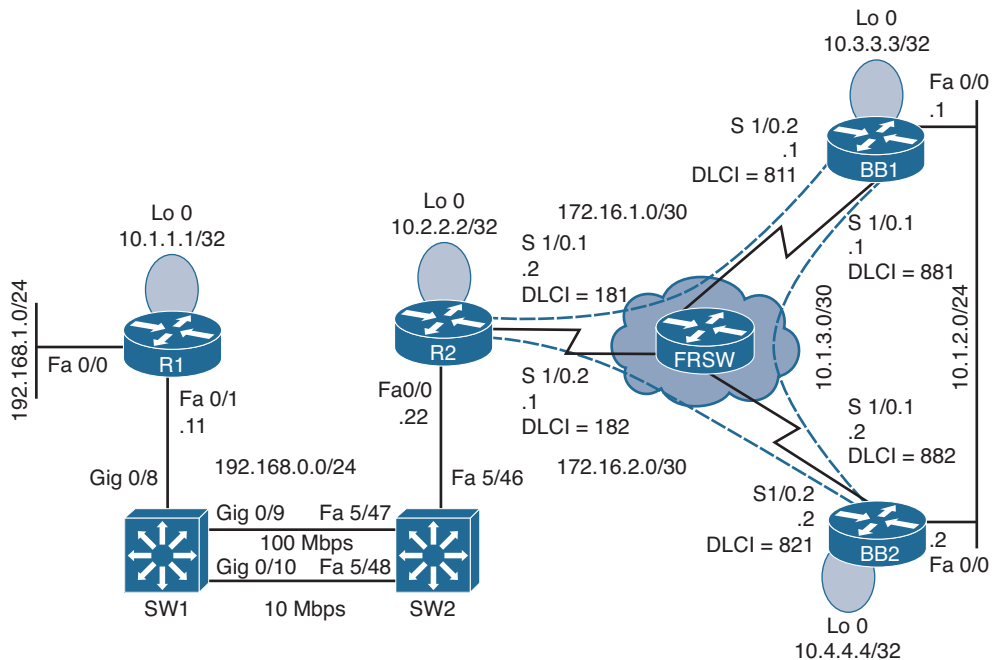


Figure 21-1 *Topology for Trouble Ticket 1*

As you follow the path of the traffic from network 192.168.1.0/24 to 10.1.2.0/24, you notice high port utilization levels on switches SW1 and SW2. Therefore, you decide to investigate these switches further.

You have previously issued **show** commands on these switches as part of your baseline collection process. A selection of the **show** command output is presented in Examples 21-1 and 21-2.

Example 21-1 *Baseline show Output from Switch SW1*

```
SW1#show spanning-tree vlan 1

VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority    32768
            Address     0009.122e.4181
            Cost       19
```

```

Port          9 (GigabitEthernet0/9)
Hello Time    2 sec  Max Age 20 sec  Forward Delay 15 sec
Bridge ID Priority 32769 (priority 32768 sys-id-ext 1)
Address       000d.28e4.7c80
Hello Time    2 sec  Max Age 20 sec  Forward Delay 15 sec
Aging Time    300

```

Interface	Role	Sts Cost	Prio.Nbr	Type
Gi0/8	Desg	FWD 19	128.8	P2p
Gi0/9	Root	FWD 19	128.9	P2p
Gi0/10	Altn	BLK 100	128.10	Shr

SW1#show spanning-tree summary

Switch is in pvst mode

Root bridge for: none

Extended system ID is enabled

Portfast Default is disabled

PortFast BPDU Guard Default is disabled

Portfast BPDU Filter Default is disabled

Loopguard Default is disabled

EtherChannel misconfig guard is enabled

UplinkFast is disabled

BackboneFast is disabled

Configured Pathcost method used is short

Name	Blocking	Listening	Learning	Forwarding	STP Active
VLAN0001	1	0	0	2	3
1 vlan	1	0	0	2	3

SW1#show spanning-tree interface gig 0/10 detail

Port 10 (GigabitEthernet0/10) of VLAN0001 is alternate blocking

Port path cost 100, Port priority 128, Port Identifier 128.10.

Designated root has priority 32768, address 0009.122e.4181

Designated bridge has priority 32768, address 0009.122e.4181

Designated port id is 128.304, designated path cost 0

Timers: message age 1, forward delay 0, hold 0

Number of transitions to forwarding state: 0

Link type is shared by default

BPDU: sent 1, received 276

Example 21-2 *Baseline show Output from Switch SW2*

```
SW2#show spanning-tree vlan 1

VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority    32768
             Address     0009.122e.4181
             This bridge is the root
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32768
             Address     0009.122e.4181
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  300

Interface Role Sts Cost Prio.Nbr Type
-----
Fa5/46    Desg FWD 19   128.302 Shr
Fa5/47    Desg FWD 19   128.303 P2p
Fa5/48    Desg FWD 100 128.304 Shr
```

When you connect to the console of switch SW1, you receive the console messages displayed in Example 21-3.

Example 21-3 *Console Messages on Switch SW1*

```
SW1#
00:15:45: %SW_MATM-4-MACFLAP_NOTIF: Host 0009.b7fa.d1e1 in vlan 1 is flapping
        between port Gi0/8 and port Gi0/9
SW1#
00:16:35: %SW_MATM-4-MACFLAP_NOTIF: Host 0009.b7fa.d1e1 in vlan 1 is flapping
        between port Gi0/8 and port Gi0/9
SW1#
00:16:37: %SW_MATM-4-MACFLAP_NOTIF: Host c001.0e8c.0000 in vlan 1 is flapping
        between port Gi0/9 and port Gi0/10
SW1#
00:16:41: %SW_MATM-4-MACFLAP_NOTIF: Host 0009.b7fa.d1e1 in vlan 1 is flapping
        between port Gi0/8 and port Gi0/9
```

You also issue the **show spanning-tree vlan 1** command on switches SW1 and SW2, as shown in Examples 21-4 and 21-5.

Example 21-4 *show spanning-tree vlan 1 Command Output on Switch SW1*

```
SW1#show spanning-tree vlan 1

Spanning tree instance(s) for vlan 1 does not exist.
```

Example 21-5 *show spanning-tree vlan 1 Command Output on Switch SW2*

```
SW2#show spanning-tree vlan 1

Spanning tree instance(s) for vlan 1 does not exist.
```

Take a moment to look through the baseline information, the topology, and the **show** command output. Then hypothesize the underlying cause for the connectivity issue reported in the trouble ticket. Finally, on a separate sheet of paper, write out a proposed action plan for resolving the reported issue.

Suggested Solution

The `%SW_MATM-4-MACFLAP_NOTIF` console message appearing on switch SW1 indicates that the MAC address in the MAC address table of switch SW1 is flapping between a couple of ports. This is a MAC address table corruption issue that is usually caused by STP not functioning correctly.

This suspicion is confirmed from the output in the **show spanning-tree vlan 1** command, issued on switches SW1 and SW2, which indicates that there is no STP instance for VLAN 1 on either switch. Therefore, as a solution, STP should be enabled for VLAN 1 on both switches, which is depicted in Examples 21-6 and 21-7.

Example 21-6 *Enabling STP for VLAN 1 on Switch SW1*

```
SW1#conf term
Enter configuration commands, one per line. End with CNTL/Z.
SW1(config)#spanning-tree vlan 1
SW1(config)#end
```

Example 21-7 *Enabling STP for VLAN 1 on Switch SW2*

```
SW2#conf term
Enter configuration commands, one per line. End with CNTL/Z.
SW2(config)#spanning-tree vlan 1
SW2(config)#end
```

After giving STP sufficient time to converge, after enabling STP for VLAN 1, the **show spanning-tree vlan 1** command is once again issued on switches SW1 and SW2, as illustrated in Examples 21-8 and 21-9. The output in these examples confirms that STP is now functioning correctly.

Example 21-8 *Checking the STP Status for VLAN 1 on Switch SW1*

```
SW1#show spanning-tree vlan 1
...OUTPUT OMITTED...
Interface Role Sts Cost Prio.Nbr Type
-----
Gi0/8      Desg FWD 19   128.8   P2p
Gi0/9      Root FWD 19   128.9   P2p
Gi0/10     Altn BLK 100  128.10  Shr
```

Example 21-9 *Checking the STP Status for VLAN 1 on Switch SW2*

```
SW2#show spanning-tree vlan 1
...OUTPUT OMITTED...
Interface Role Sts Cost Prio.Nbr Type
-----
Fa5/46     Desg FWD 19   128.302 Shr
Fa5/47     Desg FWD 19   128.303 P2p
Fa5/48     Desg FWD 100  128.304 Shr
```

Trouble Ticket 2

You receive the following trouble ticket:

A new network technician configured HSRP on routers BB1 and BB2, where BB1 is the active router. The configuration was initially working; however, now BB2 is the active router even though BB1 is operational.

This trouble ticket references the topology shown in Figure 21-2.

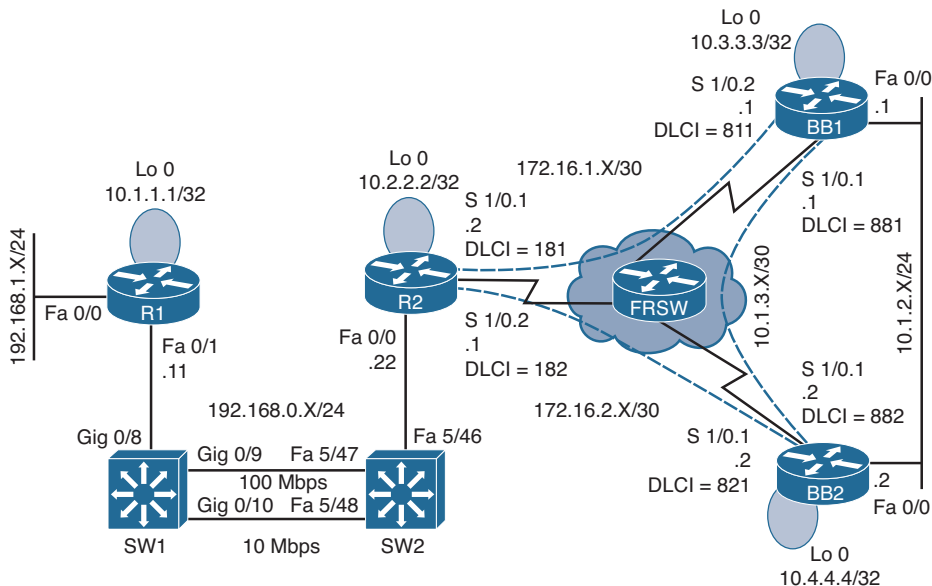


Figure 21-2 *Trouble Ticket 2 Topology*

As you investigate this issue, you examine baseline data collected after Hot Standby Router Protocol (HSRP) was initially configured. Examples 21-10 and 21-11 provide **show** and **debug** commands output collected when HSRP was working properly. Notice that router BB1 was acting as the active HSRP router, whereas router BB2 was acting as the standby HSRP router.

Example 21-10 Baseline Output for Router BB1

```
BB1#show standby brief
                P indicates configured to preempt.
                |
Interface  Grp Prio P State   Active        Standby        Virtual IP
Fa0/0      1   150  Active local     10.1.2.2       10.1.2.3

BB1#debug standby
HSRP debugging is on
*Mar  1 01:14:21.487: HSRP: Fa0/0 Grp 1 Hello in 10.1.2.2 Standby pri 100 vIP
10.1.2.3
*Mar  1 01:14:23.371: HSRP: Fa0/0 Grp 1 Hello out 10.1.2.1 Active pri 150 vIP
10.1.2.3

BB1#u all
All possible debugging has been turned off

BB1#show standby fa 0/0 1
FastEthernet0/0 - Group 1
  State is Active
    10 state changes, last state change 00:12:40
  Virtual IP address is 10.1.2.3
  Active virtual MAC address is 0000.0c07.ac01
    Local virtual MAC address is 0000.0c07.ac01 (v1 default)
  Hello time 3 sec, hold time 10 sec
    Next hello sent in 1.536 secs
  Preemption disabled
  Active router is local
  Standby router is 10.1.2.2, priority 100 (expires in 9.684 sec)
  Priority 150 (configured 150)
  IP redundancy name is "hsrp-Fa0/0-1" (default)

BB1#show run
...OUTPUT OMITTED...
hostname BB1
!
interface Loopback0
 ip address 10.3.3.3 255.255.255.255
!
interface FastEthernet0/0
 ip address 10.1.2.1 255.255.255.0
```

```
standby 1 ip 10.1.2.3
standby 1 priority 150
!
interface FastEthernet0/1
  no ip address
!
router ospf 1
  network 0.0.0.0 255.255.255.255 area 0
```

Example 21-11 *Baseline Output for Router BB2*

```
BB2#show standby brief
                P indicates configured to preempt.
                |
Interface   Grp Prio P State   Active       Standby       Virtual IP
Fa0/0       1   100   Standby 10.1.2.1    local        10.1.2.3
BB2#show run
...OUTPUT OMITTED...
hostname BB2
!
interface Loopback0
  ip address 10.4.4.4 255.255.255.255
!
interface FastEthernet0/0
  ip address 10.1.2.2 255.255.255.0
  standby 1 ip 10.1.2.3
!
interface FastEthernet0/1
  no ip address
!
router ospf 1
  network 0.0.0.0 255.255.255.255 area 0
```

As part of testing the initial configuration, a ping was sent to the virtual IP address of 10.1.2.3 from router R2 to confirm that HSRP was servicing requests for that IP address. Example 21-12 shows the output from the **ping** command.

Example 21-12 *Pinging the Virtual IP Address from Router R2*

```
R2#ping 10.1.2.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.2.3, timeout is 2 seconds:
!!!!!
```

As you begin to gather information about the reported problem, you reissue the **show standby brief** command on routers BB1 and BB2. As shown in Examples 21-13 and 21-14, router BB1 is administratively up with an HSRP priority of 150, whereas router BB2 is administratively up with a priority of 100.

Example 21-13 *Examining the HSRP State of Router BB1's Fast Ethernet 0/0 Interface*

```
BB1#show standby brief
                P indicates configured to preempt.

Interface  Grp Prio P State      Active      Standby      Virtual IP
Fa0/0      1   150 Standby  10.1.2.2    local        10.1.2.3
```

Example 21-14 *Examining the HSRP State of Router BB2's Fast Ethernet 0/0 Interface*

```
BB2#show standby brief
                P indicates configured to preempt.

Interface  Grp Prio P State      Active      Standby      Virtual IP
Fa0/0      1   100 Active   local        10.1.2.1    10.1.2.3
```

Take a moment to look through the baseline information, the topology, and the **show** command output. Then, hypothesize the underlying cause, explaining why router BB2 is currently the active HSRP router, even though router BB1 has a higher priority. Finally, on a separate sheet of paper, write out a proposed action plan for resolving the reported issue.

Suggested Solution

Upon examination of BB1's output, it becomes clear that the preempt feature is not enabled for the Fast Ethernet 0/0 interface on BB1. The absence of the preempt feature explains the reported symptom. Specifically, if BB1 had at one point been the active HSRP router for HSRP group 1, and either router BB1 or its Fast Ethernet 0/0 interface became unavailable, BB2 would have become the active router. Then, if BB1 or its Fast Ethernet 0/0 interface once again became available, BB1 would assume a standby HSRP role, because BB1's Fast Ethernet 0/0 interface was not configured for the preempt feature.

To resolve this configuration issue, the preempt feature is added to BB1's Fast Ethernet 0/0 interface, as shown in Example 21-15. After enabling the preempt feature, notice that router BB1 regains its active HSRP role.

Example 21-15 *Enabling the Preempt Feature on Router BB1's Fast Ethernet 0/0 Interface*

```
BB1#conf term
Enter configuration commands, one per line. End with CNTL/Z.
BB1(config)#int fa 0/0
BB1(config-if)#standby 1 preempt
BB1(config-if)#end
BB1#
*Mar  1 01:17:39.607: %HSRP-5-STATECHANGE: FastEthernet0/0 Grp 1 state Standby ->
Active
```

```
BB1#show standby brief
```

```
P indicates configured to preempt.
```

Interface	Grp	Prio	P	State	Active	Standby	Virtual IP
Fa0/0	1	150	P	Active	local	10.1.2.2	10.1.2.3

Trouble Ticket 3

You receive the following trouble ticket:

Enhanced Interior Gateway Routing Protocol (EIGRP) has just been configured as the routing protocol for the network. After configuring EIGRP on all routers and instructing all router interfaces to participate in EIGRP, router R2 does not appear to be load balancing across its subinterfaces to BB1 and BB2 when sending traffic to network 10.1.2.0/24.

This trouble ticket references the topology shown in Figure 21-3.

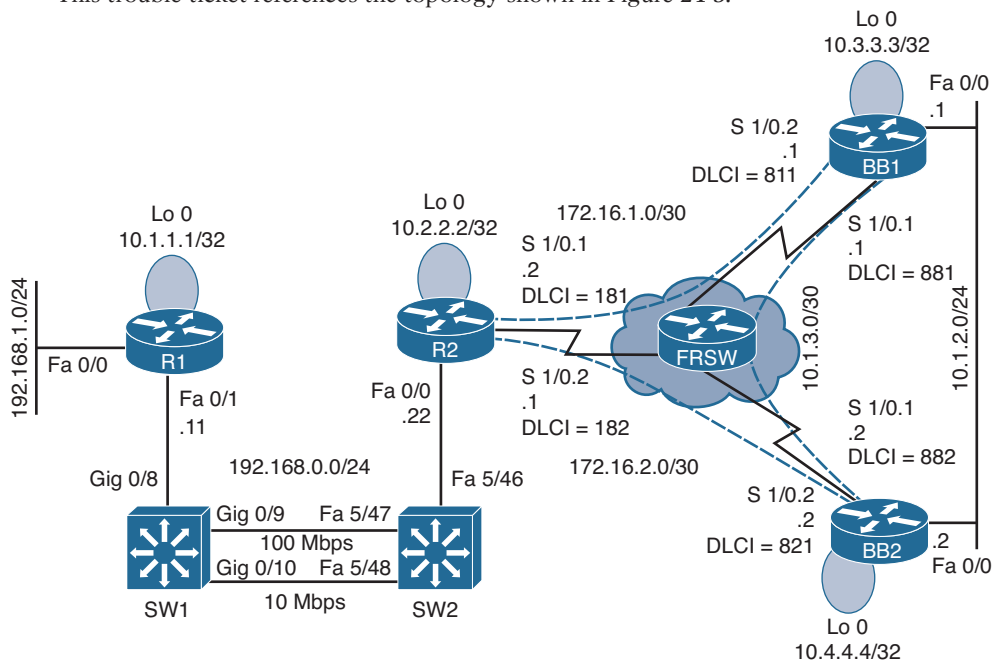


Figure 21-3 *Trouble Ticket 3 Topology*

As you investigate this issue, you examine baseline data collected after EIGRP was initially configured. Example 21-16 confirms that router R2's IP routing table contains only a single path to get to the backbone network of 10.1.2.0/24.

Example 21-16 Baseline IP Routing Table on Router R2

```
R2#show ip route
```

```
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
```

```
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
```

```
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
```

```

E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

172.16.0.0/30 is subnetted, 2 subnets
C    172.16.1.0 is directly connected, Serial1/0.1
C    172.16.2.0 is directly connected, Serial1/0.2
10.0.0.0/8 is variably subnetted, 6 subnets, 3 masks
C    10.2.2.2/32 is directly connected, Loopback0
D    10.1.3.0/30 [90/3072000] via 172.16.2.2, 00:00:34, Serial1/0.2
D    10.3.3.3/32 [90/2713600] via 172.16.2.2, 00:00:34, Serial1/0.2
D    10.1.2.0/24 [90/2585600] via 172.16.2.2, 00:00:34, Serial1/0.2
D    10.1.1.1/32 [90/409600] via 192.168.0.11, 00:00:46, FastEthernet0/0
D    10.4.4.4/32 [90/2688000] via 172.16.2.2, 00:00:34, Serial1/0.2
C    192.168.0.0/24 is directly connected, FastEthernet0/0
D    192.168.1.0/24 [90/284160] via 192.168.0.11, 00:18:33, FastEthernet0/0

```

You then view the EIGRP topology table on router R2 to see whether EIGRP has learned more than one route to reach the 10.1.2.0/24 network. The output, shown in Example 21-17, indicates that the EIGRP topology table knows two routes that could be used to reach the 10.1.2.0/24 network.

Example 21-17 EIGRP Topology Table on Router R2

```

R2#show ip eigrp topology
IP-EIGRP Topology Table for AS(1)/ID(10.2.2.2)

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 10.1.3.0/30, 1 successors, FD is 3072000
    via 172.16.2.2 (3072000/2169856), Serial1/0.2
    via 172.16.1.1 (4437248/2169856), Serial1/0.1
P 10.2.2.2/32, 1 successors, FD is 128256
    via Connected, Loopback0
P 10.1.2.0/24, 1 successors, FD is 2585600
    via 172.16.2.2 (2585600/281600), Serial1/0.2
    via 172.16.1.1 (3950848/281600), Serial1/0.1
P 10.3.3.3/32, 1 successors, FD is 2713600
    via 172.16.2.2 (2713600/409600), Serial1/0.2
    via 172.16.1.1 (4053248/128256), Serial1/0.1
P 10.1.1.1/32, 1 successors, FD is 409600
    via 192.168.0.11 (409600/128256), FastEthernet0/0
P 10.4.4.4/32, 1 successors, FD is 2688000

```

```

        via 172.16.2.2 (2688000/128256), Serial1/0.2
        via 172.16.1.1 (4078848/409600), Serial1/0.1
P 192.168.0.0/24, 1 successors, FD is 281600
    via Connected, FastEthernet0/0
P 192.168.1.0/24, 1 successors, FD is 284160
    via 192.168.0.11 (284160/28160), FastEthernet0/0
P 172.16.1.0/30, 1 successors, FD is 3925248
    via Connected, Serial1/0.1
P 172.16.2.0/30, 1 successors, FD is 2560000
    via Connected, Serial1/0.2

```

Finally, you examine the EIGRP configuration on router R1, as presented in Example 21-18.

Example 21-18 EIGRP Configuration on Router R2

```

R2#show run | begin router
router eigrp 1
  network 10.2.2.2 0.0.0.0
  network 172.16.1.0 0.0.0.3
  network 172.16.2.0 0.0.0.3
  network 192.168.0.0
  auto-summary

```

Take a moment to look through the **show** command output and the topology. Then, hypothesize the underlying cause, explaining why router R2's IP routing table only shows one route to network 10.1.2.0/24, even though the EIGRP topology table knows of two routes to that network. Finally, on a separate sheet of paper, write out a proposed action plan for resolving the reported issue.

Suggested Solution

Upon examination of router R2's EIGRP topology table (as previously shown in Example 21-17), it becomes clear that the reason router R2 is only injecting one of the 10.1.2.0/24 routes into the IP routing table is that the feasible distances of the two routes are different. By default, EIGRP load balances over routes with equal metrics (that is, equal feasible distances); however, the two routes present in the EIGRP topology table have different metrics.

Examine the two metrics (that is, 2585600 and 3950848), and notice that the metrics differ by less than a factor of 2. Specifically, if you took the smallest metric of 2585600 and multiplied it by 2, the result would be 5171200, which is greater than the largest metric of 3950848.

Because the metrics for the two routes vary by less than a factor of 2, EIGRP's variance feature could be configured to specify a variance of 2, as shown in Example 21-19. Specifically, this configuration tells EIGRP on router R2 to not only inject the best EIGRP route into the IP routing table, but rather inject the route with the best metric in addition

to any route whose metric is within a factor of two of the best metric (that is, in the range 2585600 to 5171200). This allows the route with a metric of 3950848 to also be injected into the IP routing table.

Example 21-19 *Enabling the Variance Feature on Router R2*

```
R2#conf term
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#router eigrp 1
R2(config-router)#variance 2
```

To confirm that router R2 can now load balance across routers BB1 and BB2 to reach the 10.1.2.0/24 network, examine the output of the **show ip route** command shown in Example 21-20. This output confirms that router R2 can now load balance over two unequal-cost paths to reach the 10.1.2.0/24 network.

Example 21-20 *Examining Router R2's IP Routing Table After Enabling the Variance Feature*

```
R2#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

172.16.0.0/30 is subnetted, 2 subnets
C      172.16.1.0 is directly connected, Serial1/0.1
C      172.16.2.0 is directly connected, Serial1/0.2
10.0.0.0/8 is variably subnetted, 6 subnets, 3 masks
C      10.2.2.2/32 is directly connected, Loopback0
D      10.1.3.0/30 [90/3072000] via 172.16.2.2, 00:00:03, Serial1/0.2
          [90/4437248] via 172.16.1.1, 00:00:03, Serial1/0.1
D      10.3.3.3/32 [90/2713600] via 172.16.2.2, 00:00:03, Serial1/0.2
          [90/4053248] via 172.16.1.1, 00:00:03, Serial1/0.1
D      10.1.2.0/24 [90/2585600] via 172.16.2.2, 00:00:03, Serial1/0.2
          [90/3950848] via 172.16.1.1, 00:00:03, Serial1/0.1
D      10.1.1.1/32 [90/409600] via 192.168.0.11, 00:00:03, FastEthernet0/0
D      10.4.4.4/32 [90/2688000] via 172.16.2.2, 00:00:03, Serial1/0.2
          [90/4078848] via 172.16.1.1, 00:00:03, Serial1/0.1
C      192.168.0.0/24 is directly connected, FastEthernet0/0
D      192.168.1.0/24 [90/284160] via 192.168.0.11, 00:00:04, FastEthernet0/0
```


Trouble Ticket 4

You receive the following trouble ticket:

For vendor interoperability reasons, a company changed its routing protocol from EIGRP to OSPF. The network was divided into areas, and all interfaces were instructed to participate in OSPF. The configuration was initially working. However, now none of the routers have full reachability to all the subnets.

This trouble ticket references the topology shown in Figure 21-4.

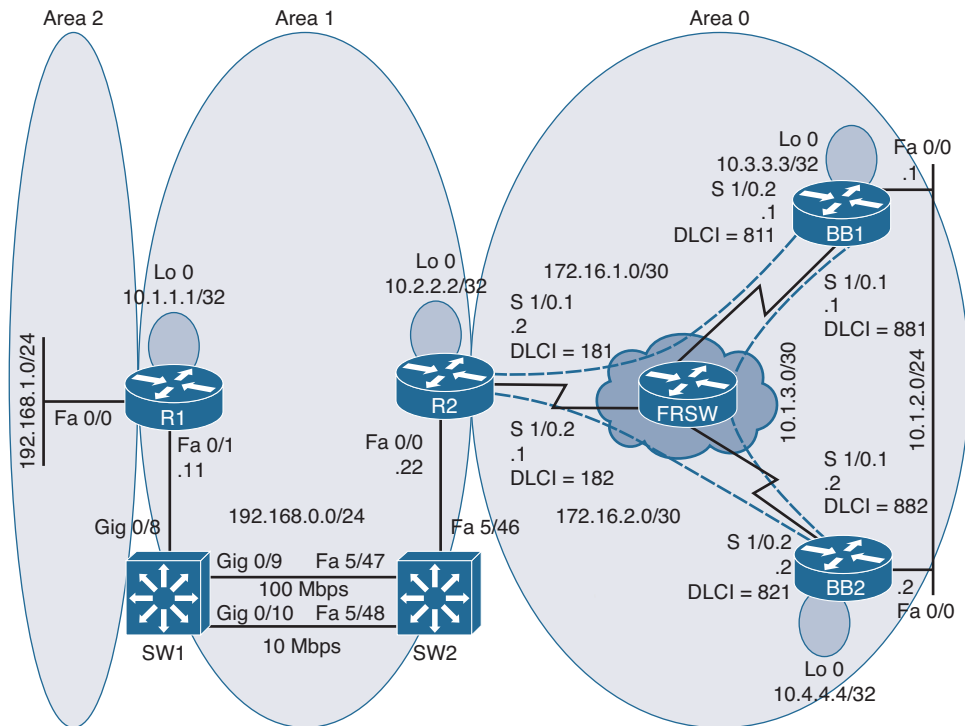


Figure 21-4 *Trouble Ticket 4 Topology*

As you investigate this issue, you examine baseline data collected after Open Shortest Path First (OSPF) was initially configured. Example 21-21 shows baseline data collected from router R1, when the network was fully operational. Notice that router R1 is configured with a virtual link because it does not physically touch area 0.

Example 21-21 *Baseline Configuration Data from Router R1*

```
R1#show run | begin router
router ospf 1
  area 1 virtual-link 10.2.2.2
  network 10.1.1.1 0.0.0.0 area 1
  network 192.168.0.0 0.0.0.255 area 1
  network 192.168.1.0 0.0.0.255 area 2
```

```
R1#show ip ospf neighbor
```

```
Neighbor ID Pri State Dead Time Address Interface
10.2.2.2 0 FULL/- - 192.168.0.22 OSPF_VL2
10.2.2.2 1 FULL/DR 00:00:38 192.168.0.22 FastEthernet0/1
```

```
R1#show ip route
```

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

ia - IS-IS inter area, * - candidate default, U - per-user static route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

```

    172.16.0.0/30 is subnetted, 2 subnets
O       172.16.1.0 [110/134] via 192.168.0.22, 01:34:44, FastEthernet0/1
O       172.16.2.0 [110/81] via 192.168.0.22, 01:34:44, FastEthernet0/1
    10.0.0.0/8 is variably subnetted, 6 subnets, 3 masks
O       10.2.2.2/32 [110/2] via 192.168.0.22, 02:24:31, FastEthernet0/1
O       10.1.3.0/30 [110/145] via 192.168.0.22, 01:34:44, FastEthernet0/1
O       10.3.3.3/32 [110/92] via 192.168.0.22, 01:34:44, FastEthernet0/1
O       10.1.2.0/24 [110/91] via 192.168.0.22, 01:34:45, FastEthernet0/1
C       10.1.1.1/32 is directly connected, Loopback0
O       10.4.4.4/32 [110/82] via 192.168.0.22, 01:34:45, FastEthernet0/1
C       192.168.0.0/24 is directly connected, FastEthernet0/1
C       192.168.1.0/24 is directly connected, FastEthernet0/0
```

```
R1#show ip ospf
```

Routing Process "ospf 1" with ID 10.1.1.1

Supports only single TOS(TOS0) routes

Supports opaque LSA

Supports Link-local Signaling (LLS)

Supports area transit capability It is an area border router

Initial SPF schedule delay 5000 msecs

Minimum hold time between two consecutive SPF's 10000 msecs

Maximum wait time between two consecutive SPF's 10000 msecs

Incremental-SPF disabled

Minimum LSA interval 5 secs

Minimum LSA arrival 1000 msecs

LSA group pacing timer 240 secs

Interface flood pacing timer 33 msecs

Retransmission pacing timer 66 msecs

Number of external LSA 0. Checksum Sum 0x000000

Number of opaque AS LSA 0. Checksum Sum 0x000000

```

Number of DCbitless external and opaque AS LSA 0
Number of DoNotAge external and opaque AS LSA 0
Number of areas in this router is 3. 3 normal 0 stub 0 nssa
Number of areas transit capable is 1
External flood list length 0
Area BACKBONE(0)
  Number of interfaces in this area is 1
  Area has no authentication
  SPF algorithm last executed 01:35:17.308 ago
  SPF algorithm executed 9 times
  Area ranges are
  Number of LSA 12. Checksum Sum 0x063B08
  Number of opaque link LSA 0. Checksum Sum 0x000000
  Number of DCbitless LSA 0
  Number of indication LSA 0
  Number of DoNotAge LSA 7
  Flood list length 0
Area 1
  Number of interfaces in this area is 2 (1 loopback)
  This area has transit capability: Virtual Link Endpoint
  Area has no authentication
  SPF algorithm last executed 02:25:04.377 ago
  SPF algorithm executed 22 times
  Area ranges are
  Number of LSA 10. Checksum Sum 0x059726
  Number of opaque link LSA 0. Checksum Sum 0x000000
  Number of DCbitless LSA 0
  Number of indication LSA 0
  Number of DoNotAge LSA 0
  Flood list length 0
Area 2
  Number of interfaces in this area is 1
  Number of indication LSA 0
  Number of DoNotAge LSA 0
  Flood list length 0
  Area has no authentication
  SPF algorithm last executed 02:25:15.880 ago
  SPF algorithm executed 9 times
  Area ranges are
  Number of LSA 10. Checksum Sum 0x05F97B
  Number of opaque link LSA 0. Checksum Sum 0x000000
  Number of DCbitless LSA 0
R1#show ip ospf interface fa0/1
FastEthernet0/1 is up, line protocol is up
  Internet Address 192.168.0.11/24, Area 1
  Process ID 1, Router ID 10.1.1.1, Network Type BROADCAST, Cost: 1

```

```

Transmit Delay is 1 sec, State BDR, Priority 1
Designated Router (ID) 10.2.2.2, Interface address 192.168.0.22
Backup Designated router (ID) 10.1.1.1, Interface address 192.168.0.11
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
oob-resync timeout 40
  Hello due in 00:00:00
Supports Link-local Signaling (LLS)
Index 2/2, flood queue length 0
Next 0x0(0)/0x0(0)
Last flood scan length is 1, maximum is 1
Last flood scan time is 0 msec, maximum is 4 msec
Neighbor Count is 1, Adjacent neighbor count is 1
  Adjacent with neighbor 10.2.2.2 (Designated Router)
Suppress hello for 0 neighbor(s)

```

Example 21-22 shows baseline configuration data collected from router R2.

Example 21-22 Baseline Configuration Data from Router R2

```

R2#show run | begin router
router ospf 1
  area 1 virtual-link 10.1.1.1
  network 10.2.2.2 0.0.0.0 area 1
  network 172.16.1.0 0.0.0.3 area 0
  network 172.16.2.0 0.0.0.3 area 0
  network 192.168.0.0 0.0.0.255 area 1

R2#show ip ospf neighbor

Neighbor ID Pri State Dead Time Address Interface
10.4.4.4 0 FULL/- 00:00:34 172.16.2.2 Serial1/0.2
10.3.3.3 0 FULL/- 00:00:37 172.16.1.1 Serial1/0.1
10.1.1.1 0 FULL/- - 192.168.0.11 OSPF_VL0
10.1.1.1 1 FULL/BDR 00:00:39 192.168.0.11 FastEthernet0/0

R2#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route
Gateway of last resort is not set

172.16.0.0/30 is subnetted, 2 subnets
C 172.16.1.0 is directly connected, Serial1/0.1

```

```

C      172.16.2.0 is directly connected, Serial1/0.2
      10.0.0.0/8 is variably subnetted, 6 subnets, 3 masks
C      10.2.2.2/32 is directly connected, Loopback0
O      10.1.3.0/30 [110/144] via 172.16.2.2, 01:34:50, Serial1/0.2
O      10.3.3.3/32 [110/91] via 172.16.2.2, 01:34:50, Serial1/0.2
O      10.1.2.0/24 [110/90] via 172.16.2.2, 01:34:50, Serial1/0.2
O      10.1.1.1/32 [110/11] via 192.168.0.11, 02:24:36, FastEthernet0/0
O      10.4.4.4/32 [110/81] via 172.16.2.2, 01:34:50, Serial1/0.2
C      192.168.0.0/24 is directly connected, FastEthernet0/0
O IA 192.168.1.0/24 [110/11] via 192.168.0.11, 01:34:50, FastEthernet0/0

```

Example 21-23 shows baseline configuration data collected from router BB1.

Example 21-23 Baseline Configuration Data from Router BB1

```

BB1#show run | begin router
router ospf 1
  network 0.0.0.0 255.255.255.255 area 0

BB1#show ip ospf neighbor

Neighbor ID  Pri  State   Dead Time  Address    Interface
10.4.4.4      1  FULL/DR 00:00:38  10.1.2.2   FastEthernet0/0
10.2.2.2      0  FULL/-  00:00:39  172.16.1.2 Serial1/0.2
10.4.4.4      0  FULL/-  00:00:38  10.1.3.2   Serial1/0.1

BB1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

      172.16.0.0/30 is subnetted, 2 subnets
C      172.16.1.0 is directly connected, Serial1/0.2
O      172.16.2.0 [110/90] via 10.1.2.2, 01:35:01, FastEthernet0/0
      10.0.0.0/8 is variably subnetted, 6 subnets, 3 masks
O IA   10.2.2.2/32 [110/91] via 10.1.2.2, 01:35:01, FastEthernet0/0
C      10.1.3.0/30 is directly connected, Serial1/0.1
C      10.3.3.3/32 is directly connected, Loopback0
C      10.1.2.0/24 is directly connected, FastEthernet0/0
O IA   10.1.1.1/32 [110/101] via 10.1.2.2, 01:35:01, FastEthernet0/0
O      10.4.4.4/32 [110/11] via 10.1.2.2, 01:35:01, FastEthernet0/0
O IA   192.168.0.0/24 [110/100] via 10.1.2.2, 01:35:01, FastEthernet0/0
O IA   192.168.1.0/24 [110/101] via 10.1.2.2, 01:35:01, FastEthernet0/0

```

Example 21-24 shows baseline configuration data collected from router BB2.

Example 21-24 *Baseline Configuration Data from Router BB2*

```
BB2#show run | begin router
router ospf 1
  network 0.0.0.0 255.255.255.255 area 0

BB2#show ip ospf neighbor
Neighbor ID      Pri   State           Dead Time   Address        Interface
10.2.2.2          0    FULL/-         00:00:32    172.16.2.1     Serial1/0.2
10.3.3.3          0    FULL/-         00:00:39    10.1.3.1        Serial1/0.1
10.3.3.3          1    FULL/BDR       00:00:35    10.1.2.1        FastEthernet0/0

BB2#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

    172.16.0.0/30 is subnetted, 2 subnets
O IA 192.168.1.0/24 [110/101] via 10.1.2.2, 01:35:01, FastEthernet0/0
O     172.16.1.0 [110/143] via 10.1.2.1, 01:35:06, FastEthernet0/0
C     172.16.2.0 is directly connected, Serial1/0.2
    10.0.0.0/8 is variably subnetted, 6 subnets, 3 masks
O IA  10.2.2.2/32 [110/81] via 172.16.2.1, 01:35:06, Serial1/0.2
C     10.1.3.0/30 is directly connected, Serial1/0.1
O     10.3.3.3/32 [110/11] via 10.1.2.1, 01:35:06, FastEthernet0/0
C     10.1.2.0/24 is directly connected, FastEthernet0/0
O IA  10.1.1.1/32 [110/91] via 172.16.2.1, 01:35:06, Serial1/0.2
C     10.4.4.4/32 is directly connected, Loopback0
O IA 192.168.0.0/24 [110/90] via 172.16.2.1, 01:35:06, Serial1/0.2
O IA 192.168.1.0/24 [110/91] via 172.16.2.1, 01:35:06, Serial1/0.2
```

Now that you have seen the baseline data, the following examples present you with data collected after the trouble ticket was issued. Example 21-25 shows information collected from router R1. Notice that router R1's routing table can no longer see the Loopback 0 IP address of router BB2 (that is, 10.4.4.4/32). Also, notice that the virtual link between area 2 and area 0 is down.

Example 21-25 *Information Gathered from Router R1 After the Trouble Ticket Was Issued*

```

R1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

    172.16.0.0/30 is subnetted, 2 subnets
O IA   172.16.1.0 [110/134] via 192.168.0.22, 00:00:31, FastEthernet0/1
O IA   172.16.2.0 [110/81] via 192.168.0.22, 00:00:31, FastEthernet0/1
    10.0.0.0/8 is variably subnetted, 5 subnets, 3 masks
O      10.2.2.2/32 [110/2] via 192.168.0.22, 00:00:51, FastEthernet0/1
O IA   10.1.3.0/30 [110/198] via 192.168.0.22, 00:00:31, FastEthernet0/1
O IA   10.3.3.3/32 [110/135] via 192.168.0.22, 00:00:31, FastEthernet0/1
O IA   10.1.2.0/24 [110/144] via 192.168.0.22, 00:00:32, FastEthernet0/1
C      10.1.1.1/32 is directly connected, Loopback0
C      192.168.0.0/24 is directly connected, FastEthernet0/1
C      192.168.1.0/24 is directly connected, FastEthernet0/0
R1#show run | begin router
router ospf 1
  log-adjacency-changes
  area 2 virtual-link 10.2.2.2
  network 10.1.1.1 0.0.0.0 area 1
  network 192.168.0.0 0.0.0.255 area 1
  network 192.168.1.0 0.0.0.255 area 2
R1#show ip ospf virtual-links
Virtual Link OSPF_VL4 to router 10.2.2.2 is down
Run as demand circuit
  DoNotAge LSA allowed.
  Transit area 2, Cost of using 65535
  Transmit Delay is 1 sec, State DOWN,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5

```

Example 21-26 shows the IP routing table on router R2 after the trouble ticket was issued. Notice that the routing table of router R2 can no longer see the Loopback 0 IP address of router BB2 (that is, 10.4.4.4/32). Also, notice that network 192.168.1.0/24, connected to router R1's Fast Ethernet 0/0 interface, is not present in router R2's IP routing table.

Example 21-26 *Router R2's IP Routing Table After the Trouble Ticket Was Issued*

```

R2#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

    172.16.0.0/30 is subnetted, 2 subnets
C       172.16.1.0 is directly connected, Serial1/0.1
C       172.16.2.0 is directly connected, Serial1/0.2
    10.0.0.0/8 is variably subnetted, 5 subnets, 3 masks
C       10.2.2.2/32 is directly connected, Loopback0
O       10.1.3.0/30 [110/197] via 172.16.1.1, 00:00:53, Serial1/0.1
O       10.3.3.3/32 [110/134] via 172.16.1.1, 00:00:53, Serial1/0.1
O       10.1.2.0/24 [110/143] via 172.16.1.1, 00:00:53, Serial1/0.1
O       10.1.1.1/32 [110/11] via 192.168.0.11, 00:00:53, FastEthernet0/0
C       192.168.0.0/24 is directly connected, FastEthernet0/0

```

Before moving forward to investigate the remainder of the network, do you already see an issue that needs to be resolved? The fact that router R2 cannot see network 192.168.1.0/24 off of router R1 is independent of any configuration on routers BB1 or BB2. So, take a few moments to review the information collected thus far, and hypothesize the issue that is preventing router R2 from seeing network 192.168.1.0/24. On a separate sheet of paper, write your solution to the issue you identified.

Issue 1: Suggested Solution

The virtual link configuration on router R1 was incorrect. Specifically, the transit area in the `area number virtual-link router_id` command was configured as area 2. However, the transit area should have been area 1. Example 21-27 shows the commands used to correct this misconfiguration.

Example 21-27 *Correcting the Virtual Link Configuration of R1*

```

R1#conf term
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#router ospf 1
R1(config-router)#no area 2 virtual-link 10.2.2.2
R1(config-router)#area 1 virtual-link 10.2.2.2

```

After you correct the virtual link configuration on router R1, network 192.168.1.0/24 is present in router R2's IP routing table, as illustrated in Example 21-28. Notice, however,

that the Loopback 0 IP address of router BB2 (that is, 10.4.4.4/32) is still not visible in router R2's IP routing table.

Example 21-28 *Router R2's IP Routing Table After Correcting the Virtual Link Configuration*

```
R2#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route
Gateway of last resort is not set
 172.16.0.0/30 is subnetted, 2 subnets
C      172.16.1.0 is directly connected, Serial1/0.1
C      172.16.2.0 is directly connected, Serial1/0.2
 10.0.0.0/8 is variably subnetted, 5 subnets, 3 masks
C      10.2.2.2/32 is directly connected, Loopback0
O      10.1.3.0/30 [110/197] via 172.16.1.1, 00:00:18, Serial1/0.1
O      10.3.3.3/32 [110/134] via 172.16.1.1, 00:00:18, Serial1/0.1
O      10.1.2.0/24 [110/143] via 172.16.1.1, 00:00:18, Serial1/0.1
O      10.1.1.1/32 [110/11] via 192.168.0.11, 00:00:18, FastEthernet0/0
C      192.168.0.0/24 is directly connected, FastEthernet0/0
O IA 192.168.1.0/24 [110/11] via 192.168.0.11, 00:00:18, FastEthernet0/0
```

With one issue now resolved, continue to collect information on router R2. Example 21-29 indicates that router R2 has not formed an adjacency with router BB2, which has an OSPF router ID of 10.4.4.4.

Example 21-29 *OSPF Neighbors of Router R2*

```
R2#show ip ospf neighbor
Neighbor ID  Pri  State       Dead Time   Address        Interface
10.3.3.3      0  FULL/-      00:00:37    172.16.1.1     Serial1/0.1
10.1.1.1      0  FULL/-      -           192.168.0.11   OSPF_VL1
10.1.1.1      1  FULL/DR     00:00:39    192.168.0.11   FastEthernet0/0

R2#show run | begin router
router ospf 2
 log-adjacency-changes
 area 1 virtual-link 10.1.1.1
 network 10.2.2.2 0.0.0.0 area 1
 network 172.16.1.0 0.0.0.3 area 0
 network 172.16.2.0 0.0.0.3 area 0
 network 192.168.0.0 0.0.0.255 area 1
```

Even though router R2 has not formed an adjacency with router BB2, Example 21-30 shows the output of a **ping** command, verifying that router R2 can reach router BB2.

Example 21-30 *Pinging Router BB2 from Router R2*

```

R2#ping 172.16.2.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.2.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 52/92/144 ms

```

The topology diagram indicates that router R2 connects with router BB2 via subinterface Serial 1/0.2. Therefore, the **show interface s1/0.2** command is issued on router R2. The output provided in Example 21-31 states that the subinterface is up and functional.

Example 21-31 *Serial 1/0.2 Subinterface of Router R2*

```

R2#show interface s1/0.2
Serial1/0.2 is up, line protocol is up
Hardware is M4T
Internet address is 172.16.2.1/30
MTU 1500 bytes, BW 1250 Kbit, DLY 20000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
Encapsulation FRAME-RELAY
Last clearing of "show interface" counters never

```

Example 21-32 confirms that router BB2 is adjacent at Layer 2 with router R2.

Example 21-32 *CDP Neighbors of Router R2*

```

R2#show cdp neighbor
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater

```

Device ID	Local Intrfce	Holdtime	Capability	Platform	Port ID
BB1	Ser 1/0.1	152	R S I	2691	Ser 1/0.2
BB2	Ser 1/0.2	143	R S I	2691	Ser 1/0.2
R1	Fas 0/0	144	R S I	2611XM	Fas 0/1

The output of Example 21-33 shows the OSPF status of router R2's Serial 1/0.2 subinterface.

Example 21-33 *OSPF Status of Router R2 on Subinterface Serial 1/0.2*

```

R2#show ip ospf interface s1/0.2
Serial1/0.2 is up, line protocol is up
Internet Address 172.16.2.1/30, Area 0
Process ID 1, Router ID 10.2.2.2, Network Type POINT_TO_POINT, Cost: 64
Transmit Delay is 1 sec, State POINT_TO_POINT,
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    oob-resync timeout 40
    Hello due in 00:00:09
Supports Link-local Signaling (LLS)

```

```

Index 3/4, flood queue length 0
Next 0x0(0)/0x0(0)
Last flood scan length is 1, maximum is 4
Last flood scan time is 0 msec, maximum is 4 msec
Neighbor Count is 0, Adjacent neighbor count is 0
Suppress hello for 0 neighbor(s)

```

Now that data has been collected for router R2, the troubleshooting focus moves to router BB1 in Example 21-34. Notice that BB1 also lacks a route to router BB2's Loopback 0 IP address of 10.4.4.4/32. Also, even though router BB1 has two direct connections to router BB2, router BB1 has not formed an OSPF adjacency with router BB2. Notice that router BB2 is router BB1's Cisco Discovery Protocol (CDP) neighbor, both on interface Fast Ethernet 0/0 and on subinterface Serial 1/0.1.

Example 21-34 Data Collected from Router BB1 After the Trouble Ticket

```

BB1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

 172.16.0.0/30 is subnetted, 2 subnets
C       172.16.1.0 is directly connected, Serial1/0.2
O       172.16.2.0 [110/213] via 172.16.1.2, 00:01:02, Serial1/0.2
 10.0.0.0/8 is variably subnetted, 5 subnets, 3 masks
O IA    10.2.2.2/32 [110/134] via 172.16.1.2, 00:01:02, Serial1/0.2
C       10.1.3.0/30 is directly connected, Serial1/0.1
C       10.3.3.3/32 is directly connected, Loopback0
C       10.1.2.0/24 is directly connected, FastEthernet0/0
O IA    10.1.1.1/32 [110/144] via 172.16.1.2, 00:01:02, Serial1/0.2
O IA    192.168.0.0/24 [110/143] via 172.16.1.2, 00:01:02, Serial1/0.2

BB1#show ip ospf neighbor

Neighbor ID  Pri  State   Dead Time   Address        Interface
10.2.2.2      0    FULL/-   00:00:30    172.16.1.2     Serial1/0.2

BB1#show run | begin router
router ospf 1
 log-adjacency-changes
 network 0.0.0.0 255.255.255.255 area 0

BB1#show cdp neigh
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater

```

```

Device ID  Local Intrfce  Holdtime  Capability  Platform  Port ID
BB2        Ser 1/0.1      148       R S I       2691      Ser 1/0.1
BB2        Fas 0/0         148       R S I       2691      Fas 0/0
R2         Ser 1/0.2         130       R S I       2691      Ser 1/0.1
BB1#show run

...OUTPUT OMITTED...
interface FastEthernet0/0
 ip address 10.1.2.1 255.255.255.0
 ip ospf network non-broadcast
 duplex auto
 speed auto
!
interface Serial1/0
 no ip address
 encapsulation frame-relay
!
interface Serial1/0.1 point-to-point
 ip address 10.1.3.1 255.255.255.252
 ip ospf hello-interval 60
 ip ospf dead-interval 200
 frame-relay interface-dlci 881
!
interface Serial1/0.2 point-to-point
 bandwidth 750
 ip address 172.16.1.1 255.255.255.252
 frame-relay interface-dlci 811
...OUTPUT OMITTED...

```

The data collection continues on router BB2. Example 21-35 provides output from several **show** commands. Notice that router BB2 has not learned networks via OSPF.

Example 21-35 Data Collected from Router BB2 After the Trouble Ticket

```

BB2#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route
Gateway of last resort is not set

    172.16.0.0/30 is subnetted, 1 subnets
C       172.16.2.0 is directly connected, Serial1/0.2
    10.0.0.0/8 is variably subnetted, 3 subnets, 3 masks
C       10.1.3.0/30 is directly connected, Serial1/0.1

```

```

C      10.1.2.0/24 is directly connected, FastEthernet0/0
C      10.4.4.4/32 is directly connected, Loopback0

BB2#show run | begin router
router ospf 1
  log-adjacency-changes
  network 0.0.0.0 255.255.255.255 area 0

BB2#show ip ospf interface s1/0.1
Serial1/0.1 is up, line protocol is up
  Internet Address 10.1.3.2/30, Area 0
  Process ID 1, Router ID 10.4.4.4, Network Type POINT_TO_POINT, Cost: 64
  Transmit Delay is 1 sec, State POINT_TO_POINT,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    oob-resync timeout 40
  Hello due in 00:00:09
  Supports Link-local Signaling (LLS)
  Index 2/2, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 3
  Last flood scan time is 0 msec, maximum is 4 msec
  Neighbor Count is 0, Adjacent neighbor count is 0
  Suppress hello for 0 neighbor(s)

BB2#show ip ospf interface s1/0.2
Serial1/0.2 is up, line protocol is up
  Internet Address 172.16.2.2/30, Area 0
  Process ID 1, Router ID 10.4.4.4, Network Type NON_BROADCAST, Cost: 80
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 10.4.4.4, Interface address 172.16.2.2
  No backup designated router on this network
  Timer intervals configured, Hello 30, Dead 120, Wait 120, Retransmit 5
    oob-resync timeout 120
  Hello due in 00:00:09
  Supports Link-local Signaling (LLS)
  Index 3/3, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 1
  Last flood scan time is 0 msec, maximum is 4 msec
  Neighbor Count is 0, Adjacent neighbor count is 0
  Suppress hello for 0 neighbor(s)
!
BB2#show run | begin interface
interface FastEthernet0/0
  ip address 10.1.2.2 255.255.255.0
!
```

```

interface Serial1/0
  no ip address
  encapsulation frame-relay
!
interface Serial1/0.1 point-to-point
  ip address 10.1.3.2 255.255.255.252
  frame-relay interface-dlci 882
!
interface Serial1/0.2 point-to-point
  bandwidth 1250
  ip address 172.16.2.2 255.255.255.252
  ip ospf network non-broadcast
  frame-relay interface-dlci 821
!
...OUTPUT OMITTED...

```

Based on the preceding **show** command output from routers R2, BB1, and BB2, hypothesize what you consider to be the issue or issues still impacting the network. Then, on a separate sheet of paper, write how you would solve the identified issue or issues.

Issue 2: Suggested Solution

Subinterface Serial 1/0.1 on router BB1 had non-default hello and dead timers, which did not match the timers at the far end of the Frame Relay link. Example 21-36 illustrates how these nondefault values were reset.

Example 21-36 *Correcting the Nondefault Timer Configuration of Router BB1*

```

BB1#conf term
Enter configuration commands, one per line. End with CNTL/Z.
BB1(config)#int s1/0.1
BB1(config-subif)#no ip ospf hello-interval 60
BB1(config-subif)#no ip ospf dead-interval 200

```

Issue 3: Suggested Solution

Interface Fast Ethernet 0/0 on router BB1 was configured with an incorrect OSPF network type of nonbroadcast. Example 21-37 demonstrates how this OSPF interface was reset to its default OSPF network type (that is, the broadcast OSPF network type).

Example 21-37 *Correcting the Incorrect OSPF Network Type Configuration of Router BB1*

```

BB1#conf term
Enter configuration commands, one per line. End with CNTL/Z.
BB1(config)#int fa 0/0
BB1(config-if)#no ip ospf network non-broadcast

```

Issue 4: Suggested Solution

Similar to the incorrect OSPF network type on router BB1's Fast Ethernet 0/0 interface, the Serial 1/0.2 subinterface on router BB2 was configured incorrectly. A point-to-point Frame Relay subinterface defaults to an OSPF network type of point-to-point; however, Serial 1/0.2 had been configured as an OSPF network type of nonbroadcast. Example 21-38 reviews how this nondefault OSPF network type configuration was removed.

Example 21-38 *Correcting Router BB2's Incorrect OSPF Network Type Configuration*

```
BB2#conf term
Enter configuration commands, one per line. End with CNTL/Z.
BB2(config)#int s1/0.2
BB2(config-subif)#no ip ospf network non-broadcast
```

After all the previous misconfigurations are corrected, all routers in the topology once again have full reachability throughout the network. Examples 21-39, 21-40, 21-41, and 21-42 show output from the **show ip route** and **show ip ospf neighbor** commands issued on all routers, confirming the full reachability of each router.

Example 21-39 *Confirming the Full Reachability of Router R1*

```
R1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

    172.16.0.0/30 is subnetted, 2 subnets
O       172.16.1.0 [110/134] via 192.168.0.22, 00:00:03, FastEthernet0/1
O       172.16.2.0 [110/81] via 192.168.0.22, 00:00:03, FastEthernet0/1
    10.0.0.0/8 is variably subnetted, 6 subnets, 3 masks
O       10.2.2.2/32 [110/2] via 192.168.0.22, 00:08:18, FastEthernet0/1
O       10.1.3.0/30 [110/145] via 192.168.0.22, 00:00:03, FastEthernet0/1
O       10.3.3.3/32 [110/92] via 192.168.0.22, 00:00:03, FastEthernet0/1
O       10.1.2.0/24 [110/91] via 192.168.0.22, 00:00:04, FastEthernet0/1
C       10.1.1.1/32 is directly connected, Loopback0
O       10.4.4.4/32 [110/82] via 192.168.0.22, 00:00:04, FastEthernet0/1
C       192.168.0.0/24 is directly connected, FastEthernet0/1
C       192.168.1.0/24 is directly connected, FastEthernet0/0

R1#show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
10.2.2.2	0	FULL/-	-	192.168.0.22	OSPF_VL5
10.2.2.2	1	FULL/BDR	00:00:34	192.168.0.22	FastEthernet0/1

Example 21-40 *Confirming the Full Reachability of Router R2*

```
R2#show ip route
```

```
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
```

```
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
```

```
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
```

```
       E1 - OSPF external type 1, E2 - OSPF external type 2
```

```
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
```

```
       ia - IS-IS inter area, * - candidate default, U - per-user static route
```

```
       o - ODR, P - periodic downloaded static route
```

```
Gateway of last resort is not set
```

```
172.16.0.0/30 is subnetted, 2 subnets
```

```
C      172.16.1.0 is directly connected, Serial1/0.1
```

```
C      172.16.2.0 is directly connected, Serial1/0.2
```

```
10.0.0.0/8 is variably subnetted, 6 subnets, 3 masks
```

```
C      10.2.2.2/32 is directly connected, Loopback0
```

```
O      10.1.3.0/30 [110/144] via 172.16.2.2, 00:00:15, Serial1/0.2
```

```
O      10.3.3.3/32 [110/91] via 172.16.2.2, 00:00:15, Serial1/0.2
```

```
O      10.1.2.0/24 [110/90] via 172.16.2.2, 00:00:15, Serial1/0.2
```

```
O      10.1.1.1/32 [110/11] via 192.168.0.11, 00:08:29, FastEthernet0/0
```

```
O      10.4.4.4/32 [110/81] via 172.16.2.2, 00:00:15, Serial1/0.2
```

```
C      192.168.0.0/24 is directly connected, FastEthernet0/0
```

```
O IA 192.168.1.0/24 [110/11] via 192.168.0.11, 00:00:15, FastEthernet0/0
```

```
R2#show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
10.4.4.4	0	FULL/ -	00:00:33	172.16.2.2	Serial1/0.2
10.3.3.3	0	FULL/ -	00:00:38	172.16.1.1	Serial1/0.1
10.1.1.1	0	FULL/ -	-	192.168.0.11	OSPF_VL1
10.1.1.1	1	FULL/DR	00:00:30	192.168.0.11	FastEthernet0/0

Example 21-41 *Confirming the Full Reachability of Router BB1*

```
BB1#show ip route
```

```
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
```

```
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
```

```
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
```

```
       E1 - OSPF external type 1, E2 - OSPF external type 2
```

```
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
```

```
       ia - IS-IS inter area, * - candidate default, U - per-user static route
```

```
       o - ODR, P - periodic downloaded static route
```


Gateway of last resort is not set

```

    172.16.0.0/30 is subnetted, 2 subnets
C       172.16.1.0 is directly connected, Serial1/0.2
O       172.16.2.0 [110/90] via 10.1.2.2, 00:00:29, FastEthernet0/0
    10.0.0.0/8 is variably subnetted, 6 subnets, 3 masks
O IA    10.2.2.2/32 [110/91] via 10.1.2.2, 00:00:29, FastEthernet0/0
C       10.1.3.0/30 is directly connected, Serial1/0.1
C       10.3.3.3/32 is directly connected, Loopback0
C       10.1.2.0/24 is directly connected, FastEthernet0/0
O IA    10.1.1.1/32 [110/101] via 10.1.2.2, 00:00:29, FastEthernet0/0
O       10.4.4.4/32 [110/11] via 10.1.2.2, 00:00:29, FastEthernet0/0
O IA    192.168.0.0/24 [110/100] via 10.1.2.2, 00:00:29, FastEthernet0/0
O IA    192.168.1.0/24 [110/101] via 10.1.2.2, 00:00:29, FastEthernet0/0
BB1#show ip ospf neighbor

```

Neighbor ID	Pri	State	Dead Time	Address	Interface
10.4.4.4	1	FULL/DR	00:00:34	10.1.2.2	FastEthernet0/
10.2.2.2	0	FULL/ -	00:00:39	172.16.1.2	Serial1/0.2
10.4.4.4	0	FULL/ -	00:00:33	10.1.3.2	Serial1/0.1

Example 21-42 Confirming the Full Reachability of Router BB2

BB2#show ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
 D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
 N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
 E1 - OSPF external type 1, E2 - OSPF external type 2
 i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
 ia - IS-IS inter area, * - candidate default, U - per-user static route
 o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

```

    172.16.0.0/30 is subnetted, 2 subnets
O       172.16.1.0 [110/143] via 10.1.2.1, 00:00:42, FastEthernet0/0
C       172.16.2.0 is directly connected, Serial1/0.2
    10.0.0.0/8 is variably subnetted, 6 subnets, 3 masks
O IA    10.2.2.2/32 [110/81] via 172.16.2.1, 00:00:42, Serial1/0.2
C       10.1.3.0/30 is directly connected, Serial1/0.1
O       10.3.3.3/32 [110/11] via 10.1.2.1, 00:00:42, FastEthernet0/0
C       10.1.2.0/24 is directly connected, FastEthernet0/0
O IA    10.1.1.1/32 [110/91] via 172.16.2.1, 00:00:42, Serial1/0.2
C       10.4.4.4/32 is directly connected, Loopback0
O IA    192.168.0.0/24 [110/90] via 172.16.2.1, 00:00:42, Serial1/0.2
O IA    192.168.1.0/24 [110/91] via 172.16.2.1, 00:00:42, Serial1/0.2

```


You begin your troubleshooting efforts by analyzing baseline information collected when the configuration was working properly. Examples 21-43, 21-44, 21-45, and 21-46 provide output from the **show ip route** command on each router.

Example 21-43 *Baseline Output for Router R1*

```
R1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

172.16.0.0/30 is subnetted, 2 subnets
D EX    172.16.1.0 [170/1734656] via 192.168.0.22, 00:04:39, FastEthernet0/1
D EX    172.16.2.0 [170/1734656] via 192.168.0.22, 00:04:39, FastEthernet0/1
10.0.0.0/8 is variably subnetted, 6 subnets, 3 masks
D        10.2.2.2/32 [90/156160] via 192.168.0.22, 00:04:39, FastEthernet0/1
D EX    10.1.3.0/30 [170/1734656] via 192.168.0.22, 00:04:39, FastEthernet0/1
D EX    10.3.3.3/32 [170/1734656] via 192.168.0.22, 00:04:39, FastEthernet0/1
D EX    10.1.2.0/24 [170/1734656] via 192.168.0.22, 00:04:40, FastEthernet0/1
C        10.1.1.1/32 is directly connected, Loopback0
D EX    10.4.4.4/32 [170/1734656] via 192.168.0.22, 00:04:40, FastEthernet0/1
C       192.168.0.0/24 is directly connected, FastEthernet0/1
C       192.168.1.0/24 is directly connected, FastEthernet0/0
```

Example 21-44 *Baseline Output for Router R2*

```
R2#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

172.16.0.0/30 is subnetted, 2 subnets
C        172.16.1.0 is directly connected, Serial1/0.1
C        172.16.2.0 is directly connected, Serial1/0.2
10.0.0.0/8 is variably subnetted, 6 subnets, 3 masks
```

```

C      10.2.2.2/32 is directly connected, Loopback0
O      10.1.3.0/30 [110/144] via 172.16.2.2, 00:07:12, Serial1/0.2
O      10.3.3.3/32 [110/91] via 172.16.2.2, 00:07:12, Serial1/0.2
O      10.1.2.0/24 [110/90] via 172.16.2.2, 00:07:12, Serial1/0.2
D      10.1.1.1/32 [90/409600] via 192.168.0.11, 00:04:46, FastEthernet0/0
O      10.4.4.4/32 [110/81] via 172.16.2.2, 00:07:12, Serial1/0.2
C      192.168.0.0/24 is directly connected, FastEthernet0/0
D      192.168.1.0/24 [90/284160] via 192.168.0.11, 00:04:46, FastEthernet0/0

```

Example 21-45 Baseline Output for Router BB1

```

BB1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

      172.16.0.0/30 is subnetted, 2 subnets
C      172.16.1.0 is directly connected, Serial1/0.2
O      172.16.2.0 [110/90] via 10.1.2.2, 00:07:08, FastEthernet0/0
      10.0.0.0/8 is variably subnetted, 6 subnets, 3 masks
O E2   10.2.2.2/32 [110/64] via 10.1.2.2, 00:07:08, FastEthernet0/0
C      10.1.3.0/30 is directly connected, Serial1/0.1
C      10.3.3.3/32 is directly connected, Loopback0
C      10.1.2.0/24 is directly connected, FastEthernet0/0
O E2   10.1.1.1/32 [110/64] via 10.1.2.2, 00:04:49, FastEthernet0/0
O      10.4.4.4/32 [110/11] via 10.1.2.2, 00:07:08, FastEthernet0/0
O E2   192.168.0.0/24 [110/64] via 10.1.2.2, 00:07:08, FastEthernet0/0
O E2   192.168.1.0/24 [110/64] via 10.1.2.2, 00:04:49, FastEthernet0/0

```

Example 21-46 Baseline Output for Router BB2

```

BB2#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

```

```

172.16.0.0/30 is subnetted, 2 subnets
O      172.16.1.0 [110/143] via 10.1.2.1, 00:08:48, FastEthernet0/0
C      172.16.2.0 is directly connected, Serial1/0.2
10.0.0.0/8 is variably subnetted, 6 subnets, 3 masks
O E2   10.2.2.2/32 [110/64] via 172.16.2.1, 00:08:48, Serial1/0.2
C      10.1.3.0/30 is directly connected, Serial1/0.1
O      10.3.3.3/32 [110/11] via 10.1.2.1, 00:08:48, FastEthernet0/0
C      10.1.2.0/24 is directly connected, FastEthernet0/0
O E2   10.1.1.1/32 [110/64] via 172.16.2.1, 00:06:30, Serial1/0.2
C      10.4.4.4/32 is directly connected, Loopback0
O E2   192.168.0.0/24 [110/64] via 172.16.2.1, 00:08:48, Serial1/0.2
O E2   192.168.1.0/24 [110/64] via 172.16.2.1, 00:06:30, Serial1/0.2

```

Router R2, acting as a boundary router, had previously been configured for mutual route redistribution. Example 21-47 illustrates this route redistribution configuration.

Example 21-47 Mutual Route Redistribution on Router R2

```

R2#show run begin router
router eigrp 100
 redistribute ospf 1 metric 1500 100 255 1 1500
 network 10.2.2.2 0.0.0.0
 network 192.168.0.0
 no auto-summary
!
router ospf 1
 redistribute eigrp 100 metric 64 subnets
 network 172.16.1.0 0.0.0.3 area 0
 network 172.16.2.0 0.0.0.3 area 0

```

To begin the troubleshooting process, you issue the **show ip route** command on all routers to determine exactly what routes are missing from the IP routing table of each router.

Router R1's IP routing table lacks all OSPF-learned routes, as shown in Example 21-48.

Example 21-48 Router R1's IP Routing Table

```

R1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

10.0.0.0/32 is subnetted, 2 subnets

```

```

D      10.2.2.2 [90/156160] via 192.168.0.22, 00:09:44, FastEthernet0/1
C      10.1.1.1 is directly connected, Loopback0
C      192.168.0.0/24 is directly connected, FastEthernet0/1
C      192.168.1.0/24 is directly connected, FastEthernet0/0

```

Router R2, which is acting as the boundary router, is actively participating in both the EIGRP and OSPF routing processes. Therefore, all routes are visible in the IP routing table of router R2, as shown in Example 21-49.

Example 21-49 IP Routing Table of Router R2

```

R2#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

      172.16.0.0/30 is subnetted, 2 subnets
C      172.16.1.0 is directly connected, Serial1/0.1
C      172.16.2.0 is directly connected, Serial1/0.2
      10.0.0.0/8 is variably subnetted, 6 subnets, 3 masks
C      10.2.2.2/32 is directly connected, Loopback0
O      10.1.3.0/30 [110/144] via 172.16.2.2, 00:07:12, Serial1/0.2
O      10.3.3.3/32 [110/91] via 172.16.2.2, 00:07:12, Serial1/0.2
O      10.1.2.0/24 [110/90] via 172.16.2.2, 00:07:12, Serial1/0.2
D      10.1.1.1/32 [90/409600] via 192.168.0.11, 00:04:46, FastEthernet0/0
O      10.4.4.4/32 [110/81] via 172.16.2.2, 00:07:12, Serial1/0.2
C      192.168.0.0/24 is directly connected, FastEthernet0/0
D      192.168.1.0/24 [90/284160] via 192.168.0.11, 00:04:46, FastEthernet0/0

```

Router BB1, which is running OSPF, has some routes that originated in EIGRP. However, the 10.1.1.1/32 and the 10.2.2.2/32 networks, which are the IP addresses of the Loopback 0 interfaces on routers R1 and R2, are missing from the IP routing table of router BB1, as illustrated in Example 21-50.

Example 21-50 IP Routing Table of Router BB1

```

BB1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

```

```

        ia - IS-IS inter area, * - candidate default, U - per-user static route
        o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

    172.16.0.0/30 is subnetted, 2 subnets
C       172.16.1.0 is directly connected, Serial1/0.2
O       172.16.2.0 [110/90] via 10.1.2.2, 00:13:00, FastEthernet0/0
    10.0.0.0/8 is variably subnetted, 4 subnets, 3 masks
C       10.1.3.0/30 is directly connected, Serial1/0.1
C       10.3.3.3/32 is directly connected, Loopback0
C       10.1.2.0/24 is directly connected, FastEthernet0/0
O       10.4.4.4/32 [110/11] via 10.1.2.2, 00:13:00, FastEthernet0/0
O E2 192.168.0.0/24 [110/64] via 10.1.2.2, 00:01:14, FastEthernet0/0
O E2 192.168.1.0/24 [110/64] via 10.1.2.2, 00:01:14, FastEthernet0/0

```

The IP routing table of router BB2, as depicted in Example 21-51, is similar to the IP routing table of router BB1.

Example 21-51 *IP Routing Table of Router BB2*

```

BB2#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default, U - per-user static route
        o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

    172.16.0.0/30 is subnetted, 2 subnets
O       172.16.1.0 [110/143] via 10.1.2.1, 00:13:39, FastEthernet0/0
C       172.16.2.0 is directly connected, Serial1/0.2
    10.0.0.0/8 is variably subnetted, 4 subnets, 3 masks
C       10.1.3.0/30 is directly connected, Serial1/0.1
O       10.3.3.3/32 [110/11] via 10.1.2.1, 00:13:39, FastEthernet0/0
C       10.1.2.0/24 is directly connected, FastEthernet0/0
C       10.4.4.4/32 is directly connected, Loopback0
O E2 192.168.0.0/24 [110/64] via 172.16.2.1, 00:01:53, Serial1/0.2
O E2 192.168.1.0/24 [110/64] via 172.16.2.1, 00:01:53, Serial1/0.2

```

Because router R2 is acting as the boundary router, you examine its redistribution configuration, as shown in Example 21-52.

Example 21-52 *Redistribution Configuration on Router R2*

```

R2#show run | begin router
router eigrp 100
  redistribute ospf 1
  network 10.2.2.2 0.0.0.0
  network 192.168.0.0
  no auto-summary
!
router ospf 1
  log-adjacency-changes
  redistribute eigrp 100 metric 64
  network 172.16.1.0 0.0.0.3 area 0
  network 172.16.2.0 0.0.0.3 area 0

```

Take a moment to look through the baseline configuration information, the topology, and the **show** command output collected after the issue was reported. Then hypothesize the underlying cause or causes of the reported issue, explaining why routers R1, BB1, and BB2 do not see all the networks in the topology, even though mutual redistribution does appear to be configured on router R2.

Suggested Solution

After examining the redistribution configuration on router R2, you might have noticed the following issues.

The EIGRP routing process on router R2 lacked a default metric, which would be assigned to routes being redistributed into the EIGRP routing process. Example 21-53 shows the commands used to correct this misconfiguration.

Example 21-53 *Adding a Default Metric for Router R2's EIGRP Routing Process*

```

R2#conf term
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#router eigrp 100
R2(config-router)#default-metric 1500 100 255 1 1500
R2(config-router)#end

```

The OSPF routing process lacked the subnets parameter at the end of the **redistribute** command. The subnets parameter is required to allow classless networks (subnets) to be redistributed into OSPF. Example 21-54 illustrates how this configuration can be corrected.

Example 21-54 *Redistributing Subnets into Router R2's OSPF Routing Process*

```

R2#conf term
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#router ospf 1
R2(config-router)#no redistribute eigrp 100 metric 64
R2(config-router)#redistribute eigrp 100 metric 64 subnets
R2(config-router)#end

```


After making the suggested corrections, all routers in the topology have IP routing tables that contain all advertised networks. Examples 21-55, 21-56, 21-57, and 21-58 illustrate the IP routing tables of these routers.

Example 21-55 IP Routing Table of Router R1

```
R1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

      172.16.0.0/30 is subnetted, 2 subnets
D EX    172.16.1.0 [170/1734656] via 192.168.0.22, 00:04:39, FastEthernet0/1
D EX    172.16.2.0 [170/1734656] via 192.168.0.22, 00:04:39, FastEthernet0/1
      10.0.0.0/8 is variably subnetted, 6 subnets, 3 masks
D        10.2.2.2/32 [90/156160] via 192.168.0.22, 00:18:05, FastEthernet0/1
D EX    10.1.3.0/30 [170/1734656] via 192.168.0.22, 00:04:39, FastEthernet0/1
D EX    10.3.3.3/32 [170/1734656] via 192.168.0.22, 00:04:39, FastEthernet0/1
D EX    10.1.2.0/24 [170/1734656] via 192.168.0.22, 00:04:40, FastEthernet0/1
C        10.1.1.1/32 is directly connected, Loopback0
D EX    10.4.4.4/32 [170/1734656] via 192.168.0.22, 00:04:40, FastEthernet0/1
C      192.168.0.0/24 is directly connected, FastEthernet0/1
C      192.168.1.0/24 is directly connected, FastEthernet0/0
```

Example 21-56 IP Routing Table of Router R2

```
R2#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

      172.16.0.0/30 is subnetted, 2 subnets
C        172.16.1.0 is directly connected, Serial1/0.1
C        172.16.2.0 is directly connected, Serial1/0.2
      10.0.0.0/8 is variably subnetted, 6 subnets, 3 masks
```

```

C      10.2.2.2/32 is directly connected, Loopback0
O      10.1.3.0/30 [110/144] via 172.16.2.2, 00:21:04, Serial1/0.2
O      10.3.3.3/32 [110/91] via 172.16.2.2, 00:21:04, Serial1/0.2
O      10.1.2.0/24 [110/90] via 172.16.2.2, 00:21:04, Serial1/0.2
D      10.1.1.1/32 [90/409600] via 192.168.0.11, 00:18:38, FastEthernet0/0
O      10.4.4.4/32 [110/81] via 172.16.2.2, 00:21:04, Serial1/0.2
C      192.168.0.0/24 is directly connected, FastEthernet0/0
D      192.168.1.0/24 [90/284160] via 192.168.0.11, 00:18:38, FastEthernet0/0

```

Example 21-57 IP Routing Table of Router BB1

```

BB1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

      172.16.0.0/30 is subnetted, 2 subnets
C      172.16.1.0 is directly connected, Serial1/0.2
O      172.16.2.0 [110/90] via 10.1.2.2, 00:21:08, FastEthernet0/0
      10.0.0.0/8 is variably subnetted, 6 subnets, 3 masks
O E2   10.2.2.2/32 [110/64] via 10.1.2.2, 00:04:44, FastEthernet0/0
C      10.1.3.0/30 is directly connected, Serial1/0.1
C      10.3.3.3/32 is directly connected, Loopback0
C      10.1.2.0/24 is directly connected, FastEthernet0/0
O E2   10.1.1.1/32 [110/64] via 10.1.2.2, 00:04:44, FastEthernet0/0
O      10.4.4.4/32 [110/11] via 10.1.2.2, 00:21:08, FastEthernet0/0
O E2   192.168.0.0/24 [110/64] via 10.1.2.2, 00:04:44, FastEthernet0/0
O E2   192.168.1.0/24 [110/64] via 10.1.2.2, 00:04:44, FastEthernet0/0

```

Example 21-58 IP Routing Table of Router BB2

```

BB2#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

```

```

172.16.0.0/30 is subnetted, 2 subnets
O      172.16.1.0 [110/143] via 10.1.2.1, 00:21:13, FastEthernet0/0
C      172.16.2.0 is directly connected, Serial1/0.2
10.0.0.0/8 is variably subnetted, 6 subnets, 3 masks
O E2   10.2.2.2/32 [110/64] via 172.16.2.1, 00:04:50, Serial1/0.2
C      10.1.3.0/30 is directly connected, Serial1/0.1
O      10.3.3.3/32 [110/11] via 10.1.2.1, 00:21:13, FastEthernet0/0
C      10.1.2.0/24 is directly connected, FastEthernet0/0
O E2   10.1.1.1/32 [110/64] via 172.16.2.1, 00:04:50, Serial1/0.2
C      10.4.4.4/32 is directly connected, Loopback0
O E2   192.168.0.0/24 [110/64] via 172.16.2.1, 00:04:50, Serial1/0.2
O E2   192.168.1.0/24 [110/64] via 172.16.2.1, 00:04:50, Serial1/0.2

```

Trouble Ticket 6

You receive the following trouble ticket:

Company A (that is, routers R1 and R2) has connections to two service providers (that is, BB1 and BB2). Router R2 is running Border Gateway Protocol (BGP) and is peering with routers BB1 and BB2. The bandwidth between routers R2 and BB2 is greater than the bandwidth between routers R2 and BB1. Therefore, company A wants to use the R2-to-BB2 link as the primary link to the backbone network (that is, a default route). However, company A noticed that the R2-to-BB1 link is being used.

This trouble ticket references the topology shown in Figure 21-6.

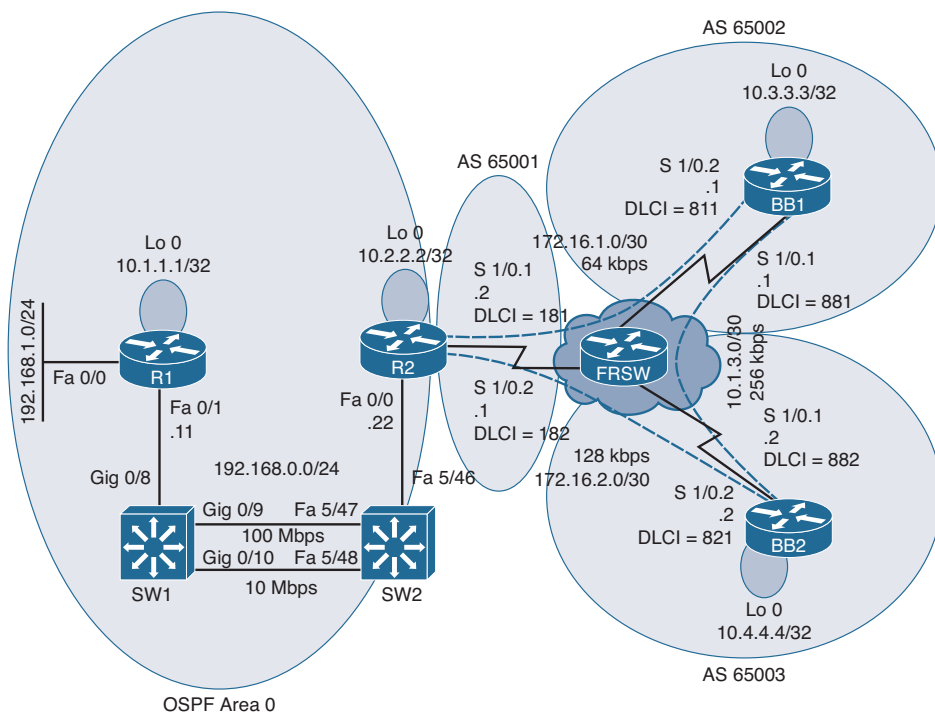


Figure 21-6 Trouble Ticket 6 Topology

You begin by examining the baseline data collected after company A was dual-homed to its two Internet service providers (ISPs). Example 21-59 shows the output from the **show ip route** command on router R1. Notice that router R1 has a default route in its IP routing table. This default route was learned via OSPF from router R2.

Example 21-59 *Baseline Output for Router R1*

```
R1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 192.168.0.22 to network 0.0.0.0

    10.0.0.0/32 is subnetted, 2 subnets
O       10.2.2.2 [110/2] via 192.168.0.22, 00:05:33, FastEthernet0/1
C       10.1.1.1 is directly connected, Loopback0
C       192.168.0.0/24 is directly connected, FastEthernet0/1
C       192.168.1.0/24 is directly connected, FastEthernet0/0
O*E2 0.0.0.0/0 [110/1] via 192.168.0.22, 00:05:33, FastEthernet0/1
```

Router R2 was configured for both OSPF and BGP, with the BGP-learned default route being injected into OSPF, and with OSPF-learned routes being redistributed into BGP. Example 21-60 shows the initial IP routing table for router R2. Notice that the next-hop router for the default route is 172.16.1.1 (that is, router BB1).

Example 21-60 *Baseline IP Routing Table on Router R2*

```
R2#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 172.16.1.1 to network 0.0.0.0

    172.16.0.0/30 is subnetted, 2 subnets
C       172.16.1.0 is directly connected, Serial1/0.1
C       172.16.2.0 is directly connected, Serial1/0.2
    10.0.0.0/8 is variably subnetted, 6 subnets, 3 masks
C       10.2.2.2/32 is directly connected, Loopback0
```

```

B      10.1.3.0/30 [20/0] via 172.16.1.1, 00:01:40
B      10.3.3.3/32 [20/0] via 172.16.1.1, 00:01:40
B      10.1.2.0/24 [20/0] via 172.16.1.1, 00:01:40
O      10.1.1.1/32 [110/11] via 192.168.0.11, 00:08:17, FastEthernet0/0
B      10.4.4.4/32 [20/0] via 172.16.2.2, 00:01:40
C      192.168.0.0/24 is directly connected, FastEthernet0/0
O      192.168.1.0/24 [110/11] via 192.168.0.11, 00:08:17, FastEthernet0/0
B*    0.0.0.0/0 [20/0] via 172.16.1.1, 00:01:40

```

Example 21-61 illustrates the initial OSPF and BGP configuration on router R2.

Example 21-61 Initial Router Configuration on Router R2

```

R2#show run | begin router
router ospf 1
  log-adjacency-changes
  network 10.2.2.2 0.0.0.0 area 0
  network 192.168.0.0 0.0.0.255 area 0
  default-information originate
!
router bgp 65001
  no synchronization
  bgp log-neighbor-changes
  network 172.16.1.0 mask 255.255.255.252
  network 172.16.2.0 mask 255.255.255.252
  redistribute ospf 1
  neighbor 172.16.1.1 remote-as 65002
  neighbor 172.16.2.2 remote-as 65003
  no auto-summary

```

Example 21-62 shows the output of the **show ip bgp summary** command on router R2, which confirms that router R2 resides in BGP autonomous system 65001. The output also confirms BGP adjacencies have been formed with routers BB1 and BB2.

Example 21-62 BGP Configuration Summary on Router R2

```

R2#show bgp ipv4 unicast summary
BGP router identifier 10.2.2.2, local AS number 65001
BGP table version is 18, main routing table version 18
11 network entries using 1287 bytes of memory
20 path entries using 1040 bytes of memory
8/5 BGP path/bestpath attribute entries using 992 bytes of memory
4 BGP AS-PATH entries using 96 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 3415 total bytes of memory
BGP activity 38/27 prefixes, 75/55 paths, scan interval 60 secs

```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
172.16.1.1	4	65002	102	97	18	0	0	00:02:47	7
172.16.2.2	4	65003	100	97	18	0	0	00:02:47	7

Router BB1 is configured for BGP and is sourcing a default route advertisement. Example 21-63 shows the IP routing table of router BB1.

Example 21-63 Initial IP Routing Table on Router BB1

```
BB1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 0.0.0.0 to network 0.0.0.0

    172.16.0.0/30 is subnetted, 2 subnets
C      172.16.1.0 is directly connected, Serial1/0.2
B      172.16.2.0 [20/0] via 10.1.3.2, 00:03:01
    10.0.0.0/8 is variably subnetted, 6 subnets, 3 masks
B      10.2.2.2/32 [20/0] via 172.16.1.2, 00:01:59
C      10.1.3.0/30 is directly connected, Serial1/0.1
C      10.3.3.3/32 is directly connected, Loopback0
C      10.1.2.0/24 is directly connected, FastEthernet0/0
B      10.1.1.1/32 [20/11] via 172.16.1.2, 00:01:59
B      10.4.4.4/32 [20/0] via 10.1.3.2, 00:40:10
B      192.168.0.0/24 [20/0] via 172.16.1.2, 00:01:59
B      192.168.1.0/24 [20/11] via 172.16.1.2, 00:01:59
S*    0.0.0.0/0 is directly connected, Null0
```

Router BB2's IP routing table, as shown in Example 21-64, is similar to router BB1's IP routing table. Notice that router BB2 is also sourcing a default route and is advertising it via BGP to router R2. Therefore, router R2 has two paths to reach a default route in its BGP table.

Example 21-64 Initial IP Routing Table on Router BB2

```
BB2#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route
```

```

Gateway of last resort is 0.0.0.0 to network 0.0.0.0

    172.16.0.0/30 is subnetted, 2 subnets
B       172.16.1.0 [20/0] via 10.1.3.1, 00:03:11
C       172.16.2.0 is directly connected, Serial1/0.2
    10.0.0.0/8 is variably subnetted, 6 subnets, 3 masks
B       10.2.2.2/32 [20/0] via 172.16.2.1, 00:02:09
C       10.1.3.0/30 is directly connected, Serial1/0.1
B       10.3.3.3/32 [20/0] via 10.1.3.1, 00:40:10
C       10.1.2.0/24 is directly connected, FastEthernet0/0
B       10.1.1.1/32 [20/11] via 172.16.2.1, 00:02:09
C       10.4.4.4/32 is directly connected, Loopback0
B       192.168.0.0/24 [20/0] via 172.16.2.1, 00:02:09
B       192.168.1.0/24 [20/11] via 172.16.2.1, 00:02:09
S*    0.0.0.0/0 is directly connected, Null0

```

As shown earlier, in Example 21-60, router R2 preferred the 64-Kbps link to router BB1 to reach a default route, as opposed to the 128-Kbps link to router BB2. Therefore, the outbound routing from router R2 is suboptimal.

Also, the inbound routing, coming into the enterprise via router R2, is suboptimal. To illustrate this point, consider Example 21-65, which shows the BGP table on router BB1. Notice that router BB1 prefers a next-hop router of router R2 to reach the 10.1.1.1/32 network, which resides inside the enterprise network (that is, the network consisting of routers R1 and R2). Using a next-hop router of R2 would force traffic over the 64-Kbps link rather than sending traffic from router BB1 over the 256-Kbps link to router BB2, and then over the 128-Kbps link to router R2, and finally across the Fast Ethernet connection to router R1.

Example 21-65 BGP Forwarding Table on Router BB1

```

BB1#show bgp ipv4 unicast
BGP table version is 130, local router ID is 10.3.3.3
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop        Metric LocPrf Weight Path
*  0.0.0.0          10.1.3.2             0           0 65003 i
*>                 0.0.0.0             0          32768 i
* 10.1.1.1/32       10.1.3.2             0           0 65003 65001 ?
*>                 172.16.1.2          11           0 65001 ?
* 10.1.2.0/24       10.1.3.2             0           0 65003 i
*>                 0.0.0.0             0          32768 i
* 10.1.3.0/30       10.1.3.2             0           0 65003 i
*>                 0.0.0.0             0          32768 i
* 10.2.2.2/32       10.1.3.2             0           0 65003 65001 ?
*>                 172.16.1.2             0           0 65001 ?

```

*> 10.3.3.3/32	0.0.0.0	0	32768	i
* 10.4.4.4/32	172.16.1.2		0	65001 65003 i
*>	10.1.3.2	0	0	65003 i
* 172.16.1.0/30	172.16.1.2	0	0	65001 i
*>	0.0.0.0	0	32768	i
* 172.16.2.0/30	172.16.1.2	0	0	65001 i
*>	10.1.3.2	0	0	65003 i
* 192.168.0.0	10.1.3.2		0	65003 65001 ?
*>	172.16.1.2	0	0	65001 ?
* 192.168.1.0	10.1.3.2		0	65003 65001 ?
*>	172.16.1.2	11	0	65001

As you formulate your solution to correct the inbound and outbound path selection issues, you should limit your configuration to router R2. The reason for this limitation is that routers BB1 and BB2 are acting as ISP routers. In a real-world environment, the administrator of an enterprise network would not have privileges to configure the ISP routers.

BGP has multiple attributes that can be manipulated to influence path selection. The suggested solution, however, focuses on how the BGP local preference attribute can influence the outbound path selection and how the BGP AS_PATH attribute can influence the inbound path selection. You can configure route maps to set these BGP attributes. If you choose to base your solution on local preference and AS_PATH attributes, Table 21-1 provides a syntax reference that might be helpful.

Table 21-1 *Configuring AS_PATH and Local Preference BGP Attributes*

Command	Description
Router(config)# route-map <i>tag</i> [permit deny] [<i>seq-num</i>]	Creates a route map
Router(config-route-map)# set local-preference <i>local-preference</i>	Sets the local preference BGP attribute for routes matched by a route map
Router(config-route-map)# set as-path prepend <i>autonomous-system-number-1</i> [...] <i>autonomous-system-number-n</i>]	Defines an autonomous system path to prepend to an autonomous system path known by the BGP forwarding table
Router(config)# router bgp <i>as-number</i>	Enables a BGP process for a specific autonomous system
Router(config-router)# neighbor ip-address route-map <i>route-map-name</i> [in out]	Applies a specified route map to routes received from or advertised to a specified BGP neighbor

Take a moment to look through the provided show command output. Then, on a separate sheet of paper, create a plan for correcting the suboptimal path selection.

Suggested Solution

Local preference values can be applied to routes coming into a router. This can cause that router to make its outbound routing decisions based on those local preference values. Higher local preference values are preferred over lower local preference values.

An autonomous system path (that is, a listing of the autonomous systems that must be transited to reach a specific destination network) advertised to a neighbor can influence the BGP path selection of that neighbor. Specifically, BGP can make routing decisions based on the smallest number of autonomous systems that must be crossed to reach a destination network. Using a route map, you can prepend one or more additional instances of your local autonomous system to the AS_PATH advertised to a router's neighbor, thereby making that path appear less attractive to your neighbor.

Therefore, the suggested solution configures local preference values for routes advertised into router R2 from routers BB1 and BB2 to prefer routes being advertised via router BB2. Example 21-66 shows this configuration, which influences outbound path selection.

Example 21-66 *Local Preference Configuration on Router R2*

```
R2 (config)#route-map LOCALPREF-BB1
R2 (config-route-map)#set local-preference 100
R2 (config-route-map)#exit
R2 (config)#route-map LOCALPREF-BB2
R2 (config-route-map)#set local-preference 200
R2 (config-route-map)#exit
R2 (config)#router bgp 65001
R2 (config-router)#neighbor 172.16.1.1 route-map LOCALPREF-BB1 in
R2 (config-router)#neighbor 172.16.2.2 route-map LOCALPREF-BB2 in
R2 (config-router)#exit
```

To influence inbound path selection, this suggested solution configures a route map to prepend two additional instances of autonomous system 65001 to routes being advertised via BGP from router R2 to router BB1. Example 21-67 shows this configuration, which causes router BB1 to use router BB2 as a next-hop router when sending traffic into the enterprise network. It does this because the path via router BB2 appears to be fewer autonomous system hops away from the enterprise networks.

Example 21-67 *AS_PATH Configuration on Router R2*

```
R2 (config)#route-map ASPATH 10
R2 (config-route-map)#set as-path prepend 65001 65001
R2 (config-route-map)#exit
R2 (config)#router bgp 65001
R2 (config-router)#neighbor 172.16.1.1 route-map ASPATH out
R2 (config-router)#end
```

Example 21-68 confirms that router R2 now prefers router BB2 (that is, a next-hop IP address of 172.16.2.2) to reach the default network.

Example 21-68 Preferred Path of Router R2 to Backbone Networks

```
R2#show bgp ipv4 unicast
BGP table version is 16, local router ID is 10.2.2.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Next Hop	Metric	LocPrf	Weight	Path
* 0.0.0.0	172.16.1.1	0	100	0	65002 i
*>	172.16.2.2	0	200	0	65003 i
*> 10.1.1.1/32	192.168.0.11	11		32768	?
* 10.1.2.0/24	172.16.1.1	0	100	0	65002 i
*>	172.16.2.2	0	200	0	65003 i
* 10.1.3.0/30	172.16.1.1	0	100	0	65002 i
*>	172.16.2.2	0	200	0	65003 i
*> 10.2.2.2/32	0.0.0.0	0		32768	?
* 10.3.3.3/32	172.16.1.1	0	100	0	65002 i
*>	172.16.2.2		200	0	65003 65002 i
* 10.4.4.4/32	172.16.1.1		100	0	65002 65003 i
*>	172.16.2.2	0	200	0	65003 i
*> 172.16.1.0/30	0.0.0.0	0		32768	i
*	172.16.1.1	0	100	0	65002 i
*	172.16.2.2		200	0	65003 65002 i
*> 172.16.2.0/30	0.0.0.0	0		32768	i
*	172.16.1.1		100	0	65002 65003 i
*	172.16.2.2	0	200	0	65003 i
*> 192.168.0.0	0.0.0.0	0		32768	?
*> 192.168.1.0	192.168.0.11	11		32768	

Example 21-69 confirms that router BB1 will not prefer to send traffic to the enterprise network (that is, to routers R1 and R2) via router R2, but rather via router BB2. Notice from the output that more autonomous system hops appear to be required to reach enterprise networks via router R2 (that is, 172.16.1.2) compared to router BB2 (that is, 10.1.3.2). Therefore, router BB1 prefers to send traffic into the enterprise network via router BB2, as opposed to router R2.

Example 21-69 Preferred Path of Router BB1 to Enterprise Networks

```
BB1#show bgp ipv4 unicast
BGP table version is 142, local router ID is 10.3.3.3
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Next Hop	Metric	LocPrf	Weight	Path
* 0.0.0.0	172.16.1.2			0	65001 65001 65001 65003 i
*	10.1.3.2	0		0	65003 i

*>	0.0.0.0	0	32768	i
*> 10.1.1.1/32	10.1.3.2		0	65003 65001 ?
*	172.16.1.2	11	0	65001 65001 65001 ?
* 10.1.2.0/24	172.16.1.2		0	65001 65001 65001 65003 i
*	10.1.3.2	0	0	65003 i
*>	0.0.0.0	0	32768	i
* 10.1.3.0/30	172.16.1.2		0	65001 65001 65001 65003 i
*	10.1.3.2	0	0	65003 i
*>	0.0.0.0	0	32768	i
*> 10.2.2.2/32	10.1.3.2		0	65003 65001 ?
*	172.16.1.2	0	0	65001 65001 65001 ?
*> 10.3.3.3/32	0.0.0.0	0	32768	i
* 10.4.4.4/32	172.16.1.2		0	65001 65001 65001 65003 i
*>	10.1.3.2	0	0	65003 i
* 172.16.1.0/30	172.16.1.2	0	0	65001 65001 65001 i
*>	0.0.0.0	0	32768	i
* 172.16.2.0/30	172.16.1.2	0	0	65001 65001 65001 i
*>	10.1.3.2	0	0	65003 i
*> 192.168.0.0	10.1.3.2		0	65003 65001 ?
*	172.16.1.2	0	0	65001 65001 65001 ?
*> 192.168.1.0	10.1.3.2		0	65003 65001 ?
*	172.16.1.2	11	0	65001 65001 65001

Trouble Ticket 7

You receive the following trouble ticket:

A new administrator for company A has forgotten the enable secret password assigned to router R1 and can no longer log in. Also, when this administrator connects to router R2 via Telnet, the connection is timed out after only 1 second. The administrator reports this short timeout does not give him sufficient time to correct the configuration. Also, the administrator configured an access list on router R2 to prevent anyone on the backbone (that is, connections coming in to router R2 via the Frame Relay network) from connecting to the Loopback 0 interfaces on routers R1 or R2 via Telnet. However, the access list does not seem to be working.

This trouble ticket references the topology shown in Figure 21-7.

The trouble ticket identified the following three issues:

- A forgotten enable secret password
- An `exec-timeout` parameter set too low
- An ACL misconfiguration

The sections that follow address each issue individually.

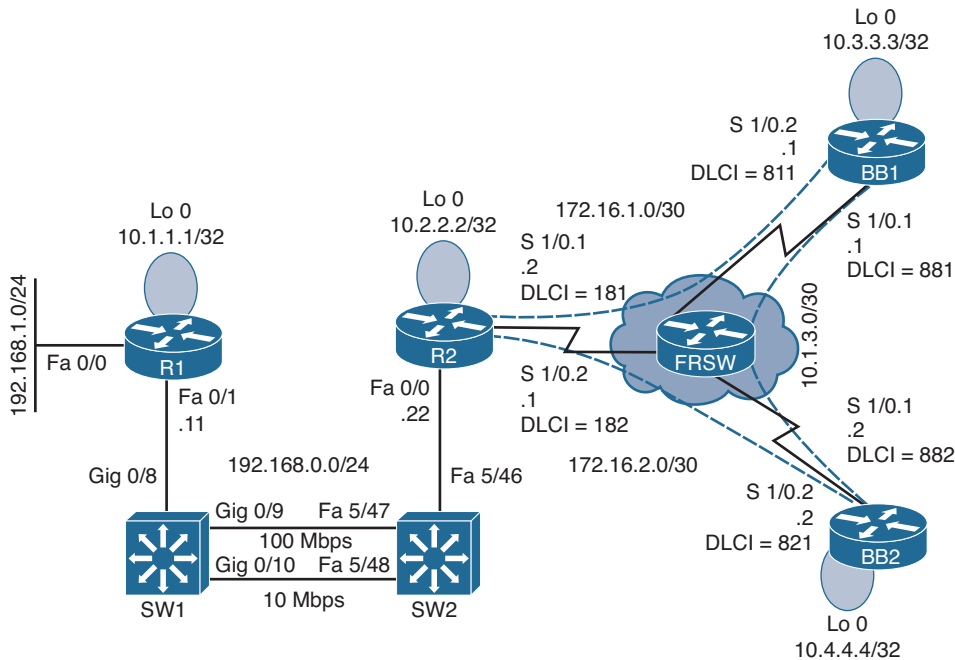


Figure 21-7 *Trouble Ticket 7 Topology*

Issue 1: Forgotten Enable Secret Password

The first issue to be addressed by this trouble ticket is password recovery. The administrator reportedly forgot the enable secret password for router R1, which is a Cisco 2900 series router.

On a separate sheet of paper, write out the steps you would go through to perform password recovery on this router. If you are not familiar with password recovery steps, you might need to research password recovery at Cisco.com.

Issue 1: Suggested Solution

To begin the password recovery process on router R1, the router was rebooted, and during the first few seconds of the router booting, a **Break** was sent from the terminal emulator to the router. The **Break** caused the ROM Monitor prompt (that is, rommon) to appear on router R1's console.

The configuration register was set to 0x2142 with the command **confreg 0x2142**. Setting the configuration register to this value causes the router to ignore its startup configuration when the router boots. The router was then rebooted by issuing the **reset** command at the rommon prompt.

Because the router ignored the startup configuration, after the router booted, a prompt was presented, asking the administrator whether he wanted to go through the setup dialog. A no was entered at this prompt. The **enable** command was entered to go into privileged configuration mode. From privileged mode, the startup configuration, stored

in the router's NVRAM, was merged with the existing running configuration using the command **copy star run**. This command does not *replace* the running configuration with the startup configuration. Rather, these two configurations are *merged*. After this merger, all the physical interfaces were administratively shut down. Therefore, the **no shutdown** command was entered for interfaces Fast Ethernet 0/0 and Fast Ethernet 0/1.

The enable secret password was reset to cisco using the command **enable secret cisco**. Next, the configuration register was set back to its normal value of 0x2102 with the command **config-register 0x2102**. The running configuration was copied to the startup configuration with the command **copy run star**. The router was then rebooted with the **reload** command. After the router rebooted, the administrator could access the router's privileged mode using an enable secret password of cisco. Example 21-70 demonstrates this password-recovery procedure.

Example 21-70 Performing Password Recovery on Router R1

```
System Bootstrap, Version 15.0(1r)M1, RELEASE SOFTWARE (fc1)
Copyright (c) 2009 by cisco Systems, Inc.
C2900 platform with 524288 Kbytes of main memory
...BREAK SEQUENCE SENT...
monitor: command "boot" aborted due to user interrupt
rommon 1 > confreg 0x2142
You must reset or power cycle for new config to take effect
rommon 2 > reset
...OUTPUT OMITTED...
      ---- System Configuration Dialog ----
Would you like to enter the initial configuration dialog? [yes/no]: no
Press RETURN to get started!
...OUTPUT OMITTED...
Router>enable
Router#copy star run
Destination filename [running-config]?
...OUTPUT OMITTED...
R1(config)#enable secret cisco
R1(config)#config-register 0x2102
R1(config)#interface fa 0/1
R1(config-if)#no shut
R1(config-if)#interface fa 0/0
R1(config-if)#no shut
R1(config-if)#end
*Mar  3 12:43:26.016: %SYS-5-CONFIG_I: Configured from console by console
R1#copy run star
Destination filename [startup-config]?
Building configuration...
[OK]
R1#reload
Proceed with reload? [confirm]
```

```
...OUTPUT OMITTED...
Press RETURN to get started!
R1>
R1>enable
Password:cisco
R1#
```

Issue 2: An exec-timeout Parameter Set Too Low

The second issue addressed in this trouble ticket is recovering from a misconfiguration on router R2, which causes a Telnet session to time out after only 1 second of inactivity. The challenge with such a misconfiguration is that when an administrator telnets to the router to correct the configuration, he might be logged out if he pauses for as little as a single second.

Example 21-71 shows router R2's misconfiguration. Note the **exec-timeout 0 1** command. This command causes a user that connected via a vty line to be timed out after only one second of inactivity.

Example 21-71 *Incorrect exec-timeout Configuration on Router R2*

```
R2#show run | begin line vty 0 4
line vty 0 4
  exec-timeout 0 1
  password cisco
  login
```

On a separate sheet of paper, write out how you would approach this seemingly paradoxical situation, where you have to log in to the router to correct the configuration, while you will be logged out of the router with only a single second's pause.

Issue 2: Suggested Solution

One fix to this issue is to continuously tap on the keyboard's down arrow with one hand, while using the other hand to enter the commands required to correct the **exec-timeout** misconfiguration. Example 21-72 shows the commands entered to set the **exec-timeout** parameter such that a Telnet session times out after 5 minutes of inactivity. You could also attach to the console or aux port on the device and change these parameters for the telnet session.

Example 21-72 *Correcting an exec-timeout Misconfiguration*

```
R2#conf term
R2 (config)#line vty 0 4
R2 (config-line)#exec-timeout 5 0
```

Issue 3: ACL Misconfiguration

This trouble ticket's final troubleshooting issue was an ACL misconfiguration. The goal of the ACL on router R2 was to prevent Telnet traffic coming in from the backbone (that is, coming in over subinterfaces Serial 1/0.1 or Serial 1/0.2) destined for the loopback interface on router R1 or R2 (that is, IP addresses 10.1.1.1 or 10.2.2.2). Example 21-73 shows the ACL configuration on router R2.

Example 21-73 *Baseline ACL Configuration on Router R2*

```
R2#show run
...OUTPUT OMITTED...
interface s1/0.1
  ip access-group 100 out
!
interface s1/0.2
  ip access-group 100 out
!
access-list 100 deny tcp any host 10.1.1.1 eq telnet
access-list 100 deny tcp any host 10.2.2.2 eq telnet
access-list 100 permit ip any any
```

Based on the trouble ticket and the proceeding **show** command output, on a separate sheet of paper, formulate your strategy for resolving the reported issue.

Issue 3: Suggested Solution

Upon examination, the ACL (an extended IP ACL numbered 100) on router R2 appears to be configured correctly. However, ACL 100 was applied in the outbound direction on router R2's Frame Relay subinterfaces. ACL 100 should have been applied in the incoming direction on these subinterfaces. This suggested solution replaces the incorrect **ip access-group** commands, as shown in Example 21-74.

Example 21-74 *Correcting the Application of ACL 100 on Router R2*

```
R2#conf term
R2(config)#interface s1/0.1
R2(config-if)#no ip access-group 100 out
R2(config-if)#ip access-group 100 in
R2(config-if)#interface s1/0.2
R2(config-if)#no ip access-group 100 out
R2(config-if)#ip access-group 100 in
```

After making the previous update, Telnet connections destined for the Loopback interfaces on routers R1 and R2, coming into router R2 over its Frame Relay subinterfaces are now denied.

Trouble Ticket 8

You receive the following trouble ticket:

Company A is dual-homed out to the Internet (that is, routers BB1 and BB2, where each router represents a different ISP). Inside IP addresses in the 192.168.0.0/24 subnet should be translated into the IP address of interface Serial 1/0.1 on router R2, whereas inside IP addresses in the 192.168.1.0/24 subnet should be translated into the IP address of interface Serial 1/0.2 on router R2. Router R2's Network Address Translation (NAT) table shows two active translations. The configuration, therefore, seems to be partially working. However, no additional NAT translations can be set up.

This trouble ticket references the topology shown in Figure 21-8.

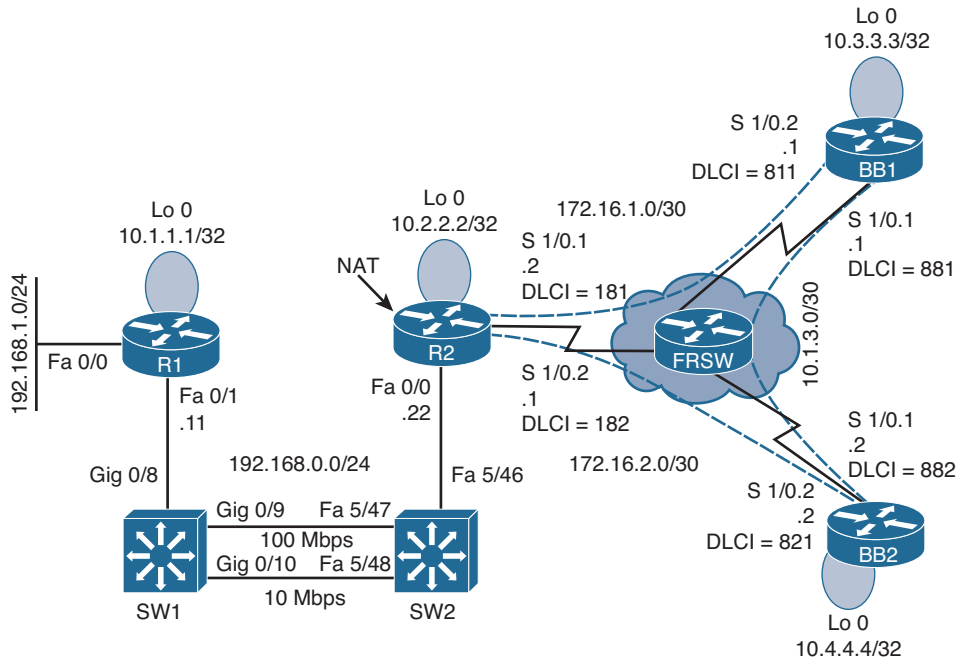


Figure 21-8 *Trouble Ticket 8 Topology*

Because router R2 is the one configured to perform NAT, the following **show** and **debug** command output collects information about the NAT configuration of router R2. Initially, notice the output of the **show ip nat translations** command issued on router R2, as shown in Example 21-75.

Example 21-75 **show ip nat translations** Command Output on Router R2

```
R2#show ip nat translations
Pro Inside global      Inside local      Outside local      Outside global
icmp 172.16.1.2:7      192.168.0.11:7   10.4.4.4:7        10.4.4.4:7
icmp 172.16.2.1:512    192.168.1.50:512 10.1.3.2:512      10.1.3.2:512
```


The **debug ip nat** command is issued next. The output provided in Example 21-76 shows NAT translations as they occur.

Example 21-76 *debug ip nat Command Output on Router R2*

```
R2#debug ip nat
IP NAT debugging is on
*Mar 1 00:34:16.651: NAT*: s=10.4.4.4, d=172.16.1.2->192.168.0.11 [4092]
*Mar 1 00:34:16.711: NAT*: s=192.168.0.11->172.16.1.2, d=10.4.4.4 [4093]
*Mar 1 00:34:16.843: NAT*: s=10.4.4.4, d=172.16.1.2->192.168.0.11 [4093]
*Mar 1 00:34:16.939: NAT*: s=192.168.0.11->172.16.1.2, d=10.4.4.4 [4094]
*Mar 1 00:34:16.963: NAT*: s=192.168.1.50->172.16.2.1, d=10.1.3.2 [13977]
*Mar 1 00:34:17.115: NAT*: s=10.4.4.4, d=172.16.1.2->192.168.0.11 [4094]
*Mar 1 00:34:17.163: NAT*: s=192.168.0.11->172.16.1.2, d=10.4.4.4 [4095]
*Mar 1 00:34:17.187: NAT*: s=10.1.3.2, d=172.16.2.1->192.168.1.50 [13977]
*Mar 1 00:34:17.315: NAT*: s=10.4.4.4, d=172.16.1.2->192.168.0.11 [4095]
```

The trouble ticket indicated that no more than two active translations can be supported at any time. To verify that symptom, Example 21-77 shows an attempt to send a ping from router R1. Notice that the ping response indicates that 10.4.4.4 is unreachable.

Example 21-77 *Attempting to Ping 10.4.4.4 from Router R1*

```
R1#ping 10.4.4.4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.4.4.4, timeout is 2 seconds:
U.U.U
Success rate is 0 percent (0/5)
```

To determine whether the inability to ping 10.4.4.4 is a result of NAT or some other issue, the NAT translation table on router R2 is cleared with the **clear ip nat translation *** command. Then, with the NAT translation table of router R2 cleared, Example 21-78 shows the result of another ping from router R1 to 10.4.4.4. This time, the ping is successful.

Example 21-78 *Reattempting to Ping 10.4.4.4 from Router R1*

```
R1#ping 10.4.4.4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.4.4.4, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 72/137/240 ms
```

Example 21-79 shows the NAT translation table of router R2 after R1 performs a ping to 10.4.4.4.

Example 21-79 NAT Translation Table of Router R2

R2#show ip nat translations				
Pro	Inside global	Inside local	Outside local	Outside global
icmp	172.16.1.2:10	192.168.0.11:10	10.4.4.4:10	10.4.4.4:10

The output from the previous commands confirms that router R2 is capable of supporting only two simultaneous NAT translations. This symptom often indicates that a router's NAT pool (or pools in this case) is depleted, perhaps because the NAT configuration did not use the **overload** option in the **ip nat inside source** command. Recall that the **overload** option enables PAT, which allows multiple inside local IP addresses to share a common inside global IP address.

Example 21-80 shows the running configuration of router R2. Interestingly, both the **ip nat inside source** commands have the **overload** option, thus eliminating that as a potential cause for the reported issue.

Example 21-80 Running Configuration of Router R2

```
R2#show run
...OUTPUT OMITTED...
hostname R2
!
interface Loopback0
 ip address 10.2.2.2 255.255.255.255
!
interface FastEthernet0/0
 ip address 192.168.0.22 255.255.255.0
 ip nat inside
!
interface Serial1/0
 no ip address
 encapsulation frame-relay
!
interface Serial1/0.1 point-to-point
 ip address 172.16.1.2 255.255.255.252
 ip nat outside
 frame-relay interface-dlci 181
!
interface Serial1/0.2 point-to-point
 ip address 172.16.2.1 255.255.255.252
 ip nat outside
 ip virtual-reassembly
 frame-relay interface-dlci 182
!
router ospf 1
 network 0.0.0.0 255.255.255.255 area 0
!
```

```
ip nat translation max-entries 2
ip nat inside source list 1 interface Serial1/0.2 overload
ip nat inside source list 2 interface Serial1/0.1 overload
!
access-list 1 permit 192.168.1.0 0.0.0.255
access-list 2 permit 192.168.0.0 0.0.0.255
!
...OUTPUT OMITTED...
```

Based on the output of the previous **show** and **debug** commands, on a separate sheet of paper, write out what you believe to be the underlying issue and how you would resolve it.

Suggested Solution

In the running configuration of router R2, you might have noticed the **ip nat translation max-entries 2** command. This command limits the maximum number of NAT translations on router R2 to only two.

To resolve this issue, this configuration command is removed, as shown in Example 21-81.

Example 21-81 Removing the `ip nat translation max-entries 2` Command of Router R2

```
R2#conf term
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#no ip nat translation max-entries 2
R2(config)#end
```

To demonstrate that the removal of the **ip nat translation max-entries 2** command did indeed resolve the reported issue, three NAT translations were established across router R2, as confirmed in Example 21-82.

Example 21-82 Confirming That Router R2 Supports Multiple NAT Translations

```
R2#show ip nat translations
```

Pro	Inside global	Inside local	Outside local	Outside global
icmp	172.16.1.2:12	192.168.0.11:12	10.4.4.4:12	10.4.4.4:12
icmp	172.16.2.1:13	192.168.1.11:13	10.3.3.3:13	10.3.3.3:13
icmp	172.16.2.1:512	192.168.1.50:512	10.1.3.2:512	10.1.3.2:512

Trouble Ticket 9

You receive the following trouble ticket:

Company A recently added IPv6 addressing to its existing IPv4 addressing. OSPFv3 is the protocol being used to route the IPv6 traffic. Although the configuration was originally functional, now several OSPFv3 adjacencies are not forming. Full IPv6 reachability throughout the topology needs to be established.

This trouble ticket references the topology shown in Figure 21-9.

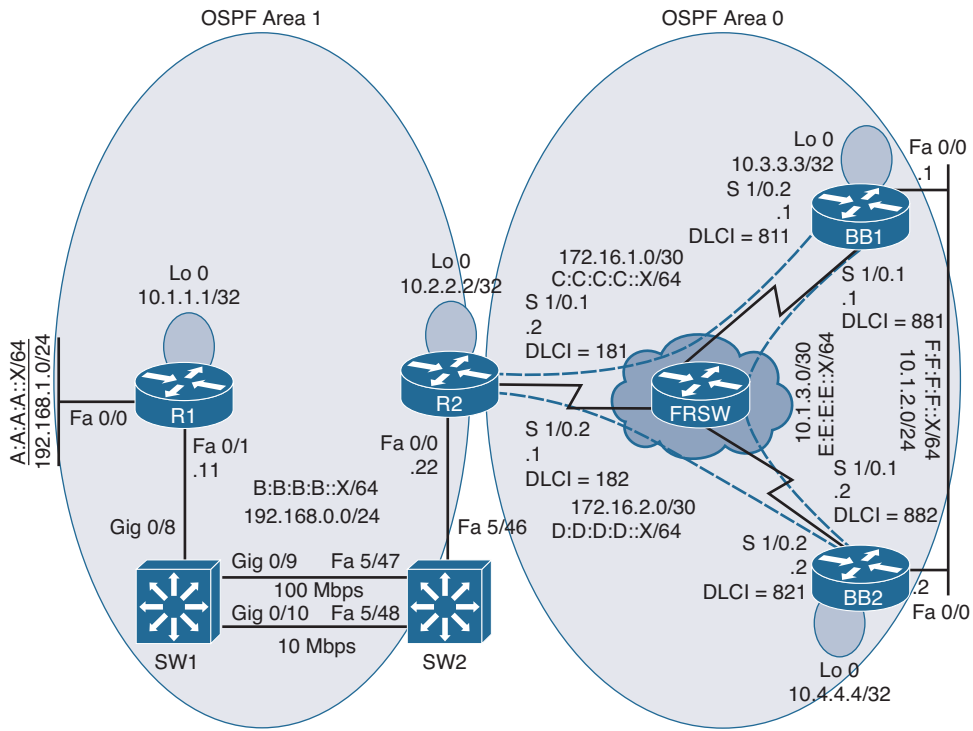


Figure 21-9 *Trouble Ticket 9 Topology*

The trouble ticket indicates that several adjacencies are not being formed. So, you decide to start your troubleshooting efforts on router R1 and check its adjacency with router R2, and then check the adjacencies between R2 and BB1 and BB2. Finally, you will check the adjacencies between BB1 and BB2.

Issue 1: Adjacency Between Routers R1 and R2

Example 21-83 shows the data collected from router R1.

Example 21-83 *Troubleshooting Data Collection on Router R1*

```
R1#show ipv6 ospf neighbor

R1#debug ipv6 ospf adj
OSPFv3 adjacency events debugging is on
R1#debug ipv6 ospf hello
OSPFv3 hello events debugging is on
R1#u all
All possible debugging has been turned off
R1#show run
...OUTPUT OMITTED...
hostname R1
```

```

!
ipv6 unicast-routing
ipv6 cef
!
interface Loopback0
 ip address 10.1.1.1 255.255.255.255
!
interface FastEthernet0/0
 ip address 192.168.1.11 255.255.255.0
 ipv6 address A:A:A:A::11/64
 ipv6 ospf 100 area 1
!
interface FastEthernet0/1
 ip address 192.168.0.11 255.255.255.0
 ipv6 address B:B:B:B::11/64
 ipv6 ospf 100 area 1
!
ipv6 router ospf 100
!
...OUTPUT OMITTED...
R1#show ipv6 ospf interface fa 0/1
FastEthernet0/1 is up, line protocol is up
  Link Local Address FE80::209:B7FF:FEFA:D1E1, Interface ID 4
  Area 1, Process ID 100, Instance ID 0, Router ID 192.168.1.11
  Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 192.168.1.11, local address FE80::209:B7FF:FEFA:D1E1
  No backup designated router on this network
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:03
  Index 1/2/2, flood queue length 0
  Next 0x0(0)/0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 2
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 0, Adjacent neighbor count is 0
  Suppress hello for 0 neighbor(s)

```

Notice that router R1 has not formed an adjacency with router R2 and there are no Hello packets being exchanged between the two routers. Example 21-84 shows the data collected from router R2.

Example 21-84 *Troubleshooting Data Collection on Router R2*

```

R2#show ipv6 ospf neighbor

R2#debug ipv6 ospf adj
  OSPFv3 adjacency events debugging is on

```

```

R2#u all
All possible debugging has been turned off
R2#show run
...OUTPUT OMITTED...
hostname R2
!
ipv6 unicast-routing
ipv6 cef
!
interface Loopback0
  ip address 10.2.2.2 255.255.255.255
!
interface FastEthernet0/0
ip address 192.168.0.22 255.255.255.0
  ipv6 address B:B:B:B::22/64
  ipv6 ospf hello-interval 60
  ipv6 ospf 1 area 1
!
interface Serial1/0
  no ip address
  encapsulation frame-relay
!
interface Serial1/0.1 point-to-point
  ip address 172.16.1.2 255.255.255.252
  ipv6 address C:C:C:C::2/64
  ipv6 ospf 1 area 0
  frame-relay interface-dlci 181
!
interface Serial1/0.2 point-to-point
  ip address 172.16.2.1 255.255.255.252
  ipv6 address D:D:D:D::1/64
  ipv6 ospf network point-to-multipoint
  ipv6 ospf 1 area 0
  frame-relay interface-dlci 182
!
ipv6 router ospf 1
  passive-interface default
!
...OUTPUT OMITTED...

```

Based on the output provided in Examples 21-83 and 21-84, hypothesize why routers R1 and R2 are not forming an adjacency. On a separate sheet of paper, write out your suggested solution to correct the issue.

Issue 1: Suggested Solution

Notice in Example 21-84 that router R2's hello timer on the Fast Ethernet 0/0 interface was set to a nondefault value, whereas the other end of the link was still set to the default. Also, router R2 had its OSPFv3 process configured with the **passive-interface default** command, which prevented any of router R2's interfaces from forming OSPFv3 adjacencies. Example 21-85 shows the correction of these configuration issues on router R2.

Example 21-85 *Correcting Router R2's Hello Timer and Passive-Interface Configuration*

```
R2#conf term
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#int fa 0/0
R2(config-if)#no ipv6 ospf hello-interval 60
R2(config-if)#exit
R2(config)#ipv6 router ospf 1
R2(config-rtr)#no passive-interface default
```

Issue 2: Adjacency Between Routers R2 and BB2

After implementing the fix shown in Example 21-85, router R2 successfully forms OSPF adjacencies with routers R1 and BB1. However, an adjacency is not successfully formed with router BB2. Example 21-86 shows the output of the **show ipv6 ospf interface s1/0.2** command issued on router BB2. This command was issued to view the OSPFv3 configuration of router BB2's Serial 1/0.2 subinterface, which is the subinterface used to connect to router R2. You compare this to R2's OSPF configuration on Serial 1/0.2 in Example 21-84.

Example 21-86 *Viewing Router BB2's OSPFv3 Configuration on Subinterface Serial 1/0.2*

```
BB2#show ipv6 ospf interface s1/0.2
Serial1/0.2 is up, line protocol is up
Link Local Address FE80::C200:8FF:FE2C:0, Interface ID 14
Area 0, Process ID 1, Instance ID 0, Router ID 10.4.4.4
Network Type POINT_TO_POINT, Cost: 64
Transmit Delay is 1 sec, State POINT_TO_POINT,
...OUTPUT OMITTED...

BB2#show ipv6 ospf neighbor
```

Based on router R2's configuration (shown in Example 21-84) and the output shown in Example 21-86, determine why an OSPF adjacency is not being formed between routers R2 and BB2. Again, on a separate sheet of paper, write out your suggested solution.

Issue 2: Suggested Solution

Router R2's OSPF network type on subinterface Serial 1/0.2 was set to point-to-multi-point; the other end of the link was the default network type of point-to-point. Example 21-87 shows the correction of router R2's misconfiguration.

Example 21-87 *Correcting Router R2's OSPF Network Type*

```
R2#conf term
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#int s1/0.2
R2(config-subif)#no ipv6 ospf network point-to-multipoint
R2(config-subif)#exit
```

At this point in the troubleshooting process, routers R1 and R2 have formed adjacencies. In addition, router R2 has formed adjacencies with routers BB1 and BB2. The output in Example 21-88 confirms the establishment of these adjacencies.

Example 21-88 *Confirming Router R2's OSPF Adjacencies*

```
R2#show ipv6 ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Interface ID	Interface
10.4.4.4	1	FULL/-	00:00:36	14	Serial1/0.2
10.3.3.3	1	FULL/-	00:00:36	14	Serial1/0.1
192.168.1.11	1	FULL/DR	00:00:39	4	FastEthernet0/0

Issue 3: Adjacency Between Routers BB1 and BB2

As shown in the output provided in Example 21-89, router BB1 has formed an adjacency with router BB2 over router BB1's Fast Ethernet 0/0 interface. However, an adjacency has not been successfully formed with router BB2 over router BB1's Serial 1/0.1 subinterface.

Example 21-89 *Determining Router BB1's Adjacencies*

```
BB1#show ipv6 ospf neigh
```

Neighbor ID	Pri	State	Dead Time	Interface ID	Interface
10.2.2.2	1	FULL/-	00:00:37	13	Serial1/0.2
10.4.4.4	1	DOWN/-	-	13	Serial1/0.1
10.4.4.4	1	FULL/DR	00:00:34	4	FastEthernet0/0

To investigate why an OSPF adjacency is not forming with router BB2 via router BB1's Serial 1/0.1 subinterface, the `debug ipv6 ospf adj` and `debug ipv6 ospf hello` commands were issued on router BB1, as shown in Example 21-90.

Example 21-90 *Debugging OSPFv3 Adjacency and Hello Events on Router BB1*

```

BB1#debug ipv6 ospf adj
    OSPFv3 adjacency events debugging is on
BB1#debug ipv6 ospf hello
    OSPFv3 hello events debugging is on
BB1#
*Mar 1 00:19:24.707: OSPFv3: Rcv DBD from 10.4.4.4 on Serial1/0.1 seq 0x1AEF opt
    0x0013 flag 0x7 len 28 mtu 1500 state EXSTART
*Mar 1 00:19:24.707: OSPFv3: Nbr 10.4.4.4 has larger interface MTU
*Mar 1 00:19:25.015: OSPFv3: Rcv hello from 10.2.2.2 area 0 from Serial1/0.2
    FE80::C201:8FF:FE2C:0 interface ID 13
*Mar 1 00:19:25.019: OSPFv3: End of hello processing
*Mar 1 00:19:28.583: OSPFv3: Send hello to FF02::5 area 0 on Serial1/0.2 from
    FE80::C202:8FF:FE98:0 interface ID 14
*Mar 1 00:19:28.647: OSPFv3: Rcv hello from 10.4.4.4 area 0 from Serial1/0.1
    FE80::C200:8FF:FE2C:0 interface ID 13
*Mar 1 00:19:28.651: OSPFv3: End of hello processing
*Mar 1 00:19:28.983: OSPFv3: Send hello to FF02::5 area 0 on FastEthernet0/0
    from FE80::C202:8FF:FE98:0 interface ID 4
*Mar 1 00:19:29.215: OSPFv3: Rcv hello from 10.4.4.4 area 0 from FastEthernet0/0
    FE80::C200:8FF:FE2C:0 interface ID 4
*Mar 1 00:19:29.219: OSPFv3: End of hello processing
BB1# u all
All possible debugging has been turned off

```

Because your troubleshooting on router BB1 is focused on BB1's Serial 1/0.1 subinterface, the **show ipv6 interface s1/0.1** command was issued, the output for which appears in Example 21-91.

Example 21-91 *Viewing the IPv6 Configuration on Router BB1's Serial 1/0.1 Subinterface*

```

BB1#show ipv6 interface s1/0.1
Serial1/0.1 is up, line protocol is up
    IPv6 is enabled, link-local address is FE80::C202:8FF:FE98:0
    Global unicast address(es):
        E::E::1, subnet is E::E::/64
    Joined group address(es):
        FF02::1
        FF02::2
        FF02::5
        FF02::1:FF00:1
        FF02::1:FF98:0
    MTU is 1400 bytes
    ICMP error messages limited to one every 100 milliseconds
    ICMP redirects are enabled

```

```
ND DAD is enabled, number of DAD attempts: 1
ND reachable time is 30000 milliseconds
Hosts use stateless autoconfig for addresses.
```

Based on the output provided in Examples 21-90 and 21-91, determine why router BB1 is failing to form an OSPF adjacency with router BB2, via router BB1's Serial 1/0.1 subinterface. On a separate sheet of paper, write out your proposed solution to this issue.

Issue 3: Suggested Solution

The **debug** output shown in Example 21-90 indicates that router BB1's neighbor (that is, 10.4.4.4) reachable over subinterface Serial 1/0.1 has a larger maximum transmission unit (MTU) than router BB1's Serial 1/0.1 subinterface. The output in Example 21-91 indicates that router BB1's Serial 1/0.1 subinterface has an MTU of 1400 bytes. This is less than the default value of 1500 bytes for this router. Example 21-92 shows how this MTU value was reset to its default value.

Example 21-92 *Correcting the MTU on Router BB1's Serial 1/0.1 Subinterface*

```
BB1#conf term
Enter configuration commands, one per line. End with CNTL/Z.
BB1(config)#int s1/0.1
BB1(config-subif)#ipv6 mtu 1500
*Mar 1 00:20:00.019: %OSPFv3-5-ADJCHG: Process 1, Nbr 10.4.4.4 on Serial1/0.1 from
LOADING to FULL, Loading Done
BB1(config-subif)#end
```

Notice, in Example 21-92, that an adjacency with router BB2 (that is, 10.4.4.4) was formed over router BB1's Serial 1/0.1 subinterface after setting the subinterface's MTU size to the default of 1500 bytes. Examples 21-93 and 21-94 further confirm that routers BB1 and BB2 have formed all appropriate adjacencies with their OSPF neighbors.

Example 21-93 *Router BB1's OSPF Adjacencies*

```
BB1#show ipv6 ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Interface ID	Interface
10.2.2.2	1	FULL/-	00:00:37	13	Serial1/0.2
10.4.4.4	1	FULL/-	00:00:30	13	Serial1/0.1
10.4.4.4	1	FULL/DR	00:00:31	4	FastEthernet0/0

Example 21-94 *Router BB2's OSPF Adjacencies*

```
BB2#show ipv6 ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Interface ID	Interface
10.2.2.2	1	FULL/-	00:00:37	14	Serial1/0.2
10.3.3.3	1	FULL/-	00:00:37	13	Serial1/0.1
10.3.3.3	1	FULL/BDR	00:00:31	4	FastEthernet0/0

To confirm that full reachability has been restored in the network, a series of **ping** commands were issued from router BB2, with one ping to each router in the topology. As shown in Example 21-95, all the pings were successful.

Example 21-95 *Confirming Reachability to All Routers*

```
BB2#ping a:a:a:a::11
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to A:A:A:A::11, timeout is 2 seconds:
!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 88/124/164 ms
BB2#ping b:b:b:b::22
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to B:B:B:B::22, timeout is 2 seconds:
!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 44/83/164 ms
BB2#ping f:f:f:f::1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to F:F:F:F::1, timeout is 2 seconds:
!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 40/79/128 ms
BB2#ping e:e:e:e::2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to E:E:E:E::2, timeout is 2 seconds:
!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/0/0 ms
```

Trouble Ticket 10

You receive the following trouble ticket for your RIPng domain:

Branch site A (that is, routers R1 and R2) has two connections to HQ. The HQ routers are BB1 and BB2. However, router R2 only sees a single path for a default route (rather than one path from each HQ router) in its IPv6 routing table. Also, router R2 is seeing other HQ advertised routes (specifically, E:E:E:E::/64 and F:F:F:F::/64) rather than just a default route in its IPv6 routing table. All routes that router R2 receives from the HQ routers, except a default route, should be suppressed.

This trouble ticket references the topology shown in Figure 21-10.

The **show ipv6 route** command was issued on router R2 to confirm that the IPv6 routing table included only a single path to reach the default network of ::/0. Example 21-95 provides the output from this command, which also confirms the presence of the routes E:E:E:E::/64 and F:F:F:F::/64 in the IPv6 routing table.

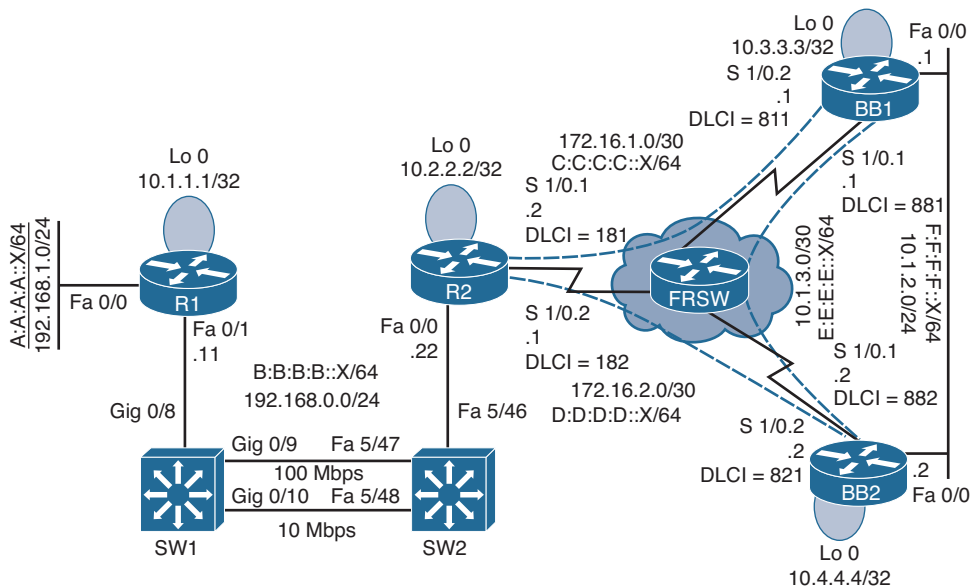


Figure 21-10 *Trouble Ticket 10 Topology*

Example 21-95 *Confirmation of Troubleshooting Issues on Router R2*

```
R2#show ipv6 route
IPv6 Routing Table - 12 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
R   ::/0 [120/2]
    via FE80::C200:EFF:FE64:0, Serial1/0.2
R   A:A:A:A::/64 [120/2]
    via FE80::209:B7FF:FEFA:D1E1, FastEthernet0/0
C   B:B:B:B::/64 [0/0]
    via ::, FastEthernet0/0
L   B:B:B:B::22/128 [0/0]
    via ::, FastEthernet0/0
C   C:C:C:C::/64 [0/0]
    via ::, Serial1/0.1
L   C:C:C:C::2/128 [0/0]
    via ::, Serial1/0.1
C   D:D:D:D::/64 [0/0]
    via ::, Serial1/0.2
L   D:D:D:D::1/128 [0/0]
    via ::, Serial1/0.2
R   E:E:E:E::/64 [120/2]
    via FE80::C200:EFF:FE64:0, Serial1/0.2
```

```

R2#show ipv6 rip database
R   F:F:F:F::/64 [120/2]
    via FE80::C200:EFF:FE64:0, Serial1/0.2
L   FE80::/10 [0/0]
    via ::, Null0
L   FF00::/8 [0/0]
    via ::, Null0

```

The **show ipv6 rip database** command, as shown in Example 21-96, proves that router R2 received two default route advertisements; however, only one of those route advertisements was injected into the IPv6 routing table.

Example 21-96 RIP Database on Router R2

```

R2#show ipv6 rip database
RIP process "PROCESS1", local RIB
A:A:A:A::/64, metric 2, installed
    FastEthernet0/0/FE80::209:B7FF:FEFA:D1E1, expires in 174 secs
B:B:B:B::/64, metric 2
    FastEthernet0/0/FE80::209:B7FF:FEFA:D1E1, expires in 174 secs
D:D:D:D::/64, metric 2
    Serial1/0.2/FE80::C200:EFF:FE64:0, expires in 160 secs
E:E:E:E::/64, metric 2, installed
    Serial1/0.2/FE80::C200:EFF:FE64:0, expires in 160 secs
F:F:F:F::/64, metric 2, installed
    Serial1/0.2/FE80::C200:EFF:FE64:0, expires in 160 secs
::/0, metric 2, installed
    Serial1/0.2/FE80::C200:EFF:FE64:0, expires in 160 secs
    Serial1/0.1/FE80::C202:EFF:FEBC:0, expires in 170 secs

```

Example 21-97 shows the running configuration on router R2.

Example 21-97 Running Configuration on Router R2

```

R2#show run
...OUTPUT OMITTED...
hostname R2
!
ipv6 unicast-routing
ipv6 cef
!
interface Loopback0
 ip address 10.2.2.2 255.255.255.255
!
interface FastEthernet0/0
 ip address 192.168.0.22 255.255.255.0
 ipv6 address B:B:B:B::22/64
 ipv6 rip PROCESS1 enable
!

```

```

interface Serial1/0
  no ip address
  encapsulation frame-relay
  serial restart-delay 0
!
interface Serial1/0.1 point-to-point
  ip address 172.16.1.2 255.255.255.252
  ipv6 address C:C:C:C::2/64
  ipv6 rip PROCESS1 enable
  frame-relay interface-dlci 181
!
interface Serial1/0.2 point-to-point
  ip address 172.16.2.1 255.255.255.252
  ipv6 address D:D:D:D::1/64
  ipv6 rip PROCESS1 enable
  frame-relay interface-dlci 182
!
ipv6 router rip PROCESS1
  maximum-paths 1
!
...OUTPUT OMITTED...

```

Issue 1: Router R2 Not Load Balancing Between Routers BB1 and BB2

The first issue you investigate is router R2 not load balancing between the HQ routers (that is, routers BB1 and BB2). Based on the **show** command output presented in Examples 21-95, 21-96, and 21-97, hypothesize why router R2's IPv6 routing table contains only a single entry for a default network (rather than having two entries, one for BB1 and one for BB2). On a separate sheet of paper, write out your proposed configuration change to resolve this issue.

Issue 1: Suggested Solution

A review of router R2's running configuration reveals the **maximum-paths 1** command in router configuration mode for the RIPng routing process. This command prevents two default route paths from appearing in router R2's IPv6 routing table. Example 21-98 shows how this command is removed from router R2's configuration to restore load balancing.

Example 21-98 *Restoring Load Balancing on Router R2*

```

R2#conf term
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#ipv6 router rip PROCESS1
R2(config-rtr)#no maximum-paths 1

```

Issue 2: Backbone Routes Not Being Suppressed

The second issue you investigate is about the specific routes (that is, E:E:E:E::/64 and F:F:F:F::/64) being advertized to Branch site A (R1 and R2). The goal is to only advertise default route information into Branch site A.

The **debug ipv6 rip** command was issued on router R2 to see if router BB2 was sending both default route information and specific route information. The output from this command, as presented in Example 21-99, confirms that router BB2 is not suppressing specific route information.

Example 21-99 Debugging RIPng Traffic on Router R2

```
R2#debug ipv6 rip
...OUTPUT OMITTED...
*Mar 1 00:33:30.747: RIPng: response received from FE80::C200:EFF:FE64:0 on
Serial1/0.2 for PROCESS1
*Mar 1 00:33:30.751:      src=FE80::C200:EFF:FE64:0 (Serial1/0.2)
*Mar 1 00:33:30.751:      dst=FF02::9
*Mar 1 00:33:30.755:      sport=521, dport=521, length=92
*Mar 1 00:33:30.755:      command=2, version=1, mbz=0, #rte=4
*Mar 1 00:33:30.755:      tag=0, metric=1, prefix=F:F:F:F::/64
*Mar 1 00:33:30.755:      tag=0, metric=1, prefix=E:E:E:E::/64
*Mar 1 00:33:30.755:      tag=0, metric=1, prefix=D:D:D:D::/64
*Mar 1 00:33:30.755:      tag=0, metric=1, prefix=::/0
...OUTPUT OMITTED...
```

Examples 21-100 and 21-101 show the RIP next generation (RIPng) configuration of the Serial 1/0.2 subinterface on routers BB1 and BB2. The Serial 1/0.2 subinterface on each router is the subinterface connecting to router R2.

Example 21-100 Viewing the RIPng Configuration on Router BB1's Serial 1/0.2 Subinterface

```
BB1#show run | begin Serial1/0.2
interface Serial1/0.2 point-to-point
 ip address 172.16.1.1 255.255.255.252
 ipv6 address C:C:C:C::1/64
 ipv6 rip PROCESS1 enable
 ipv6 rip PROCESS1 default-information only
 frame-relay interface-dlci 811
```

Example 21-101 Viewing the RIPng Configuration on Router BB2's Serial 1/0.2 Subinterface

```
BB2#show run | begin Serial1/0.2
interface Serial1/0.2 point-to-point
 ip address 172.16.2.2 255.255.255.252
 ipv6 address D:D:D:D::2/64
```

```

ipv6 rip PROCESS1 enable
ipv6 rip PROCESS1 default-information originate
frame-relay interface-dlci 821

```

Based on the **debug** and **show** commands output presented in Examples 21-99, 21-100, and 21-101, hypothesize why router R2 is receiving specific route information for networks E:E:E:E::/64 and F:F:F:F::/64. On a separate sheet of paper, write out your proposed configuration change to resolve this issue.

Issue 2: Suggested Solution

An inspection of router BB2's running configuration reveals the **ipv6 rip PROCESS1 default-information originate** command under subinterface configuration mode for Serial 1/0.2. The **originate** keyword in this command sources a default router advertisement, but it does not suppress the sending of more specific routes. Example 21-102 shows how this configuration was changed to use the **only** parameter. The **only** parameter causes the interface to only originate default route information, while suppressing more specific routes.

Example 21-102 *Suppressing Specific Route Information on Router BB2's Serial Interface*

```

BB2#conf term
Enter configuration commands, one per line. End with CNTL/Z.
BB2(config)#int s1/0.2
BB2(config-subif)#ipv6 rip PROCESS1 default-information only

```

After giving the E:E:E:E::/64 and F:F:F:F::/64 routes sufficient time to time out of router R2's IPv6 routing table, the **show ipv6 route** was once again issued. The output, as shown in Example 21-103, confirms that the issues reported in the trouble ticket are resolved. Specifically, router R2 sees two paths across which it can load balance to reach a default route. Also, specific routes (that is, E:E:E:E::/64 and F:F:F:F::/64) do not appear in router R2's IPv6 routing table.

Example 21-103 *Router R2's IPv6 Routing Table After Troubleshooting*

```

R2#show ipv6 route
IPv6 Routing Table - 10 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
R   ::/0 [120/2]
    via FE80::C200:EFF:FE64:0, Serial1/0.2
    via FE80::C202:EFF:FEBC:0, Serial1/0.1
R   A:A:A:A::/64 [120/2]
    via FE80::209:B7FF:FEFA:D1E1, FastEthernet0/0

```



```
C   B::B::B::/64 [0/0]
    via ::, FastEthernet0/0
L   B::B::B::22/128 [0/0]
    via ::, FastEthernet0/0
C   C::C::C::/64 [0/0]
    via ::, Serial1/0.1
L   C::C::C::2/128 [0/0]
    via ::, Serial1/0.1
C   D::D::D::/64 [0/0]
    via ::, Serial1/0.2
L   D::D::D::1/128 [0/0]
    via ::, Serial1/0.2
L   FE80::/10 [0/0]
    via ::, Null0
L   FF00::/8 [0/0]
    via ::, Null0
```

This page intentionally left blank



<https://t.me/learningnets>

Final Preparation

The first two chapters of this book introduced you to a structured troubleshooting process and the different tools that can assist you during the different steps of the process. Chapters 3 through 20 covered the issues that may arise with the different technologies, protocols, and features deployed within a network. They also provide troubleshooting approach examples using various **show** and **debug** commands, and focus on how you can identify the cause of the issues and fix them. Chapter 21 provides an additional set of sample trouble tickets to give you more exposure to troubleshooting.

Although these chapters supply the detailed information needed to prepare you for the 300-135 TSHOOT exam, most people need more preparation than simply reading the chapters of this book. This chapter details a set of tools and a study plan to help you complete your preparation for the exam. It has two sections. The first section lists the exam preparation tools useful at this point in the study process. The second section details a suggested study plan now that you have completed all the preceding chapters in this book.

Note The Glossary and Appendixes C, D, and E exist as soft-copy appendixes on the CD included in the back of this book.

Tools for Final Preparation

This section lists additional information about exam preparation tools and how to access the tools.

Exam Engine and Questions on the CD

The CD in the back of the book includes the Pearson Cert Practice Test (PCPT) engine. This software presents you with a set of multiple-choice questions, covering the topics you will be challenged with on the real exam. The PCPT engine lets you study the exam content (using study mode) or take a simulated exam (in practice exam mode).

The CD in the back of the book contains the exam engine. Once installed, you can then activate and download the current TSHOOT exam from Pearson's website. Installation of the exam engine takes place in two steps:

- Step 1.** Install the exam engine from the CD.
- Step 2.** Activate and download the TSHOOT practice exam.

Install the Exam Engine

The software installation process is routine as compared with other software installation processes. To be complete, the following steps outline the installation process:

- Step 1.** Insert the CD into your PC.
- Step 2.** The software that automatically runs is the Cisco Press software to access and use all CD-based features, including the exam engine and the CD-only appendices. From the main menu, click the **Install the Exam Engine** option.
- Step 3.** Respond to prompts as with any typical software installation process.

The installation process gives you the option to activate your exam with the activation code supplied on the paper in the CD sleeve. This process requires that you establish a Pearson website login. You need this login to activate the exam. Therefore, please do register when prompted. If you already have a Pearson website login, you do not need to register again. Just use your existing login.

Activate and Download the Practice Exam

Once the exam engine is installed, you should then activate the exam associated with this book (if you did not do so during the installation process), as follows:

- Step 1.** Start the Pearson Cert Practice Test (PCPT) software.
- Step 2.** To activate and download the exam associated with this book, from the My Products or Tools tab, click the **Activate** button.
- Step 3.** At the next screen, enter the activation key from the paper inside the cardboard CD holder in the back of the book. Once entered, click the **Activate** button.
- Step 4.** The activation process will download the practice exam. Click **Next**, and then click **Finish**.

Once the activation process is completed, the **My Products** tab should list your new exam. If you do not see the exam, make sure that you selected the **My Products** tab on the menu. At this point, the software and practice exam are ready to use. Simply select the exam, and click the **Use** button.

To update a particular exam you have already activated and downloaded, simply select the **Tools** tab, and select the **Update Products** button. Updating your exams will ensure you have the latest changes and updates to the exam data.

If you want to check for updates to the Pearson Cert Practice Test exam engine software, simply select the **Tools** tab, and click the **Update Application** button. This will ensure you are running the latest version of the software engine.

Activating Other Exams

The exam software installation process, and the registration process, only has to happen once. Then, for each new exam, only a few steps are required. For instance, if you buy another new Cisco Press Official Cert Guide or Pearson IT Certification Cert Guide, remove the activation code from the CD sleeve in the back of that book; you do not even need the CD at this point. From there, all you have to do is start the exam engine (if not still up and running), and perform Steps 2 through 4 from the previous list.

Premium Edition

In addition to the free practice exam provided on the CD-ROM, you can purchase additional exams with expanded functionality directly from Pearson IT Certification. The Premium Edition of this title contains an additional two full practice exams as well as an eBook (in both PDF and ePub format). In addition, the premium edition title also has remediation for each question to the specific part of the eBook that relates to that question.

Because you have purchased the print version of this title, you can purchase the Premium Edition at a deep discount. There is a coupon code in the CD sleeve that contains a one-time use code, as well as instructions for where you can purchase the Premium Edition.

To view the premium edition product page, go to www.ciscopress.com/title/9780133414363.

The Cisco Learning Network

Cisco provides a wide variety of CCNP Routing and Switching preparation tools at a Cisco website called the Cisco Learning Network. Resources found here include sample questions, forums on each Cisco exam, learning video games, and information about each exam.

To reach the Cisco Learning Network, go to learningnetwork.cisco.com, or just search for “Cisco Learning Network.” To access some of the features/resources, you need to use the login you created at Cisco.com. If you don’t have such a login, you can register for free. To register, simply go to Cisco.com; click **Register** at the top of the page; and supply some information.

Memory Tables

Like most certification guides from Cisco Press, this book purposefully organizes information into tables and lists for easier study and review. Rereading these tables can be very useful before the exam. However, it is easy to skim over the tables without paying attention to every detail, especially when you remember having seen the table’s contents when reading the chapter.

Instead of simply reading the tables in the various chapters, this book’s Appendixes B and C give you another review tool. Appendix C, “Memory Tables,” lists partially completed

versions of many of the tables from the book. You can open Appendix C (a PDF on the CD that comes with this book) and print the appendix. For review, you can attempt to complete the tables. This exercise can help you focus during your review. It also exercises the memory connectors in your brain; plus it makes you think about the information without as much information, which forces a little more contemplation about the facts.

Appendix D, “Memory Tables Answer Key,” also a PDF located on the CD, lists the completed tables to check yourself. You can also just refer to the tables as printed in the book.

Chapter-Ending Review Tools

Chapters 1 through 20 each have several features in the “Exam Preparation Tasks” section at the end of the chapter. You may have used some of or all these tools at the end of each chapter. It can also be useful to use these tools again as you make your final preparations for the exam.

Suggested Plan for Final Review/Study

This section lists a suggested study plan from the point at which you finish reading through Chapter 21 until you take the TSHOOT exam. Certainly, you can ignore this plan; use it as is, or just take suggestions from it.

The plan uses six steps. If following the plan verbatim, you should proceed by part through the steps. That is, starting with Part I (Fundamental Troubleshooting and Maintenance Concepts), do the following six steps. Then, for Part II (Troubleshooting Cisco Catalyst Switch Features), do the following six steps, and so on. The steps are as follows:

- Step 1. Review key topics and DIKTA questions:** You can use the table that lists the key topics in each chapter, or just flip the pages looking for the Key Topic icons. Also, reviewing the DIKTA questions from the beginning of the chapter can be helpful for review.
- Step 2. Complete memory tables:** Open Appendix C on the CD and print the entire appendix, or print the tables by major part. Then complete the tables, and check your answers in Appendix D, which also appears on the CD.
- Step 3. Hands-on practice:** Most people practice CCNP configuration and verification before the exam. Whether you use real gear, a simulator, or an emulator, practice the configuration and verification commands.
- Step 4. Build troubleshooting checklists:** This is one of the most important things you can do. Start by glancing through the Table of Contents, or even the sections covered in a chapter. For each of the topics you see presented in a section, (for example, VLANs, or EIGRP adjacencies) create a listing (from memory) of all the issues that may arise and then write the solutions to the issues down. Then compare your issues and solutions to those presented in the chapters.

Step 5. Subnetting practice: If you can no longer do subnetting well and quickly without a subnetting calculator, take some time to get better and faster before going to take the TSHOOT exam.

Step 6. Use the exam engine to practice: The exam engine on the CD can be used to study using a bank of unique exam-realistic multiple-choice questions available only with this book.

The rest of this section describes Steps 1, 3, 5, and 6 for which a little more explanation might be helpful.

Step 1: Review Key Topics and DIKTA Questions

This review step focuses on the core facts related to the TSHOOT exam. The exam certainly covers other topics as well, but the DIKTA questions and the Key Topic items attempt to focus attention on the more important topics in each chapter.

As a reminder, if you follow this plan after reading the first 20 chapters, working a major part at a time helps you pull each major topic together.

Step 3: Hands-On Practice

Although this book gives you many troubleshooting checklists, specific configuration examples, examples of output, and explanations for the meaning of that output, there is no substitute for hands-on practice. This short section provides a few suggestions regarding your efforts to practice from the command-line interface (CLI).

First, most people use one or more of the following options for hands-on skills:

- **Real gear:** Either purchased (often used), borrowed, or rented
- **Simulators:** Software that acts like real gear
- **Emulators:** Software that emulates Cisco hardware and runs Cisco IOS

For real gear, the minimum recommended home lab configuration would have three ISR (or ISR2) routers running Cisco IOS 15.2 (or later) and 3 Catalyst switches running IOS 15.0 (or later). One switch should be a 3750 or 3560, the others can be 2960s. This would allow you to experiment with most of the routing and switching technologies discussed in this book.

Pearson IT Certification offers an excellent simulator with nearly 400 structured labs to help you get hands-on experience. Even though the simulator targets the CCNA exam, many of its labs are appropriate for your TSHOOT studies as they will solidify the topics and concepts of routing and switching. You can learn more about the “CCNA Routing and Switching 200-120 Network Simulator” at <http://bit.ly/ccnasimulator>.

As for emulators, you can purchase access to emulated routers from the Cisco Learning Network. What you are purchasing is a block of hours to access the emulated gear, along with structured labs to follow. The product is called *Cisco Learning Labs*, and you can find more information at <http://bit.ly/route-emulator>.

Step 5: Subnetting Practice

This book assumes that you have mastered subnetting and the related math. However, many people who progress through CCNA, and move on to CCNP, follow a path like this:

- Step 1.** Learn subnetting conceptually.
- Step 2.** Get really good at doing the math quickly.
- Step 3.** Pass CCNA.
- Step 4.** Do not practice regularly and therefore become a lot slower at doing the subnetting math.
- Step 5.** Study for CCNP ROUTE, then SWITCH, then TSHOOT.

Although subnetting should not be assessed as an end to itself on CCNP TSHOOT, it may be required that you understand subnetting math and do that math just as quickly as you did when you passed CCNA. If you are a little slow on doing subnetting math, before you go to the TSHOOT exam, try some of the following exercises:

- Practice finding the subnet number, broadcast address, and range of addresses in a subnet. To do so, pick a network address and mask; calculate the values; and use your favorite subnet calculator to check your work.
- Use the Cisco Subnetting Game, also at the Cisco Learning Network. You can find it at <http://bit.ly/subnet-game>.
- Practice choosing the best summary route for a range of subnets. Pick three or four addresses/masks. Calculate the subnet number and range. Then, try to choose the summary (subnet number/mask) that includes those three or four subnets, without including any more subnets than what is required. You can check your math with a subnet calculator.

If you like performing binary/decimal conversions when you work through these problems, but just need to go faster, check out the Cisco Binary game, also at the Cisco Learning Network. You can find it at <http://bit.ly/binary-game>.

Step 6: Use the Exam Engine

The PCPT engine on the CD lets you access a database of questions created specifically for this book. The PCPT engine can be used either in *study mode* or *practice exam mode*, as follows:

- **Study mode:** Study mode is most useful when you want to use the questions for learning and practicing. In study mode, you can select options like randomizing the order of the questions and answers, automatically viewing answers to the questions as you go, testing on specific topics, and many other options.

- **Practice Exam mode:** This mode presents questions in a timed environment, providing you with a more exam realistic experience. It also restricts your ability to see your score as you progress through the exam and view answers to questions as you are taking the exam. These timed exams not only allow you to study for the actual 300-135 TSHOOT exam, they also help you simulate the time pressure that can occur on the actual exam.

When doing your final preparation, you can use study mode, practice exam mode, or both. However, after you have seen each question a couple of times, you will likely start to remember the questions, and the usefulness of the exam database may go down. So, consider the following options when using the exam engine:

- Use the question database for review. Use study mode to study the questions by chapter, just as with the other final review steps listed in this chapter. Consider upgrading to the Premium Edition of this book if you want to take additional simulated exams.
- Save the question database, not using it for review during your review of each book part. Save it until the end; so you will not have seen the questions before. Then, use practice exam mode to simulate the exam.

Picking the correct mode from the exam engine's user interface is pretty obvious. The following steps show how to move to the screen from which you can select the study or practice exam mode:

The steps are as follows:

- Step 1.** Click the **My Products** tab if you are not already in that screen.
- Step 2.** Select the exam you wish to use from the list of available exams.
- Step 3.** Click the **Use** button.

By taking these actions, the engine should display a window from which you can choose **Study Mode** or **Practice Exam Mode**. When in study mode, you can further choose the book chapters, limiting the questions to those explained in the specified chapters of the book.

Summary

The tools and suggestions listed in this chapter have been designed with one goal in mind: to help you develop the skills required to pass the TSHOOT exam. This book has been developed from the beginning to not just tell you how to troubleshoot, but also help you learn how to successfully apply your troubleshooting efforts. No matter what your experience level is leading up to when you take the exam, it is my hope that the broad range of preparation tools, and even the structure of the book, can help you pass the exam with ease. I wish you all the best in your studies and on your exam.



<https://t.me/learningnets>

Answers to the “Do I Know This Already” Quizzes

Chapter 1

1. B, C, and D
2. D
3. B
4. B and D
5. A
6. B
7. B
8. D
9. C
10. A, C, and D
11. B and C
12. C
13. D
14. A
15. C and D
16. A, B, and D
17. B
18. A, C, and D
19. A, B, and D
20. A and C

Chapter 2

1. A, C, and D
2. A, B, and D
3. C
4. B
5. B
6. A and D
7. C
8. A
9. C
10. B
11. C
12. C

Chapter 3

1. B and C
2. B and C
3. A, B, and D
4. B
5. B
6. D
7. D
8. A and B

Chapter 4

1. C
2. D
3. D
4. A and B
5. C and D
6. C
7. C
8. A and B
9. D
10. A and B

Chapter 5

1. C
2. A
3. B
4. C
5. A and C
6. B and D
7. C
8. B
9. A, B, and C
10. C

Chapter 6

1. A
2. D
3. A and C
4. B
5. C

6. A

7. C

8. C

9. D

10. C

Chapter 7

1. B
2. B and C
3. B and C
4. A and B
5. B
6. A and B
7. C
8. B
9. C and D
10. D

Chapter 8

1. B
2. A
3. C
4. B and D
5. C
6. A and C
7. C and D
8. D
9. B
10. A and C

Chapter 9

1. B and D
2. A and C
3. A
4. B
5. D
6. B
7. C
8. C
9. B
10. B

Chapter 10

1. C
2. C
3. B and C
4. A
5. A, B, and C
6. C
7. D
8. C
9. C
10. C

Chapter 11

1. D
2. C
3. B
4. C
5. B
6. B and D

7. D

8. D

9. A

10. A

Chapter 12

1. A and B
2. C and D
3. C
4. A, B, and C
5. D
6. B
7. B
8. C
9. D
10. C
11. B
12. A
13. B and D

Chapter 13

1. D
2. C
3. A
4. B and C
5. C
6. D
7. A and C
8. A
9. A
10. A, B, and C

Chapter 14

1. B
2. A, B, and D
3. A
4. C
5. A, B, and C
6. C
7. A
8. A
9. A
10. C
11. A and B

Chapter 15

1. A, B, and C
2. C and D
3. B
4. B
5. A
6. C
7. B
8. D
9. B and C
10. A

Chapter 16

1. B
2. A
3. B
4. C

5. D

6. A

7. A

Chapter 17

1. C
2. A and B
3. C
4. D
5. A
6. A
7. C
8. C
9. D
10. A, B, and C

Chapter 18

1. C and D
2. C
3. B and C
4. C
5. B
6. B, C, and D
7. B
8. C
9. A
10. A and D
11. A
12. A and C
13. A

Chapter 19

1. C
2. D
3. D
4. C
5. A
6. B and D
7. B
8. A and D
9. C and D
10. A, C, and D

Chapter 20

1. A and C
2. C
3. A
4. B
5. A
6. B
7. D
8. B
9. D
10. B



<https://t.me/learningnets>

TSHOOT Exam Updates

Over time, reader feedback allows Cisco Press to gauge which topics give our readers the most problems when taking the exams. To assist readers with those topics, the authors create new materials clarifying and expanding on those troublesome exam topics. As mentioned in the Introduction, the additional content about the exam is contained in a PDF document on this book's companion website, at <http://www.ciscopress.com/title/9781587205613>.

This appendix is intended to provide you with updated information if Cisco makes minor modifications to the exam upon which this book is based. When Cisco releases an entirely new exam, the changes are usually too extensive to provide in a simple update appendix. In those cases, you might need to consult the new edition of the book for the updated content.

This appendix attempts to fill the void that occurs with any print book. In particular, this appendix does the following:

- Mentions technical items that might not have been mentioned elsewhere in the book
- Covers new topics if Cisco adds new content to the exam over time
- Provides a way to get up-to-the-minute current information about content for the exam

Always Get the Latest at the Companion Website

You are reading the version of this appendix that was available when your book was printed. However, given that the main purpose of this appendix is to be a living, changing document, it is important that you look for the latest version online at the book's companion website. To do so, complete these steps:

- Step 1.** Browse to <http://www.ciscopress.com/title/9781587205613>.
- Step 2.** Select the **Appendix** option under the More Information box.
- Step 3.** Download the latest Appendix B document.

Note Note that the downloaded document has a version number. Comparing the version of the print Appendix B (Version 1.0) with the latest online version of this appendix, you should do the following:

- Same version: Ignore the PDF that you downloaded from the companion website.
- Website has a later version: Ignore this Appendix B in your book and read only the latest version that you downloaded from the companion website.

Technical Content

The current version of this appendix does not contain any additional technical coverage.

This page intentionally left blank

Index

Numbers

!H, 491

2Way, adjacency states, 591

802.1Q trunking, 33, 141

A

AAA (Cisco IOS), 858-861

access control, 273

protected ports, 273-275

PVLANs (private VLANs), 275-279

VACLs (VLAN access control lists),
279

access trunking mode, 143

verifying, 143

accounting management, FCAPS, 28

ACLs, 279

BGP (Border Gateway Protocol),
neighbor adjacencies, 759-761

EIGRP (Enhanced Interior Gateway
Routing Protocol)

IPv4, 527-528

IPv6, 564

IPv6 ACLs. *See* IPv6 ACLs

OSPFv2, 601-602

RIPng (RIP next generation), 496-497

RIPv2, 485

verifying, IPv4 ACLs, 406

activating PCPT (Pearson Cert
Practice Test), 944

active virtual forwarder. *See* AVF
(active virtual forwarder)

active virtual gateway. *See* AVG
(active virtual gateway)

AD (administrative distance), 437-438

adapting network maintenance mod-
els, 28-29

adding

HTTP server login credentials to
router configurations, 49

login credentials, FTP servers, 49

address families. *See* AFs (address
families)

addressing

IPv4 addressing. *See* IPv4 addressing

NAT (Network Address Translation).
See NAT (Network Address
Translation)

adjacency states, 591

adjacency tables, 431

administrative distance of route
sources, 437

advanced redistribution, 737

routing loops, 739-744

suboptimal routing, 737-739

advanced tools, 57

NetFlow, 57-61

SNMP (Simple Network Management
Protocol), 57-59

ADVERTISE, 384

AFs (address families), OSPFv3,
655-664

trouble tickets, 664-668

Align-Err, 98
allowed VLANs, trunks, 147-148
answers to quizzes. *See* Appendix A
archive configuration, confirming, 51
archived configurations, restoring, 53
area numbers, mismatched area numbers, OSPFv2, 596-597
area types, mismatched area types, OSPFv2, 597-598
ARP (Address Resolution Protocol), 371
ARP cache
 MAC address lookup, 440
 proxy ARP enabled, 442
ARP Input process, 107-108
ARP reply, 135-137
ARP requests, 133-135
ASBR (Autonomous System Boundary Router), redistribution, 712
assignments, IPv6 addressing, 375
 stateless address autoconfiguration/SLAAC, 375-380
attempt, adjacency states, 591
authentication
 EIGRP (Enhanced Interior Gateway Routing Protocol)
 IPv4, 525-527
 IPv6, 562-563
 mismatched authentication
 BGP (Border Gateway Protocol), 763-764
 OSPFv2, 600-601
 RIPv2, 477-479

automated backups, confirming, 51
automatic archive configuration, 50
automating documentation, 35
autonomous system numbers
 EIGRP (Enhanced Interior Gateway Routing Protocol)
 IPv4, 518-520
 IPv6, 562
 private autonomous system numbers, BGP path selection, 784
autosummarization, RIPv2, 482-483
AVF (active virtual forwarder), 318-323
AVG (active virtual gateway), 318

B

backing up router start up configuration
 to FTP servers, 48-53
 on FTP servers without specifying login credentials, 49
backplane, Cisco Catalyst switches, 96
bad or missing network statements, RIPv2, 470-471
baseline data, 15
baselines
 creating
 with NetFlow, 58
 with SNMP, 58
 establishing, 36

basic tools, 47

best path decision-making process, BGP (Border Gateway Protocol), 780-784

BGP (Border Gateway Protocol), 123-124, 748

Do I Know This Already? quizzes, 748-752

neighbor adjacencies, 753-754

ACLs, 759-761

BGP packets sourced from wrong IP address, 758-759

incorrect neighbor statement, 757-758

interface is down, 754

Layer 3 connectivity, 754-755

misconfigured peer groups, 764-765

mismatched authentication, 763-764

neighbor does not have a route to local router, 756-757

path to neighbor is via default route, 755-756

timers, 765-766

TTL of BGP packet expires, 761-763

redistribution, 715-718

trouble tickets, 791, 910-918

trouble ticket 18-1, 791-796

trouble ticket 18-2, 796-802

trouble ticket 18-3, 802-806

BGP for IPv6, 786-790

BGP packets sourced from wrong IP address, 758-759

BGP path selection, 780

best path decision-making process, 781-784

debug, 784-786

private autonomous system numbers, 784

BGP routes, 766-768

missing or bad network mask command, 768-770

next-hop router not reachable, 770-772

route filtering, 775-780

source information, 773-775

split horizon, 772-773

blocking nondesignated port (X), 177

Border Gateway Protocol (BGP). *See* BGP (Border Gateway Protocol)

bottom-up method, 21-22

BPDU Filter, 187-188

BPDU Guard, 184-187

branch, GRE tunnel configuration, 451

broadcast storms, STP (Spanning-Tree Protocol), 181-182

buffer leaks, 122-123

C

Carri-Sen, 98

CEF (Cisco Express Forwarding), 115-116, 431

change management, troubleshooting, 37-38

Cisco Catalyst switches

mismatched duplex settings, 99-101

port errors, 97-98

STP (Spanning-Tree Protocol) topology, 177

troubleshooting, 96-97

Cisco Express Forwarding (CEF), 115-116, 431

Cisco IOS, 64

AAA, 858-861

collect information, 68-69

filtering output of show commands, 69-73

- ping, 64-66
- telnet, 67
- traceroute, 67-68
- troubleshooting
 - hardware*, 74
 - high processor utilization*, 108-113
- Cisco IOS IP SLA, 827-833
- Cisco Learning Network, 945
- Cisco Lifecycle Services, 28
- Cisco Support tools, 64
- clear ip nat translation, 354
- CLI (command-line interface), 47
- CLI tools, 47-48
- collect information
 - Cisco IOS, 68-69
 - filtering output of show commands*, 69-73
 - STP (Spanning-Tree Protocol), 177
 - gathering STP information*, 177-179
 - MSTP (Multiple Spanning Tree Protocol)*, 179-180
 - structured troubleshooting, 14
 - in transit, 75
 - packet captures*, 75-76
 - RSPAN (Remote SPAN)*, 78-79
 - SPAN (Switched Port Analyzer)*, 76-78
- commands
 - network mask command, BGP (Border Gateway Protocol), 768-770
 - OSPFv3, IPv6, 641-647
- communication, troubleshooting, 36-37
- comparing
 - HSRP, VRRP, and GLBP, 330
 - running configuration and startup configuration before issuing the copy command, 52
 - trunking administrative modes, 145
- comparing configurations method, 23-24
- component swapping method, 24-25
- configuration archives, viewing, 50
- configuration changes, routine maintenance tasks, 29
- configuration information, documentation, 33
- configuration management, FCAPS, 28
- configuration merge, witnessing, 53
- configuring, routed ports, SW1, 237
- CONFIRM, 384
- confirming
 - archive configuration, 51
 - automated backups, 51
- console access, 854-855
- console line, 854
- control plane, Cisco Catalyst switches, 96
- converging after a failure, HSRP (Hot Standby Router Protocol), 291
- copy command, comparing running configuration and startup configuration, 52
- corrupt switches, troubleshooting, MAC address tables, 180-181
- COUNTER_RESET command, 63
- CPU utilization, excessive CPU utilization. *See* excessive CPU utilization

D

- DAI (dynamic ARP inspection), 267-268
- data structures, routing information sources, 436
- debug
 - BGP path selection, 784-786
 - HSRP (Hot Standby Router Protocol), 296-297
 - RIPng (RIP next generation), 494
 - verifying, PBR (policy-based routing), 691
- debug ip bgp, 785
- debug ip bgp updates, 786
- debug ip dhcp server events, 349
- debug ip dhcp server packet, 349-350
- debug ip nat, 355
- debug ip ospf adj, 601
- debug ip ospf hello, 598
- debug ip policy, 684
- debug ip policy output, 689
- debug ip rip, 469, 477
- debug ip routing, 785
- debug standby terse, 296
- DECLINE, 384
- default gateways, verifying, 380
- default port costs
 - STP (Spanning-Tree Protocol), 175
- default routes
 - OSPFv2, 627
 - RIPng (RIP next generation), 495-496
- default trunking mode on SW2, verifying, 145
- deny sequence, 679
- designated ports
 - STP (Spanning-Tree Protocol), 176
- destination routing protocol, 702
- device logs, 54
- device performance, quizzes, 92-95
- DHCP (Dynamic Host Configuration Protocol), 334
 - IPv4 addressing, 342
 - reviewing DHCP operations, 342-347*
 - troubleshooting issues, 347-348*
 - message types, 384
 - stateful DHCPv6, 381-382
 - stateless DHCPv6, 382-384
- DHCP (Dynamic Host Configuration Protocol) IPv4 addressing troubleshooting commands, 348-350
- DHCP snooping, 265-267
 - verifying, 266
- DHCP snooping bindings, verifying, 267
- DHCPv6, 384
- DHCPv6 relay agent, 385-386
- diagnosing problems, 10
- different subnets
 - EIGRP (Enhanced Interior Gateway Routing Protocol), IPv4, 524-526
 - OSPFv2, 598-599
- discontiguous areas, OSPFv2, 624-626
- discontiguous networks and autosummarization, EIGRP (Enhanced Interior Gateway Routing Protocol), IPv4 issues, 542-543
- displaying, OSPFv3 routes, 646
- distribute-list, EIGRP (Enhanced Interior Gateway Routing Protocol), 535
- divide-and-conquer method, 22
- Do I Know This Already? quizzes, 2-8, 128-131
 - addressing, 334-337
 - answers. *See* Appendix A

- BGP (Border Gateway Protocol), 748-752
- device performance, 92-95
- EIGRP (Enhanced Interior Gateway Routing Protocol), 512-516
- first-hop redundancy protocols, 286-289
- Inter-VLAN routing, 208-211
- IPv4 ACLs, IPv6 ACLs, prefix lists, 396-400
- IPv4/IPv6 routing and GRE tunnels, 422-426
- IPv6 addressing, 366-369
- maintenance tools, 40-44
- management access, 850-853
- management protocols, 814-817
- OSPF (Open Shortest Path First), 586-589
- redistribution, 696-699
- RIPng (RIP next generation), 462-465
- RIPv2, 462-465
- route maps and policy-based routing, 674-677
- STP (Spanning-Tree Protocol), 168-171
- switch security, 246-249
- do not fragment bit set, pinging, 65**
- documentation, 16**
 - automating, 35
 - maintaining current network documentation, 35
 - network maintenance, 32-33
- domain name mismatch, VTP domain name mismatch, 148-149**
- down, adjacency states, 591**
- downloading, PCPT (Pearson Cert Practice Test), 944**
- DP (designated port), 174**
- DR (designated router), verifying, 617-618**

- duplicate router IDs**
 - OSPFv2, 603-604
 - OSPFv2 routes, 619-620
- dynamic ARP inspection (DAI), 267-268**
- dynamic auto, 143**
- dynamic desirable, 143**
 - verifying, 144

E

- eBGP, TTLs, 761-763**
- EIGRP (Enhanced Interior Gateway Routing Protocol), 462, 512**
 - Do I Know This Already? quizzes, 512-516
 - IPv4, 517
 - ACLs, 527-528*
 - authentication, 525-527*
 - different subnets, 524-526*
 - incorrect network statements, 520-522*
 - interface is down, 518*
 - mismatched autonomous system numbers, 518-520*
 - mismatched K values, 522-523*
 - neighbor adjacencies, 517-518*
 - passive interface feature, 523-524*
 - redistribution, 707*
 - timers, 528*
 - trouble ticket 14-1, 546-553*
 - trouble ticket 14-2, 553-557*
 - trouble ticket 14-3, 557-560*
 - trouble tickets, 546*
 - IPv4 issues, 539
 - discontiguous networks and auto-summarization, 542-543*
 - load balancing, 544-545*

- route summarization*, 543-544
- successors*, 539-542
- IPv4 routes, 528-530
 - bad or missing network commands*, 529-530
 - interface is down*, 537
 - route filtering*, 534-535
 - source information*, 533-534
 - split horizon*, 537-539
 - stub configuration*, 535-537
- IPv6, 561
 - ACLs, 564
 - interface is down*, 561-562
 - interface not participating in routing process*, 563-564
 - mismatched authentication*, 562-563
 - mismatched autonomous system numbers*, 562
 - mismatched K values*, 562
 - neighbor issues*, 561
 - passive interface feature*, 562
 - redistribution*, 706
 - timers*, 563
 - trouble ticket 14-4*, 568-571
 - trouble tickets*, 567
- IPv6 route, 564
 - interface not participating in routing process*, 564
 - route filtering*, 565
 - source information*, 565
 - split horizon*, 566-567
 - stub configuration*, 565-566
- named EIGRP configurations, 572-573
 - trouble ticket 14-5*, 577-581
 - trouble tickets*, 577
 - verification commands*, 573-576
 - redistribution*, 706-710
 - trouble tickets*, 880-883
- EIGRP routes, verifying, 686-687
- eliminate potential causes, structured troubleshooting, 16-17
- EMM applets, 63
- EMM configuration, testing, 63-64
- encapsulation mismatch, trunks, 141-142
- end-user IP addresses, verifying, 153
- Enhanced Interior Gateway Routing Protocol. *See* EIGRP (Enhanced Interior Gateway Routing Protocol)
- entries, verifying, 448
- err-disable reason, 259-260
- err-disable recovery feature, 258-259
- err-disabled state, 255-257
- error message, SW1, 236
- EtherChannel
 - Layer 3 EtherChannel, 237-239
 - options for forming, 238
 - trouble tickets, 200
 - trouble ticket 5-4*, 201-204
 - trouble ticket 5-5*, 204-205
- EtherChannel modes, 199
- Ethernet switches, 132
- EUI-64, 373-375
- Exam Engine, installing, 944
- examine collected information, structured troubleshooting, 15-16
- exams, study plans, 946-949
- Excess-Col, 98
- excessive BGP memory use, 123-124
- excessive CPU utilization, 107
 - processes that cause excessive CPU utilization, 107-113
- excessive memory utilization, router performance issues, 121

exchange, adjacency states, 591
 exit interface specified, static routes, 441
 exstart, adjacency states, 591
 extended numbered ACL, IPv4 ACLs, 402

F

failed pings, 214-215
 fast switching, 114-115
 fault management, FCAPS, 28
 FCAPS, 28
 FCS-Err, 98
 FHRPs (first-hop routing protocols), 834
 FIB (Forwarding Information Base), 431
 files, redirecting show command output to, 73-74
 filtering
 IPv4 ACLs, 403
 IPv6 ACLs, 409-410
 filtering output of show commands, show processes cpu, 70
 first hop, verifying
 GLBP (Gateway Load Balancing Protocol), 325-326
 HSRP (Hot Standby Router Protocol), 294-296
 VRRP (Virtual Router Redundancy Protocol), 310-312
 first-hop redundancy protocols, Do I Know This Already? quizzes, 286-289
 first-hop routing protocols (FHRPs), 834
 following the traffic path method, 23
 forwarding, nondesignated port (X), 177
 Forwarding Information Base (FIB), 431

forwarding logic, Cisco Catalyst switches, 96
 frame-forwarding process, 132-140
 FTP servers
 adding login credentials, 49
 backing up router start up configuration, 49
 full, adjacency states, 591

G

Gateway Load Balancing Protocol. *See* GLBP (Gateway Load Balancing Protocol)
 generic routing encapsulation. *See* GRE (generic routing encapsulation)
 Giants, 98
 GLBP (Gateway Load Balancing Protocol), 318
 comparing to HSRP and VRRP, 330
 object tracking, 323-325
 reviewing, 319-321
 trouble tickets, 326
 trouble ticket 8-6, 327-329
 trouble ticket 8-7, 329-330
 verifying, 321-323
 first hop, 325-326
 virtual router MAC address, 323
 GRE (generic routing encapsulation)
 IPsec modes, 457
 troubleshooting considerations, 453-454
 tunnels, 450-458
 IPv6 traffic, 455
 GRE tunnels, 450-458
 GUI (graphical user interface), 47
 GUI tools, 48

H

hardware

troubleshooting, Cisco IOS, 74

high CPU utilization

switch performance issues, 105-106

high processor utilization, troubleshooting, 108-113

higher revision number, VTP (VLAN Trunking Protocol), 151-152

Hot Standby Router Protocol. *See* HSRP (Hot Standby Router Protocol)

HQ, GRE tunnel configuration, 451

HSRP (Hot Standby Router Protocol), 290

comparing to VRRP and GLBP, 330

converging after a failure, 291

debug, 296-297

interface tracking, 293-294

reviewing, 290-291

trouble tickets, 297, 876-880

trouble ticket 8-1, 297-300

trouble ticket 8-2, 300-302

trouble ticket 8-3, 302-306

verifying, 292-293

first hop, 294-296

virtual router MAC address, 293

HTTP server login credentials to router configurations, adding, 49

hypothesis

proposing, structured troubleshooting, 17-18

verifying, structured troubleshooting, 18

I

identifying

security violations, 255

wedged interfaces, 122

inappropriate EtherChannel distribution algorithm, Layer 3 EtherChannel, 238

incompatible trunking modes, 143-146

incorrect IP addressing, VLANs, 152-153

incorrect neighbor statement, BGP (Border Gateway Protocol), neighbor adjacencies, 757-758

incorrect network statements, EIGRP (Enhanced Interior Gateway Routing Protocol), IPv4, 520-522

incorrect port assignments, VLANs, 154-155

information collection, 45-46

INFORMATION-REQUEST, 384

init, adjacency states, 591

inside global

NAT (Network Address Translation), 352

inside local

NAT IPs, 352

installing Exam Engine, 944

inter-VLAN routing

Do I Know This Already? quizzes, 208-211

issues that prevent, 213

router-on-a-trunk/stick, 212-213

SVIs (switched virtual interfaces). *See* SVIs (switched virtual interfaces)

interface IP addresses, verifying, 471

interface is down

BGP (Border Gateway Protocol), neighbor adjacencies, 754

- EIGRP (Enhanced Interior Gateway Routing Protocol)
 - IPv4*, 518
 - IPv4 routes*, 537
 - IPv6*, 561-562
- OSPFv2, 593
- OSPFv2 routes, 614
- interface tracking, HSRP (Hot Standby Router Protocol), 293-294
- interfaces
 - RIPv2, shut down interfaces, 469
 - SVIs. *See* SVIs (switched virtual interfaces)
- interrupt-driven tasks, 27
- inventory of network equipment, documentation, 32
- IP address assignments, documentation, 32
- IP addresses
 - determining IP addresses within subnets, 341-342
 - verifying, end-user IP addresses, 153
- IP addressing, verifying, 340-341
- IP Background process, 108
- IP helper address, verifying, 360
- ip policy route-map, 692
- IP routing tables, 430
- IP Source Guard, 268-269
 - verifying, 269
- ipconfig, 360
 - PC1, 215
 - verifying IP addresses, 342
- ipconfig/all, 216
- IPsec modes, GRE (generic routing encapsulation), 457
- IPv4**
 - EIGRP (Enhanced Interior Gateway Routing Protocol), 517
 - ACLs*, 527-528
 - authentication*, 525-527
 - different subnets*, 524-526
 - discontiguous networks and auto-summarization*, 542-543
 - incorrect network statements*, 520-522
 - interface is down*, 518
 - load balancing*, 544-545
 - mismatched autonomous system numbers*, 518-520
 - mismatched K values*, 522-523
 - neighbor adjacencies*, 517-518
 - passive interface feature*, 523-524
 - route summarization*, 543-544
 - successors*, 539-542
 - timers*, 528
 - trouble ticket 14-1*, 546-553
 - trouble ticket 14-2*, 553-557
 - trouble ticket 14-3*, 557-560
 - trouble tickets*, 546
 - prefix lists, 414
 - redistribution. *See* redistribution
- IPv4 ACLs, 401**
 - filtering, 403
 - reading, 401-402
 - time-based, 403-404
 - trouble tickets, 405-407
- IPv4 addressing, 338**
 - determining IP addresses within subnets, 341-342
- DHCP (Dynamic Host Configuration Protocol), 342
 - reviewing DHCP operations*, 342-347

troubleshooting commands,
348-350

troubleshooting issues, 347-348

issues, 338-341

trouble tickets, 356

trouble ticket 9-1, 356-358

trouble ticket 9-2, 358-361

trouble ticket 9-3, 361-363

IPv4 routes, EIGRP (Enhanced Interior Gateway Routing Protocol), 528-530

bad or missing network commands,
529-530

interface is down, 537

route filtering, 534-535

source information, 533-534

split horizon, 537-539

stub configuration, 535-537

IPv4 static routes, 439-443

IPv6, 334

BGP for IPv6, 786-790

EIGRP (Enhanced Interior Gateway
Routing Protocol), 561

ACLs, 564

interface is down, 561-562

interface not participating in routing process, 563-564

mismatched authentication,
562-563

*mismatched autonomous system
numbers, 562*

mismatched K values, 562

neighbor issues, 561

timers, 563

trouble ticket 14-4, 568-571

trouble tickets, 567

OSPFv3, 641

troubleshooting commands,
641-647

redistribution. *See* redistribution

IPv6 ACLs, 407

filtering, 409-410

reading, 408-409

trouble tickets, 410-414

IPv6 addressing, 370

assignments, 375

stateless address autoconfiguration/SLAAC, 375-380

DHCPv6, 384

DHCPv6 relay agent, 385-386

Do I Know This Already? quizzes,
366-369

EIGRP (Enhanced Interior Gateway
Routing Protocol), passive interface
feature, 562

EUI-64, 373-375

NS (Neighbor Solicitation), 370-373

reviewing, 370

stateful DHCPv6, 381-382

stateless DHCPv6, 382-384

trouble tickets, 386

trouble ticket 10-1, 386-389

trouble ticket 10-2, 389-393

tunnel interface, 457

IPv6 route, EIGRP (Enhanced Interior Gateway Routing Protocol), 564

interface not participating in routing
process, 564

route filtering, 565

source information, 565

split horizon, 566-567

stub configuration, 565-566

IPv6 static routes, 443-445

IPv6 unicast, RIPng (RIP next generation), 492

ISL (Inter-Switch Link), 141

issuing pings, from PC1, 158

ITIL (IT Infrastructure Library), 28

J-K

K values, EIGRP (Enhanced Interior Gateway Routing Protocol)

IPv4, 522-523

IPv6, 562

L

LACP (Link Aggregation Control Protocol), 239

Late-Col, 98

Layer 2 EtherChannel, 199

reviewing, 199

Layer 2 loops, 172

Layer 2 switch communication, troubleshooting, 140

Layer 2 trouble tickets, 157

trouble ticket 4-1, 158-160

trouble ticket 4-2, 160-164

Layer 3 connectivity, BGP (Border Gateway Protocol), neighbor adjacencies, 754-755

Layer 3 EtherChannel, 237-239

trouble tickets, 239-243

Layer 3 packet-forwarding process, 427-431

Layer 3 switch, routed ports, 233-234

Layer 3 to Layer 2 mapping table, 430

learning, nondesignated port (X), 177

Link Aggregation Control Protocol (LACP), 239

listening, nondesignated port (X), 177

listing of interconnections, 32

load balancing

EIGRP (Enhanced Interior Gateway Routing Protocol), IPv4 issues, 544-545

OSPFv2, 626-627

RIPng (RIP next generation), 495

load sharing, RIPv2, 485-486

loading, adjacency states, 591

logging configurations, 55

logging tools, 53-55

severity levels, 54

logic of route maps, 680

logical topology diagrams, 32

login credentials

adding to router configurations, 49

HTTP servers, adding to router configurations, 49

Loop Guard, 190

LSAs (link-state advertisements), 712

OSPF (Open Shortest Path First),

LSDB, OSPFv3, 644

M

MAC address lookup, ARP cache, 440

MAC address tables, 155-157

troubleshooting corrupt switches, 180-181

MAC addresses

Ethernet switches, 132

IP MAC filtering without port security, 269

maximum number reached, 253-254

static MAC addresses, 251-253

maintenance procedures, 29

network changes, managing, 30-31

network documentation, 32-33

network performance, measuring, 34

restoring operations after a failure, 33-34

routine maintenance tasks, 29-30

scheduled maintenance, 30

troubleshooting

change management, 37-38

communication, 36-37

establishing baselines, 36

maintenance tools

advanced tools, 57

basic tools, 47

Cisco Support tools, 64

CLI tools, 47-48

GUI tools, 48

logging tools, 53-55

network documentation tools, 46-47

NTP (Network Time Protocol), 56-57

recovery tools, 48-53

management access

Cisco IOS AAA, 858-861

Do I Know This Already? quizzes,
850-853

password encryption levels, 858

SSH (Secure Shell), 857-858

Telnet, 855-857

trouble tickets, 861, 918-922

trouble ticket 20-1, 862-863

trouble ticket 20-2, 864-865

trouble ticket 20-3, 865-867

vtty access, 855

management access console access, 854-855

management protocols, 818

Cisco IOS IP SLA, 827-833

Do I Know This Already? quizzes,
814-817

NTP (Network Time Protocol), 818-821

SNMP (Simple Network Management
Protocol), 823-826

syslog, 821-823

trouble tickets, 837-844

management tools, 826-827

object tracking, 833-834

RSPAN (Remote SPAN), 835-837

SPAN (Switched Port Analyzer),
835-837

managing network changes, 30-31

max hop count exceeded, RIPv2,
475-477

measuring, network performance, 34

memory-allocation failure, 122

memory leaks, router performance
issues, 121-122

memory tables, 945

message types, DHCP (Dynamic Host
Configuration Protocol), 384

metrics, routing protocols, 701

misconfigured peer groups, BGP (Border
Gateway Protocol), neighbor adjacencies,
764-765

mismatched area numbers, OSPFv2,
596-597

mismatched area types, OSPFv2,
597-598

mismatched authentication

BGP (Border Gateway Protocol), neighbor
adjacencies, 763-764

EIGRP (Enhanced Interior Gateway
Routing Protocol), IPv6, 562-563

OSPFv2, 600-601

mismatched autonomous system numbers,
EIGRP (Enhanced Interior
Gateway Routing Protocol)

IPv4, 518-520

IPv6, 562

mismatched duplex settings, Cisco
Catalyst switches, 99-101

mismatched EtherChannel configuration,
Layer 3 EtherChannel, 238

mismatched K values, EIGRP (Enhanced Interior Gateway Routing Protocol)
 IPv4, 522-523
 IPv6, 562
 mismatched network types, OSPFv2, 604-605
 mismatched port configurations, Layer 3 EtherChannel, 237
 mismatched timers, OSPFv2, 594-595
 missing default route, RIPv2, 486-487
 missing routes, RIPv2, 466-468
 missing VLANs, 153-154
 mode mismatch, VTP (VLAN Trunking Protocol), 149-150
 modifying route map configuration, 688
 monitoring network performance, 30
 MP-BGP, trouble tickets, 807-809
 MSTP (Multiple Spanning Tree Protocol), 179-180
 MTU (maximum transmission unit), 65
 MTU mismatch, OSPFv2, 602-603
 Multi-Col, 98
 Multiple Spanning Tree Protocol (MSTP), 179-180

N

named ACL configuration mode, IPv4 ACLs, 406-407
 named EIGRP configurations, 572-573
 trouble tickets, 577-581
 verification commands, 573-576
 NAT (Network Address Translation), 350
 names of IP addresses, 352
 reviewing, 350-352
 trouble tickets, 923-926

troubleshooting commands, 354-355
 troubleshooting issues, 353-354
 native VLAN mismatch, trunks, 146-147
 NDP (Neighbor Discovery Protocol), 371
 neighbor adjacencies
 BGP (Border Gateway Protocol), 753-754
 ACLs, 759-761
 BGP packets sourced from wrong IP address, 758-759
 incorrect neighbor statement, 757-758
 interface is down, 754
 Layer 3 connectivity, 754-755
 misconfigured peer groups, 764-765
 mismatched authentication, 763-764
 neighbor does not have a route to local router, 756-757
 path to neighbor is via default route, 755-756
 timers, 765-766
 TTL of BGP packet expires, 761-763
 EIGRP (Enhanced Interior Gateway Routing Protocol), IPv4, 517-518
 OSPFv2, 590-593
 neighbor advertisement, IPv6 addressing, 370-373
 Neighbor Discovery Protocol. *See* NDP (Neighbor Discovery Protocol)
 neighbor issues, EIGRP (Enhanced Interior Gateway Routing Protocol), IPv6, 561
 neighbor remote-as, 758
 Net Background, 108

NetFlow, 57-61

- baselines, creating, 58

Network Address Translation. *See* NAT (Network Address Translation)

- network changes, managing, 30-31

- network commands, EIGRP (Enhanced Interior Gateway Routing Protocol, IPv4 routes, 529-530

- network documentation, 32-33

- network documentation tools, 46-47, 80-84

- network events, notifications, 61-64

- network maintenance, 26

- defining, 26-27

- proactive versus reactive, 27-28

- troubleshooting, 34-35

- maintaining current network documentation, 35*

- network maintenance models, 28

- adapting, 28-29

- network mask command, BGP routes, 768-770

- network performance

- measuring, 34

- monitoring, 30

- network statements

- EIGRP (Enhanced Interior Gateway Routing Protocol), IPv4, 520-522

- RIPv2, 470-471

- Network Time Protocol. *See* NTP (Network Time Protocol)

- network types, mismatched network types, OSPFv2, 604-605

- Next-Hop option, static routes, IPv4, 439-443

- Next Hop Resolution Protocol (NHRP), 434

- next-hop router not reachable, BGP routes, 770-772

- NHRP (Next Hop Resolution Protocol), 434

- nondesignated port (X), 174

- STP (Spanning-Tree Protocol), 176-177

- nonroot bridges, STP (Spanning-Tree Protocol), 173

- notifications, for network events, 61-64

- NS (Neighbor Solicitation), IPv6 addressing, 370-373

- NTP (Network Time Protocol), 56-57, 818-821

- adding source command, 844

O

- object tracking, 833-834

- GLBP (Gateway Load Balancing Protocol), 323-325

- VRRP (Virtual Router Redundancy Protocol), 309-310

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

- operations, STP (Spanning-Tree Protocol), 173-175

- original design documents, documentation, 33

- OSPF authentication keys, verifying, 600

- OSPF distribute list command, verifying, 612

- OSPF LSAs, 621

- OSPF (Open Shortest Path First), 462, 586

- Do I Know This Already? quizzes, 586-589

- LSAs

- for OSPF v3

- redistribution, 710

- trouble tickets, 884-901

OSPFv2, 590

- ACLs (access control lists), 601-602
- default routes, 627
- different subnets, 598-599
- discontiguous areas, 624-626
- duplicate router IDs, 603-604
- interface is down, 593
- interface not running the OSPF process, 593-594
- load balancing, 626-627
- mismatched area numbers, 596-597
- mismatched area types, 597-598
- mismatched authentication, 600-601
- mismatched network types, 604-605
- mismatched timers, 594-595
- MTU mismatch, 602-603
- neighbor adjacencies, 590-593
- passive interface feature, 599
- redistributed routes in RIP domain, 713
- route summarization, 622-624
- trouble tickets, 627
 - trouble ticket 15-1, 628-635*
 - trouble ticket 15-2, 635-639*
 - troubled ticket 15-3, 639-641*

OSPFv2 redistributed routes, 712**OSPFv2 routes, 606**

- duplicate router IDs, 619-620
- interface is down, 614
- interface not running the OSPF process, 606-607
- route filtering, 611-613
- source information, 607-610
- stub area configuration, 613-614
- wrong designated router was elected, 615-618

OSPFv2 tracking OSPF advertisements through a network, 620-621**OSPFv3**

- AFs (address families), 655-664
 - trouble tickets, 664-668*
- displaying routes, 646
- IPv6, 641
 - troubleshooting commands, 641-647*
- LSDB, 644
- trouble tickets, 647, 926-934
 - trouble ticket 15-4, 647-650*
 - trouble ticket 15-5, 650-654*

outside global

- NAT (Network Address Translation), 352

outside local

- NAT (Network Address Translation), 352

P**packet captures, collect information, 75-76****packet-forwarding process, 427**

- Layer 3 packet-forwarding process, 427-431
- troubleshooting, 431-435

packet matches, verifying, IPv4 ACLs, 407**packet-switching modes, 113, 116-121**

- commands for troubleshooting
- fast switching, 114-115
- process switching, 113-114

passive interface feature

- EIGRP (Enhanced Interior Gateway Routing Protocol)
 - IPv4, 523-524*
 - IPv6, 562*

- OSPFv2, 599
- RIPv2, 471-473
- password encryption levels, 858
- password mismatch, VTP (VLAN Trunking Protocol), 151
- PBR (policy-based routing), 681-684
- PBR path, verifying, 683
- PC1
 - ipconfig, 215
 - issuing pings, 158
- PCPT (Pearson Cert Practice Test), 943
 - activating and downloading, 944
- Pearson Cert Practice Test (PCPT) engine, 943
- peer groups, misconfigured peer groups, BGP (Border Gateway Protocol), 764-765
- performance management, FCAPS, 28
- permit sequence, 679
- permit statement, 679
- physical topology diagrams, 32
- ping, 64-66
- ping sweeps, 66
- pings
 - failed pings, 214-215
 - issuing from PC1, 158
 - successful pings, 218
- policy-based routing. *See* PBR (policy-based routing)
 - Do I Know This Already? quizzes, 674-677
 - trouble tickets, 684
 - trouble ticket 16-1*, 685-688
 - trouble ticket 16-2*, 689-691
 - trouble ticket 16-3*, 691-692
- policy matches, verifying, 688
- populating, TCAM (ternary content-addressable memory), 102
- port errors, Cisco Catalyst switches, 97-98
- port roles, STP (Spanning-Tree Protocol), 174
- port security, 250
 - common issues, 250
 - configured but not enabled*, 250-251
 - legitimate users being blocked because of violation*, 254-260
 - maximum number of MAC addresses reached*, 253-254
 - running configuration not saved to startup configuration*, 260-261
 - trouble tickets, 261-265
- port security common issues static MAC address not configured correctly, 251-253
- port type during configuration, Layer 3 EtherChannel, 238
- PortFast, 183-184
- ports
 - Cisco Catalyst switches, 96
 - incorrect port assignments, VLANs, 154-155
 - protected ports, 273-275
 - routed ports, 233-234
- practice exercises, selecting a troubleshooting approach, 25-26
- prefix lists, 414
 - processing, 415-416
 - reading, 414-415
 - trouble tickets, 416-418
- Premium Edition, 945
- private autonomous system numbers, BGP path selection, 784
- private VLANs. *See* PVLANS (private VLANs)

proactive network maintenance, 27-28
 problem reports, structured trouble-shooting, 13-14
 problem resolution, structured trouble-shooting, 19
 problems, diagnosing, 10
 process-switching modes, 116
 process switching, packet-switching modes, 113-114
 processes that cause excessive CPU utilization, 107-113
 ARP Input process, 107-108
 IP Background process, 108
 Net Background, 108
 TCP Timer process, 108
 processing, prefix lists, 415-416
 processor utilization, 106
 propose an hypothesis, structured troubleshooting, 17-18
 protected ports, 273-275
 protocols, management protocols. *See* management protocols
 proxy ARP enabled, 442
 punting, TCAM (ternary content-addressable memory), 102
 PVLANS (private VLANs), 275-279

Q

quizzes

addressing, 334-337
 BGP (Border Gateway Protocol), 748-752
 device performance, 92-95
 EIGRP (Enhanced Interior Gateway Routing Protocol), 512-516
 first-hop redundancy protocols, 286-289
 Inter-VLAN routing, 208-211

IPv4 ACLs, IPv6 ACLs, prefix lists, 396-400
 IPv4/IPv6 routing and GRE tunnels, 422-426
 IPv6 addressing, 366-369
 maintenance tools, 40-44
 management access, 850-853
 management protocols, 814-817
 OSPF (Open Shortest Path First), 586-589
 redistribution, 696-699
 RIPng (RIP next generation), 462-465
 RIPv2, 462-465
 route maps and policy-based routing, 674-677
 STP (Spanning-Tree Protocol), 168-171
 switch security, 246-249

R

R1

show cdp neighbors, 81-82
 show version, 83
 show vlans, 219-220
 updating routers, 83

RA, verifying, 379

Rcv-Err, 98

reactive network maintenance, 27-28

reading

IPv4 ACLs, 401-402
 IPv6 ACLs, 408-409
 prefix lists, 414-415
 route maps, 678-680

REBIND, 384

RECONFIGURE, 384

recovery tools, 48-53

recursive lookup, IPv4 static routes, 440

redirecting, show command output to files, 73-74

redistribution, 700

advanced redistribution, 737

routing loops, 739-744

BGP (Border Gateway Protocol), 715-718

Do I Know This Already? quizzes, 696-699

EIGRP (Enhanced Interior Gateway Routing Protocol), 706-710

OSPF, 710-715

RIP, 703-705

route maps, 718

route redistribution overview, 700-702

suboptimal routing, suboptimal routing, 737-739

trouble tickets, 718, 901-910

trouble ticket 17-1, 439-442

trouble ticket 17-2, 723-727

trouble ticket 17-3, 727-732

trouble ticket 17-4, 733-737

redistribution configuration, 702

RELAY-FORW, 384

RELAY-REPL, 384

RELEASE, 384

RENEW, 384

replacement of hardware, routine maintenance tasks, 30

REPLY, 384

REQUEST, 384

restoring, archived configurations, 53

restoring operations after a failure, maintenance procedures, 33-34

reviewing

DHCP operations, IPv4 addressing, 342-347

GLBP (Gateway Load Balancing Protocol), 319-321

HSRP (Hot Standby Router Protocol), 290-291

IPv6 addressing, 370

NAT (Network Address Translation), 350-352

SVIs (switched virtual interfaces), 221-223

VRRP (Virtual Router Redundancy Protocol), 306-308

RFC 1918, 334

RIP (Routing Information Protocol), 462

redistribution, 703-705

RIPng (RIP next generation), 462, 492-497

ACLs, 496-497

debug, 494

default routes, 495-496

Do I Know This Already? quizzes, 462-465

load balancing, 495

trouble tickets, 498, 934-940

trouble ticket 13-1, 498-502

trouble ticket 13-2, 502-506

trouble ticket 13-3, 506-508

verifying interfaces, 497

viewing routes, 493

RIPv2, 466

ACLs, 485

authentication, 477-479

autosummarization, 482-483

bad or missing network statements, 470-471

better source of information, 483-484

Do I Know This Already? quizzes, 462-465

interface is shut down, 469

load sharing, 485-486

max hop count exceeded, 475-477

missing default route, 486-487

- missing routes, 466-468
- passive interface feature, 471-473
- route filtering, 479-480
- route summarization, 487-491
- split horizon, 480-481
- trouble tickets, 498
 - trouble ticket 13-1*, 498-502
 - trouble ticket 13-2*, 502-506
 - trouble ticket 13-3*, 506-508
- wrong subnets, 469-470
- wrong version, 473-475
- root bridges, STP (Spanning-Tree Protocol), 173**
- Root Guard, 189-190**
- root ports**
 - STP (Spanning-Tree Protocol), 175-176
- route caching, 114-115**
- route filtering**
 - BGP routes, 775-780
 - EIGRP (Enhanced Interior Gateway Routing Protocol)
 - IPv4 routes*, 534-535
 - IPv6 route*, 565
 - OSPFv2 routes, 611-613
 - RIPv2, 479-480
- route filters, verifying with show ip protocols, 479**
- route information, sources of, 436-438**
- route map configuration**
 - modifying, 688
 - verifying, 688
- route maps, 678**
 - Do I Know This Already? quizzes, 674-677
 - logic of, 680
 - reading, 678-680
 - redistribution, 718
 - route selection, 702

route summarization

- EIGRP (Enhanced Interior Gateway Routing Protocol), IPv4 issues, 543-544

- OSPFv2, 622-624

- RIPv2, 487-491

routed ports, 233-234

- configuring, on SW1, 237

- trouble tickets, 234-237

router-on-a-trunk/stick, 212-213

- trouble tickets, 213

- trouble ticket 6-1*, 214-218

- trouble ticket 6-2*, 218-220

router performance issues, 106-107

- buffer leaks, 122-123

- CEF (Cisco Express Forwarding), 115-116

- excessive BGP memory use, 123-124

- excessive CPU utilization, 107

- processes that cause excessive CPU utilization*, 107-113

- excessive memory utilization, 121

- memory-allocation failure, 122

- memory leaks, 121-122

- packet-switching modes, 113, 116-121

- fast switching*, 114-115

- process switching*, 113-114

- process-switching modes, 116

router start up configuration, backing up without specifying login credentials, 49

routers, 92

- updating, R1, 83

routes

- BGP (Border Gateway Protocol). *See* BGP routes

- missing default route, RIPv2, 486-487

Routing Information Protocol. *See* RIP (Routing Information Protocol)

routing information sources

- sources of route information, 436-438
- troubleshooting, 435

- routing information sources data structures and routing tables, 436

routing loops

- redistribution, 739-744
- verifying, with trace, 490-491

- routing protocols, metrics, 701

- routing table entries, verifying, 446, 686

- routing tables, routing information sources, 436

- RP (root port), 174

- RSPAN (Remote SPAN), 78-79, 835-837

- running configuration, comparing to startup configuration before issuing copy command, 52

- Runts, 98

S

- scheduled backups, routine maintenance tasks, 30

- scheduled maintenance, 30

SDM template

- Cisco Catalyst switches, 104
- verifying, 105

- Secure Shell (SSH), 857-858

- security. *See also* switch security

- access control. *See* access control
- spoof-prevention features. *See* spoof-prevention features

- security management, FCAPS, 28

- security violations, identifying, 255

- seed metrics, redistribution, 702

- selecting, troubleshooting methods, 25-26

severity levels

- logging tools, 54

- shoot from the hip method, 6-12

- show adjacency detail, 116, 120, 435

- show buffers, 123

- show cdp neighbors, 81, 548

- R1, 81-82

show commands

- filtering output of, 69-73

- redirecting output to files, 73-74

- show controllers, 74

- show etherchannel summary, 203-204, 239

- show frame-relay map, 434

- show glbp, 322, 328

- show glbp brief, 321, 329-330

- show interface interface t, 110

- show interface interface_type interface number, 108

- show interface switchport, 141-142

- show interface trunk, 142

- show interface tunnel, 456-457

- show interfaces, 74, 98

- show interfaces gig1/0/10 switchport, SW1, 236

- show interfaces switchport, 141

- show interfaces trunk, 142, 147, 217

- show ip arp, 108-109, 433

- show ip cache, 116-117

- show ip cef, 116, 120

- show ip cef adjacency egress_interface_id next_hop_ip_address detail, 116-119

- show ip cef exact-route source_address destination_address, 433

- show ip cef ip_address, 120, 433

- show ip cef [ip_address] [subnet_mask], 433

show ip dhcp binding, 349
 show ip dhcp conflict, 349
 show ip eigrp, 539
 show ip eigrp interfaces, 548-551, 721
 show ip eigrp neighbors, 548-551
 show ip eigrp topology, 540-541
 show ip interface, RIPv2, 470
 show ip interface brief, 71, 80, 204, 304
 SW1, 236
 show ip interface interface_type interface_number, 116-117
 show ip nat statistics, 355
 show ip nat translations, 354
 show ip nhrp, 434
 show ip ospf database, 609
 show ip ospf database router, 609
 show ip policy, 683
 show ip protocols, 547
 RIP settings, 468
 verifying route filters, 479
 show ip rip database, 467, 704
 show ip route, 72-73, 303, 547, 550-551
 SW1, 232
 show ip route [ip_address], 432
 show ip route ip_address, 120
 show ip route [ip_address] [subnet_mask], 432
 show ip route [ip_address] [subnet_mask] [longer-prefixes], 432
 show ip route ospf, 608
 show ip route rip, 467-468
 show ip sla application, 829
 show ip sla configuration, 829
 show ip sla responder, 832
 show ip sla statistics, 831
 show ipv6 ospf, 642
 show ipv6 ospf database, redistribution, 714
 show ipv6 ospf interface brief, 643
 show ipv6 protocols, 493, 641
 RIPng (RIP next generation) redistribution, 705
 show ipv6 rip TSHOOT_RIP, 493
 show ipvt ospf interface interface_type interface_number, 643
 show ipvt ospf neighbor, 644
 show ipvt protocols, 641
 show mac address-table, 133
 show mac address-table dynamic, 216
 show memory, 74
 show memory allocating-process totals, 121
 show ospfv3 database, 661
 show ospfv3 interface, 659
 show ospfv3 neighbor, 660
 show platform, 74
 show platform tcam utilization, 103
 show processes cpu, 69, 74, 108, 111-112, 116-118
 filtering output of show commands, 70
 show processes cpu history, 108, 112-113
 show processes memory| include, 123
 show route-map, 683-684
 show route-map TSHOOT_ROUTE_MAP, 679
 show run, 212-213
 show run interface gigabitethernet, 202
 show run interface vlan 10, 305
 show run | section router eigrp, 549
 show running-config, 71-72
 show sdm prefer, 103
 show spanning-tree, 184

- show spanning-tree inconsistent ports, 198
- show spanning-tree interface interface_type interface number detail, 178-179
- show spanning-tree mst configuration, 180
- show spanning-tree vlan, 178
- show spanning-tree [vlan {vlan_id}], 178
- show standby, 294, 304
- show standby brief, 292, 303
- show standby fastethernet 0/0, 293
- show tcp statistics, 108-111
- show track, 310, 325
- show version, R1, 83
- show vlan brief, 159
- show vlans, 217
 - R1, 219-220
- show vrrp, 310
- show vrrp brief, 308
- show vrrp interface vlan 20, 309
- shut down interfaces, RIPv2, 469
- simplified troubleshooting flow, 10
- Single-Col, 98
- SLAAC (stateless address autoconfiguration), 388
 - IPv6 addressing, 375-380
- SNMP (Simple Network Management Protocol), 57-59
 - baselines, creating, 58
 - notifications for network events, 61-64
- SNMP traps, enabling, 62
- SNMPv3, 824-825
- software, updating, 30
- SOLICIT, 384

- source information
 - BGP routes, 773-775
 - EIGRP (Enhanced Interior Gateway Routing Protocol)
 - IPv4 routes, 533-534
 - IPv6 route, 565
 - OSPFv2 routes, 607-610
 - RIPv2, 483-484
- source routing protocol, 702
- SPAN (Switched Port Analyzer), 835-837
 - collect information, 76-78
- Spanning-Tree Protocol. *See* STP (Spanning-Tree Protocol)
- specific routes, verifying, 447
- split horizon
 - BGP routes, 772-773
 - EIGRP (Enhanced Interior Gateway Routing Protocol)
 - IPv4 routes, 537-539
 - IPv6 route, 566-567
 - RIPv2, 480-481
- spoof-prevention features, 265
 - DAI (dynamic ARP inspection), 267-268
 - DHCP snooping, 265-267
 - IP Source Guard, 268-269
 - trouble tickets, 270-272
- SSH (Secure Shell), 857-858
- startup configuration, comparing to running configuration before issuing copy command, 52
- stateful DHCPv6, IPv6 addressing, 381-382
- stateless address autoconfiguration/SLAAC, IPv6 addressing, 375-380
- stateless DHCPv6, IPv6 addressing, 382-384

static routes, 439

- exit interface specified, 441
- IPv4, 439-443
- IPv6, 443-445
- trouble tickets, 445
 - trouble ticket 12-1, 445-448*
 - trouble ticket 12-2, 448-450*

sticky features, port security, 260-261**STP (Spanning-Tree Protocol)**

- collect information, 177
 - gathering STP information, 177-179*
 - MSTP (Multiple Spanning Tree Protocol), 179-180*
- default port costs, 175
- designated ports, 176
- Do I Know This Already? quizzes, 168-171
- nondesignated port (X), 176-177
- overview, 172
- port roles, 174
- reviewing operation, 173-175
- root ports, determining, 175-176
- trouble tickets, 872-876
 - trouble ticket 5-1, 191-193*
 - trouble ticket 5-2, 194-196*
 - trouble ticket 5-3, 196-199*
- troubleshooting
 - broadcast storms, 181-182*
 - corruption of a switch's MAC address table, 180-181*

STP features, 182

- BPDU Filter, 187-188
- BPDU Guard, 184-187
- Loop Guard, 190
- PortFast, 183-184
- Root Guard, 189-190

structured tasks, 27**structured troubleshooting, 11-13**

- collect information, 14
- eliminate potential causes, 16-17
- examine collected information, 15-16
- problem reports, 13-14
- problem resolution, 19
- propose an hypothesis, 17-18
- value of, 11-13
- verify hypothesis, 18

stub area configuration, OSPFv2 routes, 613-614**stub configuration, EIGRP (Enhanced Interior Gateway Routing Protocol)**

- IPv4 routes, 535-537
- IPv6 route, 565-566

study plans, 946-949**subnets**

- different subnets, OSPFv2, 598-599
- EIGRP (Enhanced Interior Gateway Routing Protocol), IPv4, 524-526
- RIPv2, 469-470

subnets determining IP addresses, 341-342**suboptimal routing, advanced redistribution, 737-739****successful pings, 218****successors, EIGRP (Enhanced Interior Gateway Routing Protocol, IPv4 issues, 539-542****SVIs (switched virtual interfaces)**

- reviewing, 221-223
- trouble tickets, 224
 - trouble ticket 6-3, 225-230*
 - trouble ticket 6-4, 230-233*
- troubleshooting, 223-224

switch performance issues, 96

Cisco Catalyst switches, 96-97

mismatched duplex settings,
99-101

port errors, 97-98

high CPU utilization, 105-106

TCAM (ternary content-addressable
memory), 101-105

switch security

Do I Know This Already? quizzes,
246-249

port security. *See* port security

Switched Port Analyzer. *See* SPAN
(Switched Port Analyzer)

switched virtual interfaces. *See* SVIs
(switched virtual interfaces)

switches, 92

syslog, 821-823

T

tables, adjacency tables, 431

tasks, network maintenance, 27

TCAM (ternary content-addressable
memory), 101-105

populating, 102

punting, 102

TCP Timer process, 108

team utilization, verifying, 105

telnet, 67, 855

testing, EMM configuration, 63-64

TFTP server, redirecting output to,
73-74

time-based IPv4 ACLs, 403-404

timers

BGP (Border Gateway Protocol), neigh-
bor adjacencies, 765-766

EIGRP (Enhanced Interior Gateway
Routing Protocol)

IPv4, 528

IPv6, 563

mismatched timers, OSPFv2, 594-595

tools

maintenance tools

advanced tools, 57

basic tools, 47

Cisco Support tools, 64

CLI tools, 47-48

GUI tools, 48

logging tools, 53-55

network documentation tools,
46-47

NTP (Network Time Protocol),
56-57

recovery tools, 48

management tools. *See* management
tools

network documentation tools. *See* net-
work documentation tools

top-down method, 21

trace, verifying, routing loops, 490-491

traceroute, 67-68, 120

tracking OSPF advertisements through a
network OSPFv2, 620-621

transit, collect information, 75

packet captures, 75-76

RSPAN (Remote SPAN), 78-79

SPAN (Switched Port Analyzer), 76-78

trouble ticket reporting system, 46

trouble tickets

BGP (Border Gateway Protocol), 790,
910-918

trouble ticket 18-1, 791-796

trouble ticket 18-2, 796-802

trouble ticket 18-3, 802-806

- EIGRP (Enhanced Interior Gateway Routing Protocol), 880-883
 - IPv4*, 546
- EtherChannel, 200
 - trouble ticket 5-4*, 201-204
 - trouble ticket 5-5*, 204-205
- GLBP (Gateway Load Balancing Protocol), 326
 - trouble ticket 8-6*, 327-329
 - trouble ticket 8-7*, 329-330
- HSRP (Hot Standby Router Protocol), 297, 876-880
 - trouble ticket 8-1*, 297-300
 - trouble ticket 8-2*, 300-302
 - trouble ticket 8-3*, 302-306
- IPv4 ACLs, 405-407
- IPv4 addressing, 356
 - trouble ticket 9-1*, 356-358
 - trouble ticket 9-2*, 358-361
 - trouble ticket 9-3*, 361-363
- IPv6 ACLs, 410-414
- IPv6 addressing, 386
 - trouble ticket 10-1*, 386-389
 - trouble ticket 10-2*, 389-393
- Layer 2 trouble tickets, 157
 - trouble ticket 4-1*, 158-160
 - trouble ticket 4-2*, 160-164
- Layer 3 EtherChannel, 239-243
- management access, 861, 918-922
 - trouble ticket 20-1*, 862-863
 - trouble ticket 20-2*, 864-865
 - trouble ticket 20-3*, 865-867
- management protocols, 837-844
- MP-BGP, 807-809
- named EIGRP configurations, 577
- NAT (Network Address Translation), 923-926
- OSPF (Open Shortest Path First), 884-901
 - OSPFv2, 627
 - trouble ticket 15-1*, 628-635
 - trouble ticket 15-2*, 635-639
 - troubled ticket 15-3*, 639-641
 - OSPFv3, 647, 926-934
 - trouble ticket 15-4*, 647-650
 - trouble ticket 15-5*, 650-654
 - policy-based routing, 684
 - trouble ticket 16-1*, 685-688
 - trouble ticket 16-2*, 689-691
 - trouble ticket 16-3*, 691-692
 - port security, 261-265
 - prefix lists, 416-418
 - redistribution, 718, 901-910
 - trouble ticket 17-1*, 439-442
 - trouble ticket 17-2*, 723-727
 - trouble ticket 17-3*, 727-732
 - trouble ticket 17-4*, 733-737
 - RIPng (RIP next generation), 498, 934-940
 - trouble ticket 13-1*, 498-502
 - trouble ticket 13-2*, 502-506
 - trouble ticket 13-3*, 506-508
 - RIPv2, 498
 - trouble ticket 13-1*, 498-502
 - trouble ticket 13-2*, 502-506
 - trouble ticket 13-3*, 506-508
 - routed ports, 234-237
 - router-on-a-trunk/stick, 213
 - trouble ticket 6-1*, 214-218
 - trouble ticket 6-2*, 218-220
 - spoof-prevention features, 270-272
 - static routes, 445
 - trouble ticket 12-1*, 445-448
 - trouble ticket 12-2*, 448-450
 - STP (Spanning-Tree Protocol), 872-876
 - trouble ticket 5-1*, 191-193
 - trouble ticket 5-2*, 194-196
 - trouble ticket 5-3*, 196-199

SVIs (switched virtual interfaces), 224

trouble ticket 6-3, 225-230

trouble ticket 6-4, 230-233

VRRP (Virtual Router Redundancy Protocol), 312

trouble ticket 8-4, 312-315

trouble ticket 8-5, 315-318

troubleshooting, 9

defining, 9-11

diagnosing problems, 10

GRE tunnels, 450-458

hardware, Cisco IOS, 74

maintenance procedures

change management, 37-38

communication, 36-37

establishing baselines, 36

network maintenance, 34-35

maintaining current network documentation, 35

packet-forwarding process, 431-435

router performance issues. *See* router performance issues

routing information sources, 435

shoot from the hip method, 6-12

simplified troubleshooting flow, 10

steps of, 45

STP features, 182

BPDU Filter, 187-188

BPDU Guard, 184-187

Loop Guard, 190

PortFast, 183-184

Root Guard, 189-190

STP (Spanning-Tree Protocol)

broadcast storms, 181-182

corruption of a switch's MAC address table, 180-181

structured troubleshooting, 11

value of, 11-13

SVIs (switched virtual interfaces), 223-224

switch performance issues. *See* switch performance issues

trunks. *See* trunks

troubleshooting methods, 20

bottom-up method, 21-22

comparing configurations method, 23-24

component swapping method, 24-25

divide-and-conquer method, 22

following the traffic path method, 23

selecting, 25-26

top-down method, 21

trunking administrative (Dynamic Auto), verifying, 144

trunking administrative mode (Access), verifying, 143

trunking administrative mode (Dynamic Desirable), verifying, 144

trunking administrative mode (Trunk), verifying, 143

trunking administrative modes, comparing, 145

trunks, 140, 143

allowed VLANs, 147-148

encapsulation mismatch, 141-142

incompatible trunking modes, 143-146

native VLAN mismatch, 146-147

verifying, 143

VTP domain name mismatch, 146

TTLs, BGP (Border Gateway Protocol), 761-763

U

UnderSize, 98

unicast routing, IPv6 addressing, 378

updated static routes, verifying, 447

updating

 routers, R1, 83

 software, 30

V

VACLs (VLAN access control lists), 279

variable-length subnet masking (VLSM), 542

verification commands, named EIGRP configurations, 573-576

verify hypothesis, structured troubleshooting, 18

verifying

 ACLs, IPv4 ACLs, 406

 default gateways, 380

 default trunking mode on SW2, 145

 DHCP snooping, 266

 DHCP snooping bindings, 267

 DR (designated router), 617-618

 EIGRP routes, 686-687

 end-user IP addresses, 153

 entry exists, 448

 first hop, VRRP (Virtual Router Redundancy Protocol (VRRP), 310-312

 GLBP (Gateway Load Balancing Protocol), 321-323
 first hop, 325-326

 HSRP (Hot Standby Router Protocol), 292-293

first hop, 294-296

 interface IP addresses, 471

 IP addresses with ipconfig, 342

 IP addressing, 340-341

 IP Source Guard, 269

 network IDs with show ip interface, 607

 OSPF authentication keys, 600

 OSPF RID, 604

 packet matches, IPv4 ACLs, 407

 PBR path, 683

 policy matches, 688

 protected ports, 274

 PVLANS (private VLANs), 278

 RAs, 379

 redistributed routes in RIP domain, 705

 RIP authentication, 478-479

 RIP distribute list command, 480

 RIPng (RIP next generation), 497

 route filters, with show ip protocols, 479

 route map configuration, 688

 routes, via tunnel interface on HQ and Branch, 452-453

 routing loops, with trace, 490-491

 routing table entries, 446, 686

 SDM template, 105

 specific routes, 447

 team utilization, 105

 trunking administrative mode (Access), 143

 trunking administrative mode (Dynamic Auto), 144

 trunking administrative mode (Dynamic Desirable), 144

 trunking administrative mode (Trunk), 143

 updated static routes, 447

 virtual links, 626

 VLANs, on a switch, 153

 VRRP (Virtual Router Redundancy Protocol), 308-309

- VTP domain name, 148
- VTP version, 149
- version mismatch, VTP (VLAN Trunking Protocol), 149
- versions, wrong version, RIPv2, 473-475
- viewing
 - configuration archives, 50
 - NAT statistics, 353
 - NAT translations, 362
 - number of IPv6 routes reachable at next-hop router, RIPv2, 494
 - RIPv2 (RIP next generation) routes, 493
- violations
 - identifying, 255
 - legitimate users being blocked because of violations, 254-260
- virtual links, verifying, 626
- virtual router MAC address
 - GLBP (Gateway Load Balancing Protocol), 323
 - HSRP (Hot Standby Router Protocol), 293
 - VRRP (Virtual Router Redundancy Protocol), 309
- Virtual Router Redundancy Protocol. *See* VRRP (Virtual Router Redundancy Protocol)
- VLAN access control lists (VACLs), 279
- VLAN access map, 279
- VLAN filter list, 279
- VLAN Trunking Protocol. *See* VTP domain name mismatch
- VLANs, 152
 - incorrect IP addressing, 152-153
 - incorrect port assignments, 154-155
 - missing VLANs, 153-154
 - verifying on a switch, 153

- VLSM (variable-length subnet masking), 542
- VRRP (Virtual Router Redundancy Protocol), 306
 - comparing to HSRP and GLBP, 330
 - object tracking, 309-310
 - reviewing, 306-308
 - trouble tickets, 312
 - trouble ticket 8-4*, 312-315
 - trouble ticket 8-5*, 315-318
 - verifying, 308-309
 - first hop*, 310-312
 - virtual router MAC address, 309
- VTP (VLAN Trunking Protocol), 148
 - domain name mismatch, 148-149
 - higher revision number, 151-152
 - mode mismatch, 149-150
 - password mismatch, 151
 - version mismatch, 149
- VTP domain name mismatch, trunks, 146
- VTP version, verifying, 149
- vti access, 855
- vti lines, 854
- vti login, 856

W

- wedged interfaces, identifying, 122
- wiki, 46
- witnessing, configuration merge, 53
- wrong subnets, RIPv2, 469-470
- wrong version, RIPv2, 473-475

X-Y-Z

- Xmit-Err, 98

Where are the Companion Content Files?



Thank you for purchasing this

Premium Edition version of:

CCNP Routing and Switching TSHOOT 300-135

Official Cert Guide

The print version of this title comes with a disc of companion content.

As an eBook reader, you have access to these files by following the steps below:

1. Go to ciscopress.com/account and log in.
2. Click on the “Access Bonus Content” link in the Registered Products section of your account page for this product, to be taken to the page where your downloadable content is available.

Please note that many of our companion content files can be very large, especially image and video files.

If you are unable to locate the files for this title by following the steps at left, please visit ciscopress.com/contact and select the “Site Problems/Comments” option. Our customer service representatives will assist you.

The Professional and Personal Technology Brands of Pearson



Cisco Press



informIT

PEARSON IT Certification



QUE

SAMS

vmware PRESS



<https://t.me/learningnets>

Memory Tables

Chapter 1

Table 1-2 *Steps to Diagnose a Problem*

Step	Description
	Because a typical problem report lacks sufficient information to give a troubleshooter insight into a problem's underlying cause, the troubleshooter should collect additional information, perhaps using network maintenance tools or by interviewing impacted users.
	After collecting sufficient information about a problem, the troubleshooter then examines that information, perhaps comparing the information against previously collected baseline information.
	Based on the troubleshooter's knowledge of the network and his interrogation of collected information, he can begin to eliminate potential causes for the problem.
	After the troubleshooter eliminates multiple potential causes for the problem, he is left with one or more causes that are more likely to have resulted in the problem. The troubleshooter hypothesizes what he considers to be the most likely cause of the problem.
	The troubleshooter then tests his hypothesis to confirm or refute his theory about the problem's underlying cause.

Table 1-3 *FCAPS Management Tasks*

Type of Management	Examples of Management Tasks
	Use network management software to collect information from routers and switches. Send an e-mail alert when processor utilization or bandwidth utilization exceeds a threshold of 80 percent. Respond to incoming trouble tickets from the help desk.
	Require logging of any changes made to network hardware or software configurations. Implement a change management system to alert relevant personnel of planned network changes.
	Invoice IP telephony users for their long-distance and international calls. Keeping track of what is being done on the network and when it is being done.
	Monitor network performance metrics for both LAN and WAN links. Deploy appropriate quality of service (QoS) solutions to make the most efficient use of relatively limited WAN bandwidth, while prioritizing mission-critical traffic.
	Deploy firewall, virtual private network (VPN), and intrusion prevention system (IPS) technologies to defend against malicious traffic. Create a security policy dictating rules of acceptable network use. Use an authorization, authentication, and accounting (AAA) server to validate user credentials, assign appropriate user privileges, and log user activity.

Chapter 2

Table 2-2 *Severity Levels*

Severity Level	Name
0	
1	
2	
3	
4	

Severity Level	Name
5	
6	
7	

Table 2-3 *Comparing SNMP and NetFlow*

Technology	Characteristics
SNMP	
NetFlow	

Table 2-4 *Cisco IOS Commands for Hardware Troubleshooting*

Command	Description
	Provides 5-second, 1-minute, and 5-minute CPU utilization statistics, in addition to a listing of processes running on a platform along with each process's utilization statistics
	Displays summary information about processor and I/O memory, followed by a more comprehensive report of memory utilization

Command	Description
	Shows Layer 1 and Layer 2 interface status, interface load information, and error statistics including the following: input queue drops: Indicates a router received information faster than the information could be processed by the router output queue drops: Indicates a router is not able to send information out the outgoing interface because of congestion (perhaps because of an input/output speed mismatch) input errors: Indicates frames were not received correctly (for example, a cyclic redundancy check (CRC) error occurred), perhaps indicating a cabling problem or a duplex mismatch output errors: Indicates frames were not transmitted correctly, perhaps due to a duplex mismatch
Note Prior to collecting statistics, interface counters can be reset using the clear counters command.	
	Displays statistical information about an interface (for example, error statistics), where the information varies for different interface types (for example, the type of connected cable might be displayed for a serial interface and whether it is the DCE side or DTE side of the cable)
	Provides detailed information about a router or switch hardware platform

Chapter 3

Table 3-3 *Commands for Troubleshooting High CPU Utilization*

Command	Description
	Displays the ARP cache for a router. If several entries are in the Incomplete state, you might suspect a malicious scan (for example, a ping sweep) of a subnet, or you have a route pointing out an Ethernet interface as described in our ARP Input process discussion.
	Displays a collection of interface statistics. If the throttles, overruns, or ignored counters continually increment, you might suspect that the Net Background process is attempting to allocate buffer space for an interface from the main buffer pool of the router.
	Provides information about the number of TCP segments a router sends and receives, including the number of connections initiated, accepted, established, and closed. A high number of connections can explain why the TCP Timer process might be consuming excessive CPU resources. If you see an excessive number of embryonic connections, you might be under a denial-of-service (DoS) attack.
	Displays average CPU utilization over 5-second, 1-minute, and 5-minute intervals, in addition to listing all the router processes and the percentage of CPU resources consumed by each of those processes.
	Displays a graphical view of CPU utilization over the past 60 seconds, 1 hour, and 3 days. This graphical view can indicate whether an observed high CPU utilization is a temporary spike in utilization or whether the high CPU utilization is an ongoing condition.

Table 3-4 *Commands for Troubleshooting a Router's Packet-Switching Modes*

Command	Description
	Displays multiple interface statistics, including information about the packet-switching mode of an interface.
	Displays the contents of the route cache from a router if fast switching is enabled.
	Displays information about the IP input process on a router. The CPU utilization for this process might show a high value if the CPU of a router is actively engaged in process-switching traffic because you turned off fast switching and CEF.
	Displays the contents of a router's FIB.
	Displays destinations reachable through the combination of the specified egress interface and next-hop IP address.
	Provides information contained in the adjacency table of a router, including protocol and timer information.

Chapter 4

Table 4-2 *Comparing Trunking Administrative Modes*

SW1	
	Dynamic Auto
	Dynamic Desirable
	Trunk
	Trunk Nonegotiate
	Access
SW2	Dynamic Auto
	Dynamic Desirable
	Trunk
	Trunk Nonegotiate
	Access

Chapter 5

Table 5-2 *STP Port Types*

Port Type	Description
Root port (RP)	
Designated port (DP)	
Nondesignated port (X)	

Table 5-3 *Default Port Costs*

Link Speed	802.1D STP Port Cost	802.1D-2004 STP Port Cost
10 Mbps (Ethernet)		
100 Mbps (Fast Ethernet)		
1 Gbps (Gigabit Ethernet)		
10 Gbps (Ten Gig Ethernet)		
100 Gbps		
1 Tbps		
10 Tbps		

Table 5-4 *EtherChannel Modes That Will Successfully Form a Bundle*

	SW1					
	MODE	PAgP Desirable	PAgP Auto	LACP Active	LACP Passive	ON
SW2	PAgP Desirable					
	PAgP Auto					
	LACP Active					
	LACP Passive					
	ON					

Chapter 6

Table 6-2 *Options for Successfully Forming an EtherChannel*

	SW1					
	MODE	PAgP Desirable	PAgP Auto	LACP Active	LACP Passive	On
SW2	PAgP Desirable					
	PAgP Auto					
	LACP Active					
	LACP Passive					
	On					

Chapter 8

Table 8-2 *Comparing HSRP, VRRP, and GLBP*

Characteristic	HSRP	VRRP	GLBP
Cisco proprietary.			
Interface IP address can act as virtual IP address.			
More than one router in a group can simultaneously forward traffic for that group.			
Hello timer default value.			
Hold timer default value.			
Preemption enabled by default.			
Default priority.			
Default weight.			
Authentication supported.			
Multicast address.			
Virtual MAC address. (xx = group number) (yy = AVF)			

Chapter 9

Table 9-2 *DHCP Message Types*

DHCP Message	Description
	A client sends this message in an attempt to locate a DHCP server. This message is sent to a broadcast IP address of 255.255.255.255 using UDP port 67.
	A DHCP server sends this message in response to a DHCPDISCOVER message using UDP port 68.
	This broadcast message is a request from the client to the DHCP server for the IP addressing information and options that were received in the DHCP Offer message.
	This message is sent from a client to a DHCP server to inform the server that an IP address is already in use on the network.
	A DHCP server sends this message to a client and includes IP configuration parameters.
	A DHCP server sends this message to a client and informs the client that the DHCP server declines to provide the client with the requested IP configuration information.
	A client sends this message to a DHCP server and informs the DHCP server that the client has released its DHCP lease, thus allowing the DHCP server to reassign the client IP address to another client.
	This message is sent from a client to a DHCP server and requests IP configuration parameters. Such a message might be sent from an access server requesting IP configuration information for a remote client attaching to the access server.

Table 9-3 *Types of NAT*

Type of NAT	Description (Based on Private to Public IPv4 Address Translations)
Static NAT	
Dynamic NAT	
NAT overloading or PAT	

Table 9-4 *Names of NAT IP Addresses*

NAT IPs	Definition
	The IP address of a device inside the network; this address will be translated to the inside global address. (Example: PC inside the network)
	The IP address that the Inside local address is translated to. (Example: public IP address used on Internet)
	The IP address of a remote device as it appears to the devices inside the network. This may or may not be the actual address of the remote device if NAT translated it. Note that usually the outside global and outside local addresses are the same.
	The IP address of the device that the inside local address is trying to communicate with. This may be translated to the outside local address (but usually is not translated). (Example: web server)

Chapter 12

Table 12-2 *Default Administrative Distance of Route Sources*

Source of Route Information	AD
Connected interface	
Static route	
EIGRP summary route	
eBGP	
EIGRP (internal)	
OSPF	
IS-IS	
RIP	
EGP	
ODR	
EIGRP (external)	
iBGP	
Unknown (not believable)	

Chapter 15

Table 15-2 *Adjacency States*

State	Description
	This state indicates that no hellos have been received from a neighbor.
	This state occurs after a router sends a unicast hello (as opposed to a multicast hello) to a configured neighbor and has not yet received a hello from that neighbor.
	This state occurs on a router that has received a hello message from its neighbor; however, the OSPF RID of the receiving router was not contained in the hello message. If a router remains in this state for a long period, something is probably preventing that router from correctly receiving hello packets from the neighboring router.

State	Description
2Way	
	This state occurs when the routers forming a full neighbor adjacency decide who will send their routing information first. This is accomplished using the RID. The router with the higher RID becomes the master and the other will become the slave. The master will send the routing information first. In a multiaccess network, the DR and BDR have to be determined first before this state starts. However, the DR does not have to be the master because each master/slave election is on a per-neighbor basis. If a router remains in this state for a long period, a maximum transmission unit (MTU) mismatch could exist between the neighboring routers, or a duplicate OSPF RID might exist.
Exchange	
	Based on the missing link-state database entries identified in the Exchange state, the Loading state occurs when each neighboring router requests the other router to send those missing entries. If a router remains in this state for a long period, a packet might have been corrupted, or a router might have a memory corruption issue. Alternatively, it is possible that such a condition could result from the neighboring routers having an MTU mismatch.
Full	

Table 15-3 *OSPF Network Types and Characteristics*

Type	Default	Neighbors	DR/BDR	Timers
Broadcast				
NBMA (Nonbroadcast)				
Point-to-Point				
Point-to-Multipoint	(Not a default) Optimal for hub-and-spoke topologies (Frame-Relay)			
Point-to-Multipoint Nonbroadcast	(Not a default) Optimal for hub-and-spoke topologies (Frame Relay) that do not support broadcast or multicast traffic			

Table 15-4 *OSPF LSAs*

LSA Type	Description
1	
2	
3	
4	
5	
7	

Table 15-5 *Additional OSPF LSAs for OSPFv3*

LSA Type	Description
8	
9	



<https://t.me/learningnets>

Memory Tables Answer Key

Chapter 1

Table 1-2 *Steps to Diagnose a Problem*

Step	Description
Collect information	Because a typical problem report lacks sufficient information to give a troubleshooter insight into a problem's underlying cause, the troubleshooter should collect additional information, perhaps using network maintenance tools or by interviewing impacted users.
Examine collected information	After collecting sufficient information about a problem, the troubleshooter then examines that information, perhaps comparing the information against previously collected baseline information.
Eliminate potential causes	Based on the troubleshooter's knowledge of the network and his interrogation of collected information, he can begin to eliminate potential causes for the problem.
Propose an hypothesis	After the troubleshooter eliminates multiple potential causes for the problem, he is left with one or more causes that are more likely to have resulted in the problem. The troubleshooter hypothesizes what he considers to be the most likely cause of the problem.
Verify hypothesis	The troubleshooter then tests his hypothesis to confirm or refute his theory about the problem's underlying cause.

Table 1-3 *FCAPS Management Tasks*

Type of Management	Examples of Management Tasks
Fault management	Use network management software to collect information from routers and switches. Send an e-mail alert when processor utilization or bandwidth utilization exceeds a threshold of 80 percent. Respond to incoming trouble tickets from the help desk.
Configuration management	Require logging of any changes made to network hardware or software configurations. Implement a change management system to alert relevant personnel of planned network changes.
Accounting management	Invoice IP telephony users for their long-distance and international calls. Keeping track of what is being done on the network and when it is being done.
Performance management	Monitor network performance metrics for both LAN and WAN links. Deploy appropriate quality of service (QoS) solutions to make the most efficient use of relatively limited WAN bandwidth, while prioritizing mission-critical traffic.
Security management	Deploy firewall, virtual private network (VPN), and intrusion prevention system (IPS) technologies to defend against malicious traffic. Create a security policy dictating rules of acceptable network use. Use an authorization, authentication, and accounting (AAA) server to validate user credentials, assign appropriate user privileges, and log user activity.

Chapter 2

Table 2-2 *Severity Levels*

Severity Level	Name
0	Emergencies
1	Alerts
2	Critical
3	Errors
4	Warnings
5	Notifications
6	Informational
7	Debugging

Table 2-3 *Comparing SNMP and NetFlow*

Technology	Characteristics
SNMP	Collects device statistics (for example, platform resource utilization, traffic counts, and error counts) Uses a pull model (that is, statistics pulled from monitored device by a network management station [NMS]) Available on nearly all enterprise network devices
NetFlow	Collects detailed information about traffic flows Uses a push model (that is, statistics pushed from the monitored device to a NetFlow collector) Available on routers and high-end switches

Table 2-4 *Cisco IOS Commands for Hardware Troubleshooting*

Command	Description
<code>show processes cpu</code>	Provides 5-second, 1-minute, and 5-minute CPU utilization statistics, in addition to a listing of processes running on a platform along with each process's utilization statistics
<code>show memory</code>	Displays summary information about processor and I/O memory, followed by a more comprehensive report of memory utilization
<code>show interfaces</code>	Shows Layer 1 and Layer 2 interface status, interface load information, and error statistics including the following: input queue drops: Indicates a router received information faster than the information could be processed by the router output queue drops: Indicates a router is not able to send information out the outgoing interface because of congestion (perhaps because of an input/output speed mismatch) input errors: Indicates frames were not received correctly (for example, a cyclic redundancy check (CRC) error occurred), perhaps indicating a cabling problem or a duplex mismatch output errors: Indicates frames were not transmitted correctly, perhaps due to a duplex mismatch

Note Prior to collecting statistics, interface counters can be reset using the `clear counters` command.

Command	Description
<code>show controllers</code>	Displays statistical information about an interface (for example, error statistics), where the information varies for different interface types (for example, the type of connected cable might be displayed for a serial interface and whether it is the DCE side or DTE side of the cable)
<code>show platform</code>	Provides detailed information about a router or switch hardware platform

Chapter 3

Table 3-3 *Commands for Troubleshooting High CPU Utilization*

Command	Description
<code>show ip arp</code>	Displays the ARP cache for a router. If several entries are in the Incomplete state, you might suspect a malicious scan (for example, a ping sweep) of a subnet, or you have a route pointing out an Ethernet interface as described in our ARP Input process discussion.
<code>show interface interface_type interface_number</code>	Displays a collection of interface statistics. If the throttles, overruns, or ignored counters continually increment, you might suspect that the Net Background process is attempting to allocate buffer space for an interface from the main buffer pool of the router.
<code>show tcp statistics</code>	Provides information about the number of TCP segments a router sends and receives, including the number of connections initiated, accepted, established, and closed. A high number of connections can explain why the TCP Timer process might be consuming excessive CPU resources. If you see an excessive number of embryonic connections, you might be under a denial-of-service (DoS) attack.
<code>show processes cpu</code>	Displays average CPU utilization over 5-second, 1-minute, and 5-minute intervals, in addition to listing all the router processes and the percentage of CPU resources consumed by each of those processes.

Command	Description
<code>show processes cpu history</code>	Displays a graphical view of CPU utilization over the past 60 seconds, 1 hour, and 3 days. This graphical view can indicate whether an observed high CPU utilization is a temporary spike in utilization or whether the high CPU utilization is an ongoing condition.

Table 3-4 *Commands for Troubleshooting a Router's Packet-Switching Modes*

Command	Description
<code>show ip interface <i>interface_type</i> <i>interface_number</i></code>	Displays multiple interface statistics, including information about the packet-switching mode of an interface.
<code>show ip cache</code>	Displays the contents of the route cache from a router if fast switching is enabled.
<code>show processes cpu include IP Input</code>	Displays information about the IP input process on a router. The CPU utilization for this process might show a high value if the CPU of a router is actively engaged in process-switching traffic because you turned off fast switching and CEF.
<code>show ip cef</code>	Displays the contents of a router's FIB.
<code>show ip cef adjacency <i>egress_interface_id</i> <i>next_hop_ip_address</i> detail</code>	Displays destinations reachable through the combination of the specified egress interface and next-hop IP address.
<code>show adjacency detail</code>	Provides information contained in the adjacency table of a router, including protocol and timer information.

Chapter 4

Table 4-2 *Comparing Trunking Administrative Modes*

SW1						
		Dynamic Auto	Dynamic Desirable	Trunk	Trunk Nonegotiate	Access
SW2	Dynamic Auto	Access	Trunk	Trunk	Limited connectivity	Access
	Dynamic Desirable	Trunk	Trunk	Trunk	Limited connectivity	Access
	Trunk	Trunk	Trunk	Trunk	Trunk	Limited connectivity
	Trunk Nonegotiate	Limited connectivity	Limited connectivity	Trunk	Trunk	Limited connectivity
	Access	Access	Access	Limited connectivity	Limited connectivity	Access

Chapter 5

Table 5-2 *STP Port Types*

Port Type	Description
Root port (RP)	Every nonroot bridge has a single root port (this is mandatory). It is the port on the switch that is closest to the root bridge, in terms of cost, which is inversely proportional to bandwidth by default. If cost is tied the upstream BID is used to break the tie. If the upstream BID is tied, the upstream port ID (PID) is used to break the tie.
Designated port (DP)	Every network segment has a single designated port (this is mandatory). It is the port on the segment that is closest to the root bridge, in terms of cost. If cost is tied, the upstream BID is used to break the tie. If the upstream BID is tied, the upstream port ID (PID) is used to break the tie.

Note Because all ports on the root bridge are as close as you could get to the root bridge, all ports on a root bridge are DPs.

Port Type	Description
Nondesignated port (X)	These are the ports blocking traffic to create a loop-free topology.

Table 5-3 *Default Port Costs*

Link Speed	802.1D STP Port Cost	802.1D-2004 STP Port Cost
10 Mbps (Ethernet)	100	2000000
100 Mbps (Fast Ethernet)	19	200000
1 Gbps (Gigabit Ethernet)	4	20000
10 Gbps (Ten Gig Ethernet)	2	2000
100 Gbps	N/A	200
1 Tbps	N/A	20
10 Tbps	N/A	2

Table 5-4 *EtherChannel Modes That Will Successfully Form a Bundle*

		SW1				
SW2	MODE	PAgP Desirable	PAgP Auto	LACP Active	LACP Passive	ON
	PAgP Desirable	Yes	Yes	No	No	No
	PAgP Auto	Yes	No	No	No	No
	LACP Active	No	No	Yes	Yes	No
	LACP Passive	No	No	Yes	No	No
	ON	No	No	No	No	Yes

Chapter 6

Table 6-2 *Options for Successfully Forming an EtherChannel*

	SW1					
	MODE	PAgP Desirable	PAgP Auto	LACP Active	LACP Passive	On
SW2	PAgP Desirable	Yes	Yes	No	No	No
	PAgP Auto	Yes	No	No	No	No
	LACP Active	No	No	Yes	Yes	No
	LACP Passive	No	No	Yes	No	No
	On	No	No	No	No	Yes

Chapter 8

Table 8-2 *Comparing HSRP, VRRP, and GLBP*

Characteristic	HSRP	VRRP	GLBP
Cisco proprietary.	Yes	No	Yes
Interface IP address can act as virtual IP address.	No	Yes	No
More than one router in a group can simultaneously forward traffic for that group.	No	No	Yes
Hello timer default value.	3 seconds	1 second	3 seconds
Hold timer default value.	10 seconds	3 seconds	10 seconds
Preemption enabled by default.	No	Yes	No for AVG, Yes for AVFs
Default priority.	100	100	100
Default weight.	—	—	100
Authentication supported.	Yes	Yes	Yes
Multicast address.	224.0.0.2	224.0.0.18	224.0.0.102
Virtual MAC address.	V1: 0000.0c07.acxx	0000.5e00.01xx	0007.b400.xxyy
(xx = group number)(yy = AVF)	V2: 0000.0c9f.fxxx		

Chapter 9

Table 9-2 *DHCP Message Types*

DHCP Message	Description
DHCPDISCOVER	A client sends this message in an attempt to locate a DHCP server. This message is sent to a broadcast IP address of 255.255.255.255 using UDP port 67.
DHCPOFFER	A DHCP server sends this message in response to a DHCPDISCOVER message using UDP port 68.
DHCPREQUEST	This broadcast message is a request from the client to the DHCP server for the IP addressing information and options that were received in the DHCP Offer message.
DHCPDECLINE	This message is sent from a client to a DHCP server to inform the server that an IP address is already in use on the network.
DHCPACK	A DHCP server sends this message to a client and includes IP configuration parameters.
DHCPNAK	A DHCP server sends this message to a client and informs the client that the DHCP server declines to provide the client with the requested IP configuration information.
DHCPRELEASE	A client sends this message to a DHCP server and informs the DHCP server that the client has released its DHCP lease, thus allowing the DHCP server to reassign the client IP address to another client.
DHCPINFORM	This message is sent from a client to a DHCP server and requests IP configuration parameters. Such a message might be sent from an access server requesting IP configuration information for a remote client attaching to the access server.

Table 9-3 *Types of NAT*

Type of NAT	Description (Based on Private to Public IPv4 Address Translations)
Static NAT	A one-to-one mapping of private internal IP addresses to public external IP addresses
Dynamic NAT	A dynamic mapping of private internal IP addresses to a pool of public external IP addresses
NAT overloading or PAT	Allows multiple private internal IP addresses to use a single public external IP address by keeping track of Layer 4 port numbers, which make each session unique (that is, Port Address Translation [PAT])

Table 9-4 *Names of NAT IP Addresses*

NAT IPs	Definition
Inside local	The IP address of a device inside the network; this address will be translated to the inside global address. (Example: PC inside the network)
Inside global	The IP address that the Inside local address is translated to. (Example: public IP address used on Internet)
Outside local	The IP address of a remote device as it appears to the devices inside the network. This may or may not be the actual address of the remote device if NAT translated it. Note that usually the outside global and outside local addresses are the same.
Outside global	The IP address of the device that the inside local address is trying to communicate with. This may be translated to the outside local address (but usually is not translated). (Example: web server)

Chapter 12

Table 12-2 *Default Administrative Distance of Route Sources*

Source of Route Information	AD
Connected interface	0
Static route	1
EIGRP summary route	5
eBGP	20
EIGRP (internal)	90
OSPF	110
IS-IS	115
RIP	120
EGP	140
ODR	160
EIGRP (external)	170
iBGP	200
Unknown (not believable)	255

Chapter 15

Table 15-2 *Adjacency States*

State	Description
Down	This state indicates that no hellos have been received from a neighbor.
Attempt	This state occurs after a router sends a unicast hello (as opposed to a multicast hello) to a configured neighbor and has not yet received a hello from that neighbor.
Init	This state occurs on a router that has received a hello message from its neighbor; however, the OSPF RID of the receiving router was not contained in the hello message. If a router remains in this state for a long period, something is probably preventing that router from correctly receiving hello packets from the neighboring router.
2Way	This state occurs when two OSPF routers receive hello messages from each other, and each router sees its own OSPF RID in the hello message it receives. The 2Way state is an acceptable state to stay in between DROthers on an Ethernet LAN.
Exstart	This state occurs when the routers forming a full neighbor adjacency decide who will send their routing information first. This is accomplished using the RID. The router with the higher RID becomes the master and the other will become the slave. The master will send the routing information first. In a multiaccess network, the DR and BDR have to be determined first before this state starts. However, the DR does not have to be the master because each master/slave election is on a per-neighbor basis. If a router remains in this state for a long period, a maximum transmission unit (MTU) mismatch could exist between the neighboring routers, or a duplicate OSPF RID might exist.
Exchange	This state occurs when the two routers forming an adjacency send one another database descriptor (DBD) packets containing information about a router's link-state database. Each router compares the DBD packets received from the other router to identify missing entries in its own link. If a router remains in this state for a long period, an MTU mismatch could exist between the neighboring routers.
Loading	Based on the missing link-state database entries identified in the Exchange state, the Loading state occurs when each neighboring router requests the other router to send those missing entries. If a router remains in this state for a long period, a packet might have been corrupted, or a router might have a memory corruption issue. Alternatively, it is possible that such a condition could result from the neighboring routers having an MTU mismatch.
Full	This state indicates that the neighboring OSPF routers have successfully exchanged their link-state information with one another, and an adjacency has been formed.

Table 15-3 *OSPF Network Types and Characteristics*

Type	Default	Neighbors	DR/BDR	Timers
Broadcast	Default on LAN interfaces	Discovered automatically	DR and BDR elected automatically	Hello 10 Dead 40
NBMA (Nonbroadcast)	Default on Frame Relay main and point-to-multipoint interfaces	Statically configured	DR must be manually configured on the hub router.	Hello 30 Dead 120
Point-to-Point	Default on point to point serial and point-to-point Frame Relay subinterfaces	Discovered automatically	No DR or BDR	Hello 10 Dead 40
Point-to-Multipoint	(Not a default) Optimal for hub-and-spoke topologies (Frame-Relay)	Discovered automatically	No DR or BDR	Hello 30 Dead 120
Point-to-Multipoint Nonbroadcast	(Not a default) Optimal for hub-and-spoke topologies (Frame Relay) that do not support broadcast or multicast traffic	Statically Configured	No DR or BDR	Hello 30 Dead 120

Table 15-4 *OSPF LSAs*

LSA Type	Description
1	All OSPF routers source Type 1 LSAs. These advertisements list information about directly connected subnets, the OSPF connection types of a router, and the known OSPF adjacencies of a router. A Type 1 LSA is not sent out of its local area.
2	The designated router on a multiaccess network sends a Type 2 LSA for that network if the network contains at least two routers. A Type 2 LSA contains a listing of routers connected to the multiaccess network and, like a Type 1 LSA, is constrained to its local area.
3	A Type 3 LSA is sourced by an ABR. Each Type 3 LSA sent into an area contains information about a network reachable in a different area. Note that network information is exchanged only between the backbone area and a nonbackbone area, as opposed to being exchanged between two nonbackbone areas.

LSA Type	Description
4	Similar to a Type 3 LSA, a Type 4 LSA is sourced by an ABR. However, instead of containing information about OSPF networks, a Type 4 LSA contains information stating how to reach an ASBR.
5	A Type 5 LSA is sourced by an ASBR and contains information about networks reachable outside the OSPF domain. A Type 5 LSA is sent to all OSPF areas, except for stub areas. Note that the ABR for a stub area sends default route information into the stub area, rather than the network-specific Type 5 LSAs.
7	A Type 7 LSA is sourced from an ASBR within a not-so-stubby area (NSSA). Whereas a stub area cannot connect to an external autonomous system, an NSSA can. The Type 7 LSA only exists in the NSSA; therefore, the external routes are announced by the ABR(s) of the NSSA into Area 0 using Type 5 LSAs. In addition, like a stub area external routes known to another OSPF area are not forwarded into an NSSA since Type 5 LSAs are not permitted in an NSSA.

Table 15-5 *Additional OSPF LSAs for OSPFv3*

LSA Type	Description
8	This LSA type (Link) provides information to neighbors about link-local addresses and the IPv6 addresses associated with the link. Therefore, it is only flooded on the local link and will not be reflooded by other OSPF routers.
9	This LSA type (Intra Area Prefix) provides information for two different scenarios. 1) It will provide information about IPv6 address prefixes associated with a transit network by referencing a Network LSA. 2) It will provide information about IPv6 address prefixes associated with a router by referencing a Router LSA. Type 9 LSAs are only flooded within an area.

Appendix E

Study Planner

Practice Test	Reading	Task

Element	Task	Goal Date	First Date Completed	Second Date Completed (Optional)
Introduction	Read Introduction			
1. Introduction to Troubleshooting and Network Maintenance	Read Foundation Topics			
1. Introduction to Troubleshooting and Network Maintenance	Review Key Topics			
1. Introduction to Troubleshooting and Network Maintenance	Define Key Terms			
1. Introduction to Troubleshooting and Network Maintenance	Review Command Reference			
2. Troubleshooting and Maintenance Tools	Read Foundation Topics			
2. Troubleshooting and Maintenance Tools	Review Key Topics			
2. Troubleshooting and Maintenance Tools	Define Key Terms			
2. Troubleshooting and Maintenance Tools	Review Memory Tables			
2. Troubleshooting and Maintenance Tools	Review Command Reference			
3. Troubleshooting Device Performance	Read Foundation Topics			
3. Troubleshooting Device Performance	Review Key Topics			
3. Troubleshooting Device Performance	Define Key Terms			
3. Troubleshooting Device Performance	Review Memory Tables			
3. Troubleshooting Device Performance	Review Command Reference			
Part I Review	Take practice test in study mode using Exam Bank #1 questions for chapters 1-3 in practice test software			
4. Troubleshooting Layer 2 Trunks, VTP, and VLANs	Read Foundation Topics			
4. Troubleshooting Layer 2 Trunks, VTP, and VLANs	Review Key Topics			
4. Troubleshooting Layer 2 Trunks, VTP, and VLANs	Define Key Terms			
4. Troubleshooting Layer 2 Trunks, VTP, and VLANs	Review Memory Tables			
4. Troubleshooting Layer 2 Trunks, VTP, and VLANs	Review Command Reference			
5. Troubleshooting STP and Layer 2 EtherChannel				

5. Troubleshooting STP and Layer 2 EtherChannel	Review Key Topics			
5. Troubleshooting STP and Layer 2 EtherChannel	Define Key Terms			
5. Troubleshooting STP and Layer 2 EtherChannel	Review Memory Tables			
5. Troubleshooting STP and Layer 2 EtherChannel	Review Command Reference			
6. Troubleshooting InterVLAN Routing and Layer 3 EtherChannels	Read Foundation Topics			
6. Troubleshooting InterVLAN Routing and Layer 3 EtherChannels	Review Key Topics			
6. Troubleshooting InterVLAN Routing and Layer 3 EtherChannels	Define Key Terms			
6. Troubleshooting InterVLAN Routing and Layer 3 EtherChannels	Review Memory Tables			
6. Troubleshooting InterVLAN Routing and Layer 3 EtherChannels	Review Command Reference			
7. Troubleshooting Switch Security Features				
7. Troubleshooting Switch Security Features	Review Key Topics			
7. Troubleshooting Switch Security Features	Define Key Terms			
7. Troubleshooting Switch Security Features	Review Command Reference			
8. Troubleshooting FHRP	Read Foundation Topics			
8. Troubleshooting FHRP	Review Key Topics			
8. Troubleshooting FHRP	Define Key Terms			
8. Troubleshooting FHRP	Review Memory Tables			
8. Troubleshooting FHRP	Review Command Reference			
Part II Review	Take practice test in study mode using Exam Bank #1 questions for chapters 4-8 in practice test software			
9. Troubleshooting IPv4 Addressing and Addressing Technologies	Read Foundation Topics			
9. Troubleshooting IPv4 Addressing and Addressing Technologies	Review Key Topics			
9. Troubleshooting IPv4 Addressing and Addressing Technologies	Define Key Terms			
9. Troubleshooting IPv4 Addressing and Addressing Technologies	Review Memory Tables			
9. Troubleshooting IPv4 Addressing and Addressing Technologies	Review Command Reference			
10. Troubleshooting IPv6 Addressing and Addressing Technologies	Read Foundation Topics			
10. Troubleshooting IPv6 Addressing and Addressing Technologies	Review Key Topics			
10. Troubleshooting IPv6 Addressing and Addressing Technologies	Define Key Terms			
10. Troubleshooting IPv6 Addressing and Addressing Technologies	Review Command Reference			
11. Troubleshooting IPv4 and IPv6 ACLs and Prefix-Lists	Read Foundation Topics			

11. Troubleshooting IPv4 and IPv6 ACLs and Prefix-Lists	Review Key Topics			
11. Troubleshooting IPv4 and IPv6 ACLs and Prefix-Lists	Define Key Terms			
11. Troubleshooting IPv4 and IPv6 ACLs and Prefix-Lists	Review Command Reference			
12. Troubleshooting Basic IPv4/IPv6 Routing and GRE Tunnels	Read Foundation Topics			
12. Troubleshooting Basic IPv4/IPv6 Routing and GRE Tunnels	Review Key Topics			
12. Troubleshooting Basic IPv4/IPv6 Routing and GRE Tunnels	Define Key Terms			
12. Troubleshooting Basic IPv4/IPv6 Routing and GRE Tunnels	Review Memory Tables			
12. Troubleshooting Basic IPv4/IPv6 Routing and GRE Tunnels	Review Command Reference			
13. Troubleshooting RIP	Read Foundation Topics			
13. Troubleshooting RIP	Review Key Topics			
13. Troubleshooting RIP	Define Key Terms			
13. Troubleshooting RIP	Review Command Reference			
14. Troubleshooting EIGRP	Read Foundation Topics			
14. Troubleshooting EIGRP	Review Key Topics			
14. Troubleshooting EIGRP	Define Key Terms			
14. Troubleshooting EIGRP	Review Command Reference			
15. Troubleshooting OSPF	Read Foundation Topics			
15. Troubleshooting OSPF	Review Key Topics			
15. Troubleshooting OSPF	Define Key Terms			
15. Troubleshooting OSPF	Review Memory Tables			
15. Troubleshooting OSPF	Review Command Reference			
16. Troubleshooting Route Maps and PBR	Read Foundation Topics			
16. Troubleshooting Route Maps and PBR	Review Key Topics			
16. Troubleshooting Route Maps and PBR	Define Key Terms			
16. Troubleshooting Route Maps and PBR	Review Command Reference			
17. Troubleshooting Redistribution	Read Foundation Topics			
17. Troubleshooting Redistribution	Review Key Topics			
17. Troubleshooting Redistribution	Define Key Terms			
17. Troubleshooting Redistribution	Review Command Reference			
18. Troubleshooting BGP	Read Foundation Topics			
18. Troubleshooting BGP	Review Key Topics			
18. Troubleshooting BGP	Define Key Terms			
18. Troubleshooting BGP	Review Command Reference			

Part III Review	Take practice test in study mode using Exam Bank #1 questions for chapters 9-18 in practice test software			
19. Troubleshooting Remote Connectivity	Read Foundation Topics			
19. Troubleshooting Remote Connectivity	Review Key Topics			
19. Troubleshooting Remote Connectivity	Define Key Terms			
19. Troubleshooting Remote Connectivity	Review Command Reference			
20. Troubleshooting Management Access	Read Foundation Topics			
20. Troubleshooting Management Access	Review Key Topics			
20. Troubleshooting Management Access	Define Key Terms			
20. Troubleshooting Management Access	Review Command Reference			
Part IV Review	Take practice test in study mode using Exam Bank #1 questions for chapters 19-20 in practice test software			
21. Additional Trouble Tickets	Review and resolve trouble ticket #1			
21. Additional Trouble Tickets	Review and resolve trouble ticket #2			
21. Additional Trouble Tickets	Review and resolve trouble ticket #3			
21. Additional Trouble Tickets	Review and resolve trouble ticket #4			
21. Additional Trouble Tickets	Review and resolve trouble ticket #5			
21. Additional Trouble Tickets	Review and resolve trouble ticket #6			
21. Additional Trouble Tickets	Review and resolve trouble ticket #7			
21. Additional Trouble Tickets	Review and resolve trouble ticket #8			
21. Additional Trouble Tickets	Review and resolve trouble ticket #9			
21. Additional Trouble Tickets	Review and resolve trouble ticket #10			
22. Final Preparation	Read Chapter			
22) Final Review	Take practice test in study mode for all Book Questions in practice test software			
22. Final Preparation	Review all Key Topics in all chapters			
22. Final Preparation	Complete all memory tables from appendix C			
22. Final Preparation	Review all Command Reference Tables in all chapters			

22. Final Preparation	Take practice test in practice exam mode using Exam Bank #2 questions for all chapters			



<https://t.me/learningnets>

224.0.0.10 The multicast IPv4 used by EIGRP routers to form a neighbor adjacency.

224.0.0.5 The All OSPF Routers multicast IPv4 address, listened for by all OSPF routers.

224.0.0.6 The All OSPF DR Routers multicast IPv4 address, listened to by DR and BDR routers.

2Way (OSPF) A neighbor state that signifies the other router has reached neighbor status, having passed the parameter check.

802.1D An IEEE standard for Spanning Tree Protocol.

802.1Q A method of passing frames and their VLAN associations over a trunk link, based on the IEEE 802.1Q standard.

802.1s An IEEE standard for Multiple Spanning Tree Protocol.

802.1w An IEEE standard for Rapid Spanning Tree Protocol.

ABR See *Area Border Router*.

access port Ports on a switch that typically connect to end stations that will never form a trunk.

ACK (EIGRP) An EIGRP message that is used to acknowledge reliable EIGRP messages (namely update, query, and reply messages). ACK messages do not require acknowledgment with an ACK message.

ACL (access control list) A list containing entries configured on a router or switch that can be used to identify traffic that will have a particular action applied to it based on the service or feature that is using the list.

active (BGP state) A BGP neighbor state in which the TCP connection has successfully completed, but the BGP neighbors have not yet agreed to exchange path information.

active (EIGRP) A state for a route in an EIGRP topology table that indicates that the router is actively sending query messages for this route, attempting to validate and learn the current best route to that subnet.

active forwarder See *HSRP active forwarder*.

active virtual forwarder (AVF) A GLBP router that takes on a virtual MAC address and forwards traffic received on that address.

active virtual gateway (AVG) The GLBP router that answers all ARP requests for the virtual router address and assigns virtual MAC addresses to each router in the GLBP group.

address family (named EIGRP/OSPFv3/MP-BGP) A method of configuring IPv4 and IPv6 routing services under the same routing process. IPv4 address families are used for IPv4 routing, and IPv6 address families are used for IPv6 routing.

address families See *address family*.

adjacency table A table used by CEF that stores the Layer 2 addressing for all FIB entries of next-hop devices.

administrative distance In Cisco routers, a means for one router to choose between multiple routes to reach the same subnet when those routes are learned by different routing protocols. The lower the administrative distance, the more preferred the source of the routing information.

ADVERTISE message DHCPv6 servers respond to SOLICIT messages with a unicast ADVERTISE message offering addressing information to the DHCPv6 client.

advertised distance See *reported distance*.

aggregate route Another term for summary route.

alternate port (RSTP) A port on a switch other than the root port that has an alternative path to the root bridge.

anycast An IPv6 address type that is used by a number of hosts in a network that are providing the same service. Hosts accessing the service are routed to the nearest host in an anycast environment based routing protocol metrics.

APIPA See *Automatic Private IP Addressing*.

archive A Cisco IOS feature that is used to create automatic archives of device configurations.

Area Border Router (ABR) A router that has interfaces connected to at least two different OSPF areas, one of which must be the backbone area. ABRs hold topology data for each area, and calculate routes for each area, and advertise about those routes between areas.

area A grouping of routers and router interfaces, typically contiguous. Routers in an area strive to learn all topology information about the area, and do not learn topology information about areas to which they do not connect.

ARP (Address Resolution Protocol) Defined in RFC 826, a protocol used on an Ethernet LAN by devices to determine the Layer 2 MAC address of a known Layer 3 IP address.

ARP cache A table that Ethernet-enabled devices use to maintain the IPv4 to MAC address mappings.

ARP input process A process in charge of sending ARP requests on a router.

ARP poisoning Also known as ARP spoofing. An attack whereby an attacker sends specially crafted ARP replies so that its own MAC address appears as the gateway or some other targeted host. From that time on, unsuspecting clients unknowingly send traffic to the attacker.

AS_PATH access list A Cisco IOS configuration tool, using the `ip as-path access-list` command that defines a list of statements that match the AS_PATH BGP path attribute using regular expressions.

AS_PATH prepending This term has two BGP-related definitions. First, it is the normal process in which a router, before sending an update to an eBGP peer, adds its local ASN to the beginning of the AS_PATH path attribute. Second, it is the routing policy of purposefully adding one or more ASNs to the beginning of a route's AS_PATH path attribute, typically to lengthen the AS_PATH and make the route less desirable in the BGP decision process.

AS_PATH A BGP path attribute that lists ASNs through which the route has been advertised. The AS_PATH includes four types of segments: AS_SEQ, AS_SET, AS_CONFED_SEQ, and AS_CONFED_SET. Often, this term is used synonymously with AS_SEQ.

ASBR (Autonomous System Border Router) A router using OSPF in which the router learns routes via another source, typically another routing protocol, exchanging routes that are external to OSPF with the OSPF domain.

ASBR-Summary Link-state advertisement (LSA). See Type 4 LSA.

asymmetric routing A routing condition where packets take one path when traveling from a source device to a destination device, but return traffic takes a different path.

authentication, authorization, and accounting (AAA) A security feature that allows a router to authenticate user credentials, determine what a user is allowed to do, and keep an audit trail of what the user did.

authentication With routing protocols, the process by which the router receiving a routing update determines whether the routing update came from a trusted router.

authNoPriv An SNMP security model used with SNMPv3 to provide improved authentication using MD5 or SHA hashing algorithms.

authPriv An SNMP security model used with SNMPv3 to provide improved authentication using MD5 or SHA hashing algorithms and privacy using encryption algorithms such as DES, 3DES, and AES.

Automatic Private IP Addressing An IPv4 addressing method used by DHCPv4 clients when the DHCPv4 server is not available. The clients automatically assign themselves an IPv4 address in the 169.254.0.0/16 network.

autonegotiation A mechanism used by a device and a switchport to automatically negotiate the link speed and duplex mode.

autonomous system In BGP, a set of routers inside a single administrative authority, grouped together for the purpose of controlling routing policies for the routes advertised by that group to the Internet.

Autonomous System Border Router See *ASBR*.

autonomous system number (ASN) A number between 1 and 64,511 (public) and 64,512 and 65,535 (private) assigned to an autonomous system for the purpose of proper BGP operation.

autosummarization A routing protocol feature in which a router that connects to more than one classful network advertises summarized routes for each entire classful network when sending updates out interfaces connected to other classful networks.

backbone area (OSPF) Area 0; the area to which all other OSPF areas must connect for OSPF to function properly.

backbone router Any OSPF router that has at least one interface connected to the backbone area.

BackboneFast An STP feature that can detect an indirect link failure and shorten the STP convergence time to 30 seconds by bypassing the max age timeout period.

backplane Physically interconnects a switch's ports. Therefore, depending on the specific switch architecture, frames flowing through a switch enter via a port (that is, an ingress port), flow across the switch's backplane, and are forwarded out of another port (that is, an egress port).

backup designated router (BDR) In OSPF, a router that is prepared to take over the designated router.

backup port (RSTP) A port that provides a redundant (but less desirable) connection to a segment where another switchport already connects.

bandwidth 1) The rate at which bits are sent on an interface. 2) The Cisco IOS Software setting, per the **bandwidth** command, that tells the Cisco IOS the speed of the interface.

baseline A collection of network measurements taken when the network is functioning properly. Measurements taken during a troubleshooting scenario could be contrasted with baseline information.

BDR See *backup designated router*.

best path algorithm A set of rules by which BGP examines the details of multiple BGP routes for the same NLRI and chooses the single best BGP route to install in the local BGP table.

BGP decision process See *best path algorithm*.

BGP hard reset The process of restarting a BGP neighbor relationship by closing the TCP connection, causing both neighboring routers to remove all paths formerly learned from that neighbor from their respective BGP tables.

BGP neighbor table Contains a listing of all BGP neighbors configured for a router, including each neighbor's IP address, the ASN, the state of the neighbor relationship, and several other statistics.

BGP peer group In BGP, a configuration construct in which multiple neighbors' parameters can be configured as a group, thereby reducing the length of the configuration. In addition, BGP performs routing policy logic against only one set of Updates for the entire peer group, improving convergence time.

BGP peer Another name for a BGP neighbor. A BGP neighbor is another router running BGP with which the local router has formed a BGP neighbor relationship for the purpose of exchanging BGP Updates.

BGP soft reset The process of restarting a BGP neighbor relationship without closing the underlying TCP connection, instead resending full updates to the neighbor, and asking for the neighbor to send a full update again.

BGP synchronization In BGP, a feature in which BGP routes cannot be considered to be a best route to reach an NLRI unless that same prefix exists in the router's IP routing table as learned via some IGP.

BGP table A table inside a router that holds the path attributes and NLRI known by the BGP implementation on that router.

BGP update A BGP message that includes withdrawn routes, path attributes, and NLRI.

BGP See *Border Gateway Protocol*.

blocking One of four Spanning Tree Protocol (STP) states for a port. A port remains in the blocking state for 20 seconds by default. During this time, a nondesignated port evaluates bridge protocol data units (BPDUs) in an attempt to determine its role in a spanning tree. If it is determined that it must stay in the blocking state to prevent a loop, it will.

Border Gateway Protocol (BGP) An exterior routing protocol designed to exchange prefix information between different autonomous systems. The information includes a rich set of characteristics called path attributes, which in turn allows for great flexibility regarding routing choices.

bottom-up method A troubleshooting method where troubleshooting starts at the bottom (that is, Layer 1) of the OSI model and works its way up.

boundary router A router that sits at the boundary of the routing domains and performs redistribution.

BPDU Filter An STP feature that prevents BPDUs from being sent or processed on a switchport.

BPDU Guard An STP feature that disables a switchport if any BPDU is received.

BPDU Bridge protocol data unit; the data message exchanged by switches participating in the Spanning Tree Protocol.

bridging loop A condition where Ethernet frames are forwarded endlessly around a Layer 2 loop formed between switches in a redundant topology.

broadcast domain The extent of a network where a single broadcast frame or packet will be seen.

buffer leak A buffer leak occurs when a process does not return a buffer to the router when the process has finished using the buffer.

CAM Content-addressable memory; the high-performance table used by a switch to correlate MAC addresses with the switch interfaces where they can be found.

Carrier sense multiple access collision detect (CSMA/CD) A mechanism used on Ethernet networks to detect collisions and cause transmitting devices to back off for a random time.

Challenge Handshake Authentication Protocol (CHAP) A security feature defined by PPP that allows either or both endpoints on a link to authenticate the other device as a particular authorized device.

change management The process of controlling how alterations are managed to minimize downtime within the organization.

CHAP See *Challenge Handshake Authentication Protocol*.

CIDR notation See *prefix notation*.

CIDR See *classless interdomain routing*.

Cisco Discovery Protocol (CDP) A Cisco proprietary protocol used to advertise and discover directly connected devices automatically.

Cisco Express Forwarding (CEF) An optimized Layer 3 forwarding path through a router or switch. CEF optimizes routing table lookup by creating a special, easily searched tree structure based on the contents of the IP routing table. The forwarding information is called the Forwarding Information Base (FIB), and the cached adjacency information is called the adjacency table.

Cisco Lifecycle Services An approach to the implementation of Cisco technologies, as defined by Cisco.

Cisco TAC A technical assistance center (resource that requires a contract) provided by Cisco to assist you with troubleshooting issues related to their products.

classful IP addressing A convention for discussing and thinking about IP addresses by which Class A, B, and C default network prefixes (of 8, 16, and 24 bits, respectively) are considered.

classful network An IPv4 Class A, B, or C network. It is called a classful network because these networks are defined by the class rules for IPv4 addressing.

classful routing protocol An inherent characteristic of a routing protocol. Specifically, the routing protocol does not send subnet masks in its routing updates. This requires the protocol to make assumptions about classful networks and makes it unable to support VLSM and manual route summarization.

classful routing A type of logic for how a router uses a default route. When a default route exists, and the Class A, B, or C network for the destination IP address does not exist in the routing table, the default route is used. If any part of that classful network exists in the routing table, but the packet does not match any existing subnet of that classful network, the packet does not match the default route and thus is discarded.

classless addressing A concept in IPv4 addressing that defines a subnetted IP address as having two parts: a prefix (or subnet) and a host.

classless interdomain routing (CIDR) Defined in RFCs 1517–1520, a scheme to help reduce Internet routing table sizes by administratively allocating large blocks of consecutive classful IP network numbers to ISPs for use in different global geographies. CIDR results in large blocks of networks that can be summarized, or aggregated, into single routes.

classless IP addressing A convention for IP addresses in which Class A, B, and C default network prefixes (of 8, 16, and 24 bits, respectively) are ignored.

classless routing protocol An inherent characteristic of a routing protocol. Specifically, the routing protocol sends subnet masks in its routing updates, thereby removing any need to make assumptions about the addresses in a particular subnet or network. This allows the protocol to support VLSM and manual route summarization.

CLI (command-line interface) The primary method of interacting with the Cisco IOS on routers and switches using commands.

collision domain The extent within a network that an Ethernet collision will be noticed or experienced.

Common Spanning Tree (CST) A single instance of STP defined in the IEEE 802.1D standard.

community string A plain-text password that is used to authenticate an SNMP NMS and an SNMP agent.

community VLAN A type of secondary private VLAN; switchports associated with the same community VLAN can communicate with each other.

comparing configurations method The comparing configurations method of troubleshooting compares a known good configuration with a current configuration. The difference in those configurations might give the troubleshooter insight into the underlying cause of a problem.

component swapping method The component swapping method of troubleshooting replaces individual network components (for example, a cable, switch, or router) in an attempt to isolate the cause of a problem.

configure replace A Cisco IOS feature that allows for the running configuration to be completely replaced with an archived configuration so a merge does not occur.

console The physical port on Cisco IOS devices that can be used for management purposes that requires a console or rollover cable.

contiguous network In IPv4, an internetwork design in which packets forwarded between any two subnets of a single classful network only pass through the subnets of that classful network.

control plane The control plane of operation encompasses protocols used between routers and switches. These protocols include, for example, routing protocols and Spanning Tree Protocol (STP). Also, a router or switch's processor and memory reside in the control plane.

convergence The time required for routing protocols to react to changes in the network, removing bad routes and adding new, better routes so that the current best routes are in all the routers' routing tables.

CSU/DSU (channel service unit/data service unit) A device that connects a physical circuit installed by the telco to some CPE device, adapting between the voltages, current, framing, and connectors used on the circuit to the physical interface supported by the DTE.

data communications equipment (DCE) From a physical layer perspective, the device providing the clocking on a WAN link, typically a CSU/DSU, is the DCE. From a packet-switching perspective, the service provider's switch, to which a router might connect, is considered the DCE.

data plane In IP routing, a term referring to a set of processes that forward packets through a router or a multilayer switch.

Database Description (DBD) A type of OSPF packet used to exchange and acknowledge LSA headers.

data-link connection identifier (DLCI) A Frame Relay address used in Frame Relay headers to identify the virtual circuit.

DBD See *Database Description*.

DCE See *data communications equipment*.

dead interval With OSPF, the timer used to determine when a neighboring router has failed, based on a router not receiving any OSPF messages, including Hellos, in this timer period. Also called the dead timer.

default route A route that is used for forwarding packets when the packet does not match any more specific routes in the IP routing table.

delay A Cisco IOS Software setting, per the **delay** command, that defines to the router an estimate of the time that a packet is expected to spend trying to exit a router interface. The **delay** command uses a unit of tens of microseconds.

designated port (STP) The port on a segment that receives and forwards frames to the root bridge. This port is the port on the segment that is closest to the root bridge based on cost.

designated router (DR) On multiaccess data links such as LANs, an OSPF router elected by the routers on that data link to perform special functions. These functions include the generation of LSAs representing the subnet and playing a key role in the database exchange process.

destination MAC address The MAC address of the recipient of a frame.

DHCP relay agent A multilayer switch or router that intercepts and relays DHCP negotiation messages between a client and a DHCP server on different VLANs/subnets.

DHCP snooping (trusted port) A port that is able to receive all types of DHCP messages and typically connects to where the DHCP server is located.

DHCP snooping (untrusted port) A port that is not able to receive DHCP Discover or DHCP Request messages. The default for all ports when DHCP snooping is enabled.

DHCP snooping A security feature that enables a switch to intercept all DHCP requests coming from untrusted switchports before they are flooded to unsuspecting users.

DHCP See *Dynamic Host Configuration Protocol*.

DHCPACK A DHCPv4 unicast message used by the DHCPv4 server to acknowledge that the addressing information is reserved for the client.

DHCPDISCOVER A DHCPv4 broadcast message used by a client to locate a DHCPv4 server.

DHCPOFFER A DHCPv4 unicast message used by a DHCPv4 server to provide a client with addressing information.

DHCPREQUEST A DHCPv4 broadcast message used by a client to request the addressing information that was provided in the offer.

DHCPv4 relay agent A device such as a router or multilayer switch that is able to relay DHCPv4 DISCOVER messages to a DHCPv4 server in a different IPv4 network.

DHCPv6 relay agent A device such as a router or multilayer switch that is able to relay DHCPv6 SOLICIT messages to a DHCPv6 server in a different IPv6 network.

DHCPv6 A DHCP service that is compatible with IPv6 clients; a switch can assign IPv6 addresses and advertise DHCP-related options.

Diffusing Update Algorithm (DUAL) A convergence algorithm used in EIGRP that provides loop-free operation at every instance throughout a route computation. Allows routers involved in a topology change to synchronize at the same time, while not involving routers that are unaffected by the change.

Dijkstra Shortest Path First (SPF) algorithm The name of the algorithm used by link-state.

Dijkstra Alternative name for the SPF algorithm, named for its inventor, Edsger W. Dijkstra.

discarding state (RSTP) In this state, incoming frames on a port are dropped and no MAC addresses are learned.

discontiguous network In IPv4, an internetwork design in which packets forwarded between two subnets of a single classful network must pass through the subnets of another classful network.

distance vector The logic behind the behavior of some interior routing protocols, such as RIP and IGRP, characterized by routers sending brief information about a subnet, and a metric (vector) describing how far away that subnet is. Distance vector routing algorithms call for each router to send its entire routing table in each periodic update, but only to its neighbors. Distance vector routing algorithms can be prone to routing loops but are computationally simpler than link-state routing algorithms. Also called Bellman-Ford routing algorithm.

distribute list A Cisco IOS configuration tool for routing protocols by which routing updates may be filtered.

divide-and-conquer method A method of troubleshooting where the troubleshooting begins in the middle (for example, Layer 3) of the OSI model and radiates out from that layer.

DLCI See *data-link connection identifier*.

documentation The process of recording the physical and logical components of your network as well as all the adds, moves, and changes that occur in the network.

DORA The DHCP process a client and server use to determine the appropriate IPv4 addressing information the client needs. (Discover, Offer, Request, Ack).

DR election (OSPF) The process by which neighboring OSPF routers examine their Hello messages and elect the DR. The decision is based on priority (highest), or RID (highest) if priority is a tie.

DR See *designated router*.

DROther The term to describe a router that is neither the DR nor the BDR on a subnet that elects a DR and BDR.

dual stack In IPv6, a mode of operation in which a host or router runs both IPv4 and IPv6.

DUAL See Diffused Update Algorithm.

duplex mismatch A condition where the devices on each end of a link use conflicting duplex modes.

duplex mode The Ethernet mode that governs how devices can transmit over a connection. See *half-duplex* and *full-duplex*.

duplicate address detection (DAD) An IPv6 mechanism through which a host can determine whether another active host on the same local link is trying to use the same IPv6 address.

dynamic ARP inspection (DAI) A security feature that can mitigate ARP-based attacks. ARP replies received on untrusted switchports are checked against known, good values contained in the DHCP snooping database.

dynamic auto An automatic trunking method that uses DTP to negotiate the formation of a trunk. This method will wait for DTP messages to arrive requesting to form a trunk.

dynamic desirable An automatic trunking method that uses DTP to negotiate the formation of a trunk. This method will attempt to form a trunk by sending DTP messages and will respond to DTP messages sent from other devices.

Dynamic Host Configuration Protocol (DHCP) A standard (RFC 2131) protocol by which a host can dynamically broadcast a request for a server to assign to it an IP address, along with other configuration settings, including a subnet mask and default gateway IP address.

Dynamic Multipoint VPN (DMVPN) A virtual private network (VPN) technology that allows a tunnel to be set up or torn down between two sites on an as-needed basis.

Dynamic NAT (DNAT) A version of Network Address Translation (NAT), where inside local addresses are dynamically assigned an inside global address from a pool of available addresses.

Dynamic Trunking Protocol (DTP) A Cisco proprietary method of negotiating a trunk link between two switches.

E1 route (OSPF) An OSPF external route for which internal OSPF cost is added to the cost of the route as it was redistributed into OSPF.

E2 route (OSPF) An OSPF external route for which internal OSPF cost is not added to the cost of the route as it was redistributed into OSPF.

eBGP multihop A BGP feature that defines the IP TTL field value in packets sent between two eBGP peers. This feature is required when using IP addresses other than the interface IP address on the link between peers.

eBGP See *External BGP*.

edge port A switchport STP mode that immediately transitions the port to the forwarding state bypassing the listening and learning states. If the link flaps, no TCNs will be generated.

EGP See *Exterior Gateway Protocol*.

egress port The port a frame will be sent out.

EIGRP (Enhanced Interior Gateway Routing Protocol) An advanced version of IGRP developed by Cisco. Provides superior convergence properties and operating efficiency and combines the advantages of link-state protocols with those of distance-vector protocols. Uses the DUAL algorithm.

EIGRP for IPv6 An interior routing protocol for IPv6 based on the original EIGRP protocol for IPv4.

EIGRP stub router A router running EIGRP that limits itself in several different ways for the purpose of limiting EIGRP DUAL algorithm computations and reducing EIGRP Query scope.

Embedded Event Manager (EEM) The EEM feature can create custom event definitions on a router and specify actions the router can take in response to these events.

encapsulation The process of adding header information and possibly trailer information at different layers of the OSI model depending on the protocol.

end-to-end VLAN A single VLAN that spans the entire switched network, from one end to the other.

err-disabled The status of a port when an issue has occurred that places the port in the err-disabled state. Examples of features that use this include port security, BPDU Guard, UDLD.

established A BGP neighbor state in which the BGP neighbors have stabilized and can exchange routing information using BGP Update messages.

EtherChannel A logical link made up of bundled or aggregated Layer 2 or Layer 3 physical links.

EUI-64 A specification for the 64-bit interface ID in an IPv6 address, composed of the first half of a MAC address (with the seventh bit flipped), added hex values of FFFE, followed by the last half of the MAC address.

extended ACL An ACL that is able to match packets based on multiple criteria such as source and destination IP address, source and destination port numbers, protocols, and QoS parameters.

Exterior Gateway Protocol (EGP) A routing protocol that was designed to exchange routing information between different autonomous systems. EGP has been replaced by BGP and is no longer supported in Cisco IOS.

External BGP A term referring to how a router views a BGP peer relationship, in which the peer is in another AS.

External LSA In OSPF, an LSA that represents a subnet that OSPF learned from another (external) routing source, typically through route redistribution by an ASBR.

external route A characteristic of a route, as defined by a particular routing protocol, that means that the route was learned by that routing protocol through the route redistribution process.

External Type 1 See *E1 route*.

External Type 2 See *E2 route*.

fast switching A router and multilayer switch packet switching mode that makes use of a route cache maintained in a router's data plane. The route cache contains information about how traffic from different data flows should be forwarded. The first packet in a data flow is process switched by a router's CPU. Once the router determines how to forward the first frame of a data flow, that forwarding information is then stored in the route cache. Subsequent packets in that same data flow are then forwarded based on information in the route cache, as opposed to being process switched. As a result, fast switching reduces a router's CPU utilization, as compared to process switching.

FCAPS FCAPS is a network management model defined by the ISO, where the acronym FCAPS stands for fault management, configuration management, accounting management, performance management, and security management.

FD See *feasible distance*.

feasibility condition With EIGRP, to be a feasible successor, the reported distance must be lower than the feasible distance of the successor.

feasible distance The name of the EIGRP metric, which defines how far a destination network is away from the local device. Lower is better. It is a combination of the reported distance from a neighbor and the distance to reach that neighbor.

feasible successor With EIGRP, a route that is not a successor route but that meets the feasibility condition and can be used when the successor route fails, without causing loops.

FIB See *Forwarding Information Base*.

flash updates See *triggered updates*.

floating static route A static route configured with an administrative distance greater than a routing protocol on that same router, resulting in the static route floating into the routing table when the routing protocol's learned route fails.

flooding (frame) An Ethernet frame is replicated and sent out every available switchport in the same VLAN.

flooding (OSPF) The process of exchanging LSA information throughout an area, by having a router send the LSAs to their neighbors who in turn send the LSAs to their neighbors, and so on.

following the traffic path method A troubleshooting method whereby the troubleshooting process will check components (for example, links and devices) over which traffic flows on its way from source to destination.

forward delay The time interval that a switch spends in the Listening and Learning states; default 15 seconds.

forwarding (STP) One of four STP states for a port. A port moves from the learning state to the forwarding state and begins to forward frames.

Forwarding Information Base A CEF database that contains Layer 3 information, similar to the information found in an IP routing table. In addition, an FIB contains information about multicast routes and directly connected hosts.

forwarding logic The process of determining how the Cisco IOS device will handle the traffic received.

Frame Relay Inverse ARP Defined in RFC 1293, this protocol enables a Frame Relay-attached device to react to a received LMI “PVC up” message by announcing its Layer 3 addresses to the device on the other end of the PVC.

Frame Relay mapping The information that correlates, or maps, a Frame Relay DLCI to the Layer 3 address of the DTE on the other end of the VC identified by the local DLCI.

Frame Relay An international standard data-link protocol that defines the capabilities to create a frame-switched (packet-switched) service, allowing DTE devices (typically routers) to send data to many other devices using a single physical connection to the Frame Relay service.

frame The result of encapsulating a Layer 3 packet with Layer 2 header and trailer information.

FTP A File Transfer Protocol that can be used to copy files (such as configuration files or the IOS) from a router or switch to an FTP server.

full mesh A network design term often used with multiaccess network such as Frame Relay, referring to the case in which a direct communications path exists between every pair of devices in the design.

full state In OSPF, a neighbor state that implies that the two routers have exchanged the complete (full) contents of their respective LSDBs.

full-duplex This duplex mode is used when only two devices share the collision domain, as a result, both devices can transmit simultaneously.

Gateway Load Balancing Protocol (GLBP) An FHRP that can load-balance traffic destined for a next-hop gateway across a collection of routers, known as a GLBP group. Specifically, when a client sends an Address Resolution Protocol (ARP) request, in an attempt to determine the MAC address corresponding to a known IP address, GLBP can respond with the MAC address of one member of the GLBP group. The next such request would receive a response containing the MAC address of a different member of the GLBP group.

gateway of last resort The notation in a Cisco IOS IP routing table that identifies the route used by that router as the default route.

ge (prefix list) Used to define that the mask of a network must be greater than or equal to the specified value for it to be a match to the prefix list.

generic routing encapsulation (GRE) A tunneling protocol that can be used to encapsulate many different protocol types, including IPv4, IPv6, IPsec, and others, to transport them across a network.

global unicast address A type of unicast IPv6 address that has been allocated from a range of public globally unique IP addresses as registered through ICANN, its member agencies, and other registries or ISPs.

going active EIGRP jargon meaning that EIGRP has placed a route into active status.

GRE tunnel A tunnel created using GRE. See *generic route encapsulation*.

GRE See *generic routing encapsulation*.

GUI (graphical user interface) A method of interacting with the Cisco IOS using a graphical interface such as a web page.

half-duplex This duplex mode only allows one device to transmit at a time, as multiple devices exist in the same collision domain.

Hello interval With OSPF and EIGRP, an interface timer that dictates how often the router should send Hello messages.

hello packet (EIGRP) An EIGRP message that identifies neighbors, exchanges parameters, and is sent periodically as a keepalive function. Hellos do not require an acknowledgment.

Hello packet (OSPF) A type of OSPF packet used to discover neighbors, check for parameter agreement, and monitor the health of another router.

hello time (BPDUs) The time interval between configuration BPDUs sent by the root bridge; defaults to 2 seconds.

Hold timer With EIGRP, the timer used to determine when a neighboring router has failed, based on a router not receiving any EIGRP messages, including Hellos, in this timer period.

holddown A state into which a route is placed so that routers neither advertise the route nor accept advertisements about it for a specific length of time (the holddown period). The holddown state is used to flush bad information about a route from all routers in the network. A route typically is placed in holddown when a link in that route fails.

hop count The metric used by RIP, RIPv2, and RIPv6 to identify how far a destination network is. It is based on the number of routers that a packet must pass through from the local router to reach the destination.

host port A switchport mapped to a private VLAN such that a connected device can communicate with only a promiscuous port or ports within the same community VLAN.

Hot Standby Routing Protocol (HSRP) HSRP uses virtual IP and MAC addresses. One router, known as the active forwarder, services requests destined for the virtual IP and MAC addresses. Another router, known as the standby router, can service such requests in the event the active router becomes unavailable.

HSRP active forwarder The router in an HSRP group that forwards traffic sent to the virtual gateway IP and MAC address.

HSRP standby router A router in an HSRP group that waits until the active router fails before taking over that role.

HTTP (Hypertext Transfer Protocol) A protocol that can be used to transfer files (such as configuration files or the IOS) from a router or switch to an HTTP server using hypertext.

IEEE 802.1X An IEEE standard that, when used with EAP, provides user authentication before their connected switchport allows the device to fully use the LAN.

IEEE 802.3 The standard upon which all generations of Ethernet (Ethernet, Fast Ethernet, Gigabit Ethernet, 10 Gigabit Ethernet) are based.

implicit deny all An invisible entry at the end of ACLs, prefix-lists, route-maps, and VACLs, that will automatically prevent all traffic or routes that do not match any entry before.

implicit permit Special invisible permanent statements in an IPv6 ACL that come before the implicit deny to allow ND traffic. They are **permit icmp any any nd-na** and **permit icmp any any nd-na**.

InARP See *Inverse ARP*.

infinity In the context of IP routing protocols, a finite metric value defined by the routing protocol that is used to represent an unusable route in a routing protocol update.

informs See *SNMP inform*.

infrastructure ACL An ACL typically configured on routers at the edge of an enterprise network, which help prevent malicious traffic from entering the network.

ingress port The port a frame is received on.

inside global address A NAT term referring to the IP address used for a host inside the trusted part of the network, but in packets as they traverse the global (untrusted) part of the network.

inside local address A NAT term referring to the IP address used for a host inside the trusted part of the network, but in packets as they traverse the local (trusted) part of the network.

interface ID Sixty-four bits at the end of an IPv6 global address, used to uniquely identify each host in a subnet.

interface table All the router interfaces that have been configured to participate in a routing process are listed in this table.

interface tracking (HSRP) A feature that allows an HSRP router to monitor the status of a local interface and decrement its priority if the interface is down.

Internal BGP (iBGP) Refers to how a router views a BGP peer relationship, in which the peer is in the same autonomous system.

Interior Gateway Protocol (IGP) A routing protocol designed to be used to exchange routing information inside a single autonomous system.

Internal BGP (iBGP) A characteristic of a BGP neighbor relationship, specifically when the two routers are internal to the same BGP ASN.

internal routers An OSPF router that has interfaces connected to only one area, making the router completely internal to that one area.

Internet Assigned Numbers Authority (IANA) An organization that directs the assignment of IPv4 and IPv6 addresses worldwide.

Internet service provider (ISP) A company that provides Internet connectivity.

Interrupt-driven task A network maintenance task that arises in response to a reported network issue.

Inter-Switch Link (ISL) The Cisco proprietary VLAN trunking protocol that predated 802.1Q by many years. ISL encapsulates the original Ethernet frame with 30-bytes of additional information and defines which VLAN the frame belongs to.

Inter-VLAN routing The function performed by a Layer 3 device that connects and forwards packets between multiple VLANs.

Inverse ARP Defined in RFC 1293, this protocol enables a Frame Relay-attached device to react to a received LMI “PVC up” message by announcing its Layer 3 addresses to the device on the other end of the PVC.

inverse neighbor discovery An IPv6 feature on non-broadcast multiaccess (NBMA) data links such as Frame Relay, providing the ability to learn a neighbor’s Layer 3 address when the underlying Layer 2 address is known. The IPv6 equivalent of Frame Relay Inverse ARP.

IP Background process When an interface changes its state, the IP Background process handles that state change.

IP prefix list See *prefix list*.

IP service level agreement (IP SLA) A feature within Cisco IOS that can be used to test how specific types of traffic are being handled end-to-end across a network.

IP SLA responder A network device that responds to probes and participates in IP SLA tests.

IP SLA source A network device using IP SLA which sends out a probe (synthetic traffic) to test the health of traffic.

IP Source Guard A switch security feature that prevents IP address spoofing by using the DHCP snooping database to verify that packets are sourced from the correct IP address.

IPsec tunnel A tunnel created using IPsec protocols.

IPsec Refers to the IP Security protocols, which is an architecture for providing encryption and authentication services, typically when creating VPN services through an IP network.

IPv4 Version 4 of the IP protocol, which is the generally deployed version worldwide (at publication) and uses 32-bit IP addresses.

IPv6 ACL An ACL that is used to identify IPv6 traffic based on multiple criteria such as source and destination IP address, source and destination port numbers, protocols, and QoS parameters and either allow or prevent it.

IPv6 Version 6 of the IP protocol, which uses 128-bit IP addresses.

isolated VLAN A type of secondary private VLAN; switchports associated with an isolated VLAN are effectively isolated from all other ports in the primary VLAN except the promiscuous port.

IST instance Internal spanning-tree instance; used by MST to represent an entire region as a single virtual bridge to a common spanning tree.

IT Infrastructure Library (ITIL) An ITIL defines a collection of best practice recommendations that work together to meet business goals.

jitter The variation in packet delivery delay times.

keepalive (BGP) A BGP message sent to maintain an active neighbor relationship and maintain the underlying TCP connection when a router has no other BGP messages to send.

key chain A collection of one or more keys (that is, passwords) used for authentication, where each key has an associated key ID and key string.

key ID (key chain) The numeric value that identifies the key used for authentication.

key string (key chain) The alphanumeric string of characters that is being used for authentication. This is not to be confused with the name of the key chain.

K value EIGRP allows for the use of bandwidth, load, delay, MTU, and link reliability; the K values refer to an integer constant that includes these five possible metric components. Only bandwidth and delay are used by default, to minimize recomputation of metrics for small changes in minor metric components.

LACP Link Aggregation Control Protocol; a standards-based method for negotiating EtherChannels automatically.

Layer 2 EtherChannel See *EtherChannel*.

Layer 3 EtherChannel See *EtherChannel*.

Layer 3 switch A Layer 3 switch can act as a Layer 2 switch (that is, making forwarding decisions based on MAC addresses), or it can make forwarding decisions based on Layer 3 information (for example, IP address information).

le (prefix list) Used to define that the mask of a network must be less than or equal to the specified value for it to be a match to the prefix-list.

learning One of four Spanning Tree Protocol (STP) states for a port. A port moves from the listening state to the learning state and remains in this state for 15 seconds by default. During this time, the port begins to add entries to its MAC address table.

level 4 encryption On Cisco IOS devices, passwords are hashed using SHA 256.

level 5 encryption On Cisco IOS devices, passwords are hashed using MD5.

level 7 encryption On Cisco IOS devices, passwords are encrypted using a weak Type 7 encryption.

line A configuration mode that can be used to manage a Cisco IOS device (for example, the console line or the vty lines).

link-local address A type of unicast IPv6 address that represents an interface on a single data link. Packets sent to a link local address cross only that particular link and are never forwarded to other subnets by a router. Used for communications that do not need to leave the local link, such as neighbor discovery.

link-state acknowledgment A type of OSPF packet used to acknowledge LSU and DBD packets.

link-state advertisement (LSA) The name of a class of OSPF data structures that hold topology information. LSAs are held in memory in the LSDB and communicated over the network in LSU messages.

link-state database (LSDB) In OSPF, the data structure in RAM of a router that holds the various LSAs, with the collective LSAs representing the entire topology of the network.

link-state identifier (LSID) A 32-bit number used to uniquely identify an OSPF LSA.

link-state request (LSR) An OSPF packet used to ask a neighboring router to send a particular LSA.

link-state update (LSU) The name of the OSPF packet that holds the detailed topology information, specifically LSAs.

link state A classification of the underlying algorithm used in some routing protocols.

listening One of the four STP states for a port. A port moves from the blocking state to the listening state and remains in this state for 15 seconds by default. During this time, the port sources bridge protocol data units (BPDU), which inform adjacent switches of the port intent to forward data.

LLDP Link Layer Discovery Protocol; a standards-based protocol used to advertise and discover directly connected devices.

LMI See *Local Management Interface*.

load A Cisco router interface statistic that measures the percentage link utilization, with the value represented as an integer between 0 to 255, and the percentage calculated as the listed number / 255. EIGRP can use load as input to the EIGRP metric calculation.

Loading An OSPF neighbor state that occurs after the completion of database description messages, but while the database exchange using link-state request and link-state update packets continues.

local computation An EIGRP router's reaction to an input event, leading to the use of a feasible successor or going active on a route.

Local Management Interface (LMI) A Frame Relay protocol used between a DTE (router) and DCE (Frame Relay switch). LMI acts as a keepalive mechanism. The absence of LMI messages means that the other device has failed. It also tells the DTE about the existence of each VC and DLCI, along with its status.

local preference See LOCAL_PREF.

local VLAN A single VLAN that is bounded by a small area of the network, situated locally with a group of member devices.

LOCAL_PREF A BGP path attribute that is communicated throughout a single AS to signify which route of multiple possible routes is the best route to be taken when leaving that AS. A larger value is considered to be better.

login local A Cisco IOS command used on lines to define that authentication is required, using the local username and password database, to access the line for management purposes.

login A Cisco IOS command used on lines to define that authentication is required, using a line password, to access the line for management purposes.

Loop Guard An STP feature that disables a switchport if expected BPDUs suddenly go missing to prevent a loop.

LSA flooding The process of successive neighboring routers exchanging LSAs such that all routers have an identical LSDB for each area to which they are attached.

LSA See *link-state advertisement*.

LSAck See Link-state acknowledgment.

LSDB See link-state database.

LSU See link-state update.

MAC address table A table used by switches to efficiently forward frames out the ports needed to reach devices based on their MAC address.

Management Information Base (MIB) A collection of information and data that a network device maintains about itself and its operation. MIB variables can be read or written through SNMP.

management plane The management plane of operation is used to manage a router or a switch. This management involves, for example, accessing and configuring a device.

manually configured tunnel A type of IPv6-over-IPv4 point-to-point tunnel in which the tunnel source and destination is preconfigured.

match (route-map) A clause that is used to define the traffic or routes that will match the route map sequence.

max age time The time interval that a switch stores a BPDU before discarding it or aging it out; the default is 20 seconds.

maximum transmission unit (MTU) An IP variable that defines the largest size allowed in an IP packet, including the IP header.

maximum paths The number of paths that can be used by a router to load balance traffic.

Media Access Control (MAC) address A MAC address is a 48-bit address assigned to various types of network hardware (for example, network interface cards in PCs). A Layer 2 switch can learn which MAC addresses reside on specific switchports and make forwarding decisions based on that information.

MED See *Multi Exit Discriminator*.

memory allocation failure A memory allocation failure (which produces a MALLOCFAIL error message) occurs when a process attempts to allocate a block of memory and fails to do so.

memory leak When a router starts a process, that process can allocate a block of memory. When the process completes, the process should return its allocated memory to the router's pool of memory. If not all the allocated memory is returned to the router's main memory pool, a memory leak occurs.

merge A situation that occurs when copying any configurations to the running configuration. The merge causes the configurations to be combined together, which could result in undesired configurations.

message digest 5 (MD5) Authentication With IP routing protocols, a method of applying a mathematical formula, with input including a private key, the message contents, and sometimes a shared text string, with the resulting digest being included with the message. The sender and the receiver perform the same math to allow authentication and to prove that no intermediate device changed the message contents.

method list (AAA authentication) A listing of methods, such as a RADIUS server, types of authentication, the local database, and the line passwords, that can be used to successfully authenticate. Typically listed in the sequence in which they will be performed.

metric With routing protocols, the measurement of favorability that determines which entry will be installed in a routing table if more than one router is advertising that exact network and mask with one routing protocol.

MIB See *Management Information Base*.

MLS See *multilayer switching*.

MST instance (MSTI) A single instance of STP running within an MST region; multiple VLANs can be mapped to the MST instance.

MST region A group of switches running compatible MST configurations.

MST Multiple Spanning Tree Protocol, used to map one or more VLANs to a single STP instance, reducing the total number of STP instances.

MTU Maximum transmission unit. The maximum packet size, in bytes, that a particular interface can handle.

Multi Exit Discriminator (MED) See MULTI_EXIT_DISC.

MULTI_EXIT_DISC (MED) A BGP path attribute that allows routers in one autonomous system to set a value and advertise it into a neighboring AS, impacting the decision process in that neighboring autonomous system. A smaller value is considered better. Also called the BGP metric.

multicast IP address range For IPv4, the multicast address range is from 224.0.0.0 through 239.255.255.255. For IPv6, multicast addresses have a prefix of ff00::/8.

multicast IP address structure For IPv4, the first 4 bits of the first octet must be 1110. The last 28 bits are unstructured. For IPv6, multicast addresses have a prefix of ff00::/8.

multicast MAC address A type of Ethernet MAC address meant to be used to send frames to a subset of the devices on a single broadcast domain. More specifically, as used with IPv4 multicast packets, a 48-bit address that is calculated from a Layer 3 multicast address by using 0x0100.5E as the multicast vendor code (OUI) for the first 24 bits, always binary 0 for the 25th bit, and copying the last 23 bits of the Layer 3 multicast address.

multilayer switch (Layer 3 switch) A process whereby a switch, when making a forwarding decision, uses not only Layer 2 logic but other OSI layer equivalents as well (such as Layer 3).

multipoint GRE A virtual private network (VPN) technology that allows multiple GRE tunnels to terminate on a single GRE tunnel interface.

multipoint redistribution When redistribution occurs at multiple points between two different routing protocols.

multipoint subinterface A configuration construct in Cisco routers, typically with Frame Relay, in which one logical subinterface can be used to forward traffic to more than one remote router.

multipoint tunnel A type of tunnel in which more than one destination may be reached over a single tunnel.

Multiprotocol BGP (MP-BGP) An updated version of BGPv4 that includes components supporting the routing of both IPv4 and IPv6 networks.

NA See Neighbor Advertisement.

named ACL An access list that identifies the various statements/entries in the ACL based on a name, rather than a number.

Named EIGRP An EIGRP configuration approach that allows you to configure all EIGRP commands under a single hierarchical configuration.

NAT overload See Port Address Translation (PAT).

NAT Virtual Interface (NVI) A feature that allows a router interface to act as either a NAT inside or a NAT outside interface.

NAT See Network Address Translation.

native VLAN The one VLAN on an 802.1Q trunk for which the endpoints do not add the 4-Byte 802.1Q tag when transmitting frames in that VLAN.

NBMA See non-broadcast multi-access (NBMA).

NCP See Network Control Protocol.

ND See Neighbor Discovery.

neighbor (EIGRP) With EIGRP, a router sharing the same primary subnet, with which Hellos are exchanged, parameters match, and with which routes can be exchanged.

neighbor (OSPF) Any other router, sharing a common data link, with which a router exchanges Hellos, and for which the parameters in the Hello pass the parameter-check process.

Neighbor Advertisement (NA) In IPv6, the Neighbor Discovery message used by an IPv6 node to send information about itself to its neighbors.

Neighbor Discovery (ND) The protocol used in IPv6 for many functions, including address autoconfiguration; duplicate address detection; router, neighbor, and prefix discovery; neighbor address resolution; and parameter discovery.

Neighbor Discovery Protocol (NDP) A longer name for IPv6 Neighbor Discovery. See *Neighbor Discovery*.

Neighbor Solicitation (NS) In IPv6, the Neighbor Discovery message used by an IPv6 node to request information about a neighbor or neighbors.

neighbor state A state variable kept by a router for each known neighbor or potential neighbor.

neighbor table For OSPF and EIGRP, a listing of routers that have reached neighbor status with the local router.

neighbor In routing protocols, another router with which a router decides to exchange routing information.

neighborship A shortened version of the phrase neighbor relationship.

net background process An interface has a certain number of buffers available to store packets. These buffers are sometimes referred to as an interface's queue. If an interface needs to store a packet in a buffer but all the interface's buffers are in use, the interface can pull from a main pool of buffers that its router maintains. The process that allows an interface to allocate one of these globally available buffers is the net background process.

NetFlow The NetFlow feature collects detailed information about traffic flows on routers and high-end switches. Collected information can optionally be sent to a NetFlow collector, which can produce reports about the traffic flows.

Network Address Translation (NAT) A mechanism for reducing the need for globally unique IPv4 addresses. NAT allows an organization with addresses that are not globally unique to connect to the Internet by translating those addresses into globally routable address space.

network command Used to enable the RIPv2, EIGRP for IPv4, and OSPFv2 routing process on an interface.

network layer reachability information A BGP term referring to an IP prefix and prefix length.

Network LSA An OSPFv2 Type 2 LSA. See *Type 2 LSA*.

Network Time Protocol (NTP) A protocol used to synchronize time among network devices.

network type (OSPF) A characteristic of OSPF interfaces that determines whether a DR election is attempted, whether neighbors must be statically configured, and the default hello and dead timer settings.

Next Hop field With a routing update, or routing table entry, the portion of a route that defines the next router to which a packet should be sent to reach the destination subnet. With routing protocols, the Next Hop field may define a router other than the router sending the routing update.

Next Hop Resolution Protocol (NHRP) A virtual private network (VPN) technology that allows a spoke, in a hub-and-spoke topology, to query the hub for the IP address of a physical interface on a different spoke that corresponds to the IP address of the far end of a tunnel.

NEXT_HOP A BGP path attribute that lists the next-hop IP address used to reach an NLRI.

Next-hop self A BGP configuration setting that tells the local router to change the NEXT_HOP path attribute to refer to its own BGP update source when advertising routes to BGP neighbors.

NLRI See *network layer reachability information*.

NMS See *SNMP manager*.

noAuthNoPriv An SNMP security model used with all versions of SNMP that only provides authentication for SNMPv1 and SNMPv2c using community strings and for SNMPv3 using a username.

nonbackbone area Any OSPF area that is not the backbone area.

nonbroadcast multiaccess (NBMA) A characterization of a type of Layer 2 network in which more than two devices connect to the network, but the network does not allow broadcast frames to be sent to all devices on the network.

nondesignated port Nondesignated ports in a spanning tree block traffic to create a loop-free topology.

Non-Stop Forwarding (NSF) A redundancy method that quickly rebuilds routing information after a redundant Catalyst switch supervisor takes over.

notification (BGP) A BGP message used to inform BGP neighbors of a protocol error.

not-so-stubby area A type of OSPF stub area, which acts like other stub areas in that ABRs inject default routes into the area, but unlike non-NSSA stub areas in that external routes can be injected into the area.

NS See *Neighbor Solicitation*.

NSSA See *not-so-stubby area*.

NTP See *Network Time Protocol*.

NTP client A device requesting NTP (Network Time Protocol) services from an NTP server so that its clock information can be synchronized.

NTP server A device providing NTP services.

object identifier (OID) A unique string of digits that identifies a variable or a tree of variables in a MIB.

object tracking An IOS feature in which IOS repeatedly checks the current state of some item so that other items can then act to a change in that state. For example, object tracking can track the state of IP SLA operations, with static routes and policy routes reacting to a change in the object-tracking feature.

offset list A Cisco IOS configuration tool for RIP and EIGRP for which the list matches routes in routing updates and adds a defined value to the sent or received metric for the routes. The value added to the metric is the offset.

one-way redistribution The process of route redistribution in which one routing protocol redistributes routes into a second routing protocol, but the reverse redistribution is not configured.

Open Shortest Path First (OSPF) A popular link-state IGP that uses a link-state database and the shortest path first (SPF) algorithm to calculate the best routes to reach each known subnet.

Open A BGP message type used when the underlying TCP connection completes, for the purpose of exchanging parameter information to determine whether the two routers are willing to become BGP neighbors.

ORIGIN A BGP path attribute that implies how the route was originally injected into some router's BGP table.

OSPF area A group of routers and links, identified by a 32-bit area number, whose detailed topology information OSPF shares among all routers in the group. Routers inside an area learn full detailed topology information about the area; this detailed information is not advertised outside the area.

OSPF network type A characteristic of OSPF interfaces that determines whether a DR election is attempted, whether neighbors must be statically configured, and the default hello and dead timer settings.

OSPF Version 3 An interior routing protocol created for IPv6 but based on OSPF Version 2, which was designed for IPv4.

OSPF See *Open Shortest Path First*.

OSPF area border router (ABR) See *Area Border Router*.

OSPF Autonomous System Boundary Router (ASBR) See *ASBR (Autonomous System Border Router)*.

OSPF interface table See *interface table*.

OSPF neighbor table See *neighbor table*.

OSPF link-state database See *link-state database*.

OSPFv3 An enhancement to OSPFv2 that supports the routing of IPv6 networks.

OSPFv3 address family A newer configuration approach for OSPFv3 that supports the routing of both IPv4 and IPv6 networks with a single OSPFv3 process (as opposed to having one OSPFv2 process for the routing of IPv4 networks and one OSPFv3 process for the routing of IPv6 networks).

outside global address A NAT term describing an IP address representing a host that resides outside the enterprise network, with the address being used in packets outside the enterprise network.

outside local address A NAT term describing an IP address representing a host that resides outside the enterprise network, with the address being used in packets inside the enterprise network.

overlapping subnets An (incorrect) IP subnet design condition in which one subnet's range of addresses includes addresses in the range of another subnet.

overloading Another term for Port Address Translation. See *PAT*.

packet forwarding The process a router uses to take a packet that arrives on an interface (ingress) and forward it out the appropriate interface.

packet forwarding The process of forwarding packets through a router. Also called IP routing.

PAgP Port Aggregation Protocol; a Cisco-developed method for negotiating EtherChannels automatically.

partial mesh A network topology in which more than two devices could physically communicate, but by choice, only a subset of the pairs of devices connected to the network are allowed to communicate directly.

passive (EIGRP) A state for a route in an EIGRP topology table that indicates that the router believes that the route is stable and that it is not currently looking for any new routes to that subnet.

passive interface A routing protocol setting on an interface for which the router does not send Updates on the interface (RIP) or the router does not attempt to dynamically discover neighbors (EIGRP and OSPF), which indirectly prevents the EIGRP or OSPF router from sending updates on the interface.

PAT See *Port Address Translation*.

path attribute Generally describes characteristics about BGP paths advertised in BGP updates.

path control A general term, with several shades of meanings, that refers to any function that impacts how routers forward packets. These functions include routing protocols and any other feature that impacts the IP routing table, plus any feature that impacts the packet-forwarding process.

path vector A category of routing protocol that includes information about the exact path packets take to reach a specific destination network. BGP is a common example of a path-vector routing protocol.

PBR See *policy-based routing*.

peer group See *BGP peer group*.

periodic update With routing protocols, the concept that the routing protocol advertises routes in a routing update on a regular periodic basis. This is typical of distance-vector routing protocols.

permanent virtual circuit (PVC) A preconfigured communications path between two Frame Relay DTEs, identified by a local DLCI on each Frame Relay access link, that provides the functional equivalent of a leased circuit but without a physical leased line for each VC.

permit An action taken with an ACL that implies that the matching packets will be allowed.

ping A tool that can be used to test IPv4/IPv6 connectivity between two devices.

point-to-point tunnel A logical path between two devices created by encapsulating packets of one protocol (the passenger protocol) inside packets of another protocol (the transport protocol) specifically in cases where only two routers exist in the tunnel.

poison reverse With RIP, the advertisement of a poisoned route out an interface when that route was formerly not advertised out that interface due to split horizon rules.

poisoned route A route in a routing protocol's advertisement that lists a subnet with a special metric value, called an infinite metric, that designates the route as a failed route.

policy-based routing Cisco IOS router feature by which a route map determines how to forward a packet, typically based on information in the packet other than the destination IP address.

port (multiple definitions) 1) In TCP and UDP, a number used to uniquely identify the application process that either sent (source port) or should receive (destination port) data. 2) In LAN switching, another term for switch interface.

port 22 Well-known port number used by SSH.

port 23 Well-known port number used by Telnet.

Port Address Translation (PAT) A NAT term describing the process of multiplexing TCP and UDP flows, based on port numbers, to a small number of public IP addresses. Also called NAT overloading.

port security A feature that is used to control the specific MAC addresses learned on an interface or the number of MAC addresses learned on an interface.

PortFast An STP feature used primarily on an access port that bypasses the Listening and Learning states so that the host can gain quick access to the network.

PPDIOO Prepare, plan, design, implement, operate, optimize. The six phases of the Cisco Lifecycle Services approach.

preempt (HSRP/VRRP/GLBP) A feature that allows a router participating in a first-hop redundancy protocol group to take over as the active forwarder or the AVG if it has a higher priority.

prefix list A Cisco IOS configuration tool that you can use to match routing updates based on a base network address, a prefix, and a range of possible masks used inside the values defined by the base network address and prefix.

primary VLAN The main VLAN associated with a PVLAN (private VLAN) that will be divided up into secondary VLANs to control the follow of traffic between the ports in the VLAN.

priority (GLBP) A numeric value from 1 to 255 that is used to control who the AVG will be. Higher is better.

priority (HSRP/VRRP) A numeric value from 1 to 255 that is used to control who the active forwarder will be. Higher is better.

priority (OSPF) An administrative setting included in hellos that is the first criteria for electing a DR. The highest priority wins, with values from 1 to 255, with priority 0 meaning a router cannot become DR or BDR.

private addresses RFC 1918-defined IPv4 network numbers that are not assigned as public IP address ranges and are not routable on the Internet. Intended for use inside enterprise networks.

private autonomous system A BGP ASN whose value is between 64,512 and 65,535. These values are not assigned for use on the Internet and can be used for private purposes, typically either within confederations or by ISPs to hide the ASN used by some customers.

private ASN An autonomous system number (ASN) that falls inside the private autonomous system range.

private IP address See *private addresses*.

private VLAN A special-purpose VLAN, designated as either primary or secondary, that can restrict or isolate traffic flows between devices in the same VLAN.

process switching A method of switching packets from an ingress interface to an egress interface on a router or multilayer switch that requires the CPU to evaluate every packet. This is the least-efficient switching method.

promiscuous port A switchport mapped to a private VLAN that is used by ports in community or isolated VLANs to access resources outside the private VLAN.

protect violation mode A port security violation mode that will prevent access to the devices that caused the violation based on the number of MAC addresses being exceeded.

protected ports A global switch security feature that will prevent edge ports that are configured as protected ports from communicating with each other unless they are forwarded through a Layer 3 device.

protocol type A field in the IP header that identifies the type of header that follows the IP header, typically a Layer 4 header, such as TCP or UDP. ACLs can examine the protocol type to match packets with a particular value in this header field.

proxy ARP A router feature used when a router sees an ARP request searching for an IP host's MAC, when the router believes the IP host could not be on that LAN because the host is in another subnet. If the router has a route to reach the subnet where the ARP-determined host resides, the router replies to the ARP request with the router's MAC address.

public address space (IPv4) The nonreserved portions of the IPv4 unicast address space.

public ASN An ASN that fits below the private ASN range, specifically from 1 through 54,511.

public IP address See *public address space*.

PVC See *permanent virtual circuit*.

PVST+ Per-VLAN Spanning Tree; a Cisco proprietary version of STP where one instance of STP runs for each VLAN present in a Layer 2 switch.

query (EIGRP) An EIGRP message that asks neighboring routers to verify their route to a particular subnet. Query messages require an acknowledgment.

query scope (EIGRP) The characterization of how far EIGRP query messages flow away from the router that first notices a failed route and goes active for a particular subnet.

RA See *router advertisement*.

RADIUS A standards-based protocol used to communicate with AAA servers.

RD See *reported distance*.

redistribution The process on a router of taking the routes from the IP routing table, as learned by one routing protocol, and injecting routes for those same subnets into another routing protocol.

reference bandwidth In OSPF, the numerator in the calculation of interface cost. The formula is reference bandwidth / interface bandwidth.

Regional Internet Registry (RIR) The generic term for one of five current organizations responsible for assigning the public, globally unique IPv4 and IPv6 address space.

registry prefix In IPv6, the prefix that describes a block of public, globally unique IPv6 addresses assigned to a Regional Internet Registry by IANA.

regular expression A list of interspersed alphanumeric literals and metacharacters used to apply complex matching logic to alphanumeric strings. Often used for matching AS_PATHs in Cisco routers.

reliability A Cisco router interface statistic that measures the percentage of packet loss, with the value represented as an integer between 0 to 255, and the percentage calculated as the listed number / 255. EIGRP can use reliability as input to the EIGRP metric calculation.

Reliable Transport Protocol A protocol used for reliable multicast and unicast transmissions. Used by EIGRP.

Reply (EIGRP) An EIGRP message that is used by neighbors to reply to a query. Reply messages require an acknowledgment.

REPLY message The DHCPv6 server finalizes the DHCPv6 addressing process with this message.

reported distance From one EIGRP router's perspective, the metric for a destination network as calculated on a neighboring router and reported in a routing update to the first router.

REQUEST message A DHCPv6 client sends this message to the DHCPv6 server confirming the addresses provided and any other parameters.

restrict violation mode A port security violation mode that will prevent access to the devices that caused the violation based on exceeding the number of MAC addresses allowed and send log messages about the violation.

RIB failure An event that occurs when the Routing Table Manager (RTM) attempts to add a route to the IP routing table, but a problem exists with the route that prevents RTM from adding the route.

RID See *router ID*.

RIP (Routing Information Protocol) An Interior Gateway Protocol (IGP) that uses distance vector logic and router hop count as the metric. RIP Version 1 (RIPv1) has become unpopular.

RIP next generation (RIPng) An IPv6 Interior Routing Protocol based on RIP (for IPv4).

RIP Version 2 (RIPv2) Provides more features than RIP, including support for VLSM.

rollover cable (console cable) The cable needed to successfully manage a Cisco IOS device from the console port.

root bridge The single STP device within a broadcast domain/VLAN that is elected as a common frame of reference for working out a loop-free Layer 2 topology.

Root Guard An STP feature that controls where candidate root bridges can be found on a switch.

root path cost The cumulative cost of all the links leading to the root bridge.

root port Each switch selects one port that has the lowest root path cost leading toward the root bridge.

route map A configuration tool in Cisco IOS that enables basic programming logic to be applied to a set of items. Often used for decisions about what routes to redistribute and for setting particular characteristics of those routes (for instance, metric values).

route poisoning The process of sending an infinite-metric route in routing updates when that route fails.

route redistribution The process of taking routes known through one routing protocol and advertising those routes with another routing protocol.

route summarization A consolidation of advertised addresses that causes a single summary.

route tag A field within a route entry in a routing update used to associate a generic number with the route. It is used when passing routes between routing protocols, allowing an intermediate routing protocol to pass information about a route that is not natively defined to that intermediate routing protocol. Often used for identifying certain routes for filtering by a downstream routing process.

route to be advertised Route was formerly not advertised out that interface due to split-horizon rules.

routed port A Layer 3 port on a multilayer switch that behaves similar to an interface on a router and is not associated with a particular VLAN.

routed protocol A Layer 3 protocol that defines a packet that can be routed, such as IPv4 and IPv6.

router advertisement (RA) In IPv6, a router advertisement message used by an IPv6 router to send information about itself to nodes and other routers connected to that router.

router ID (RID) In OSPF, a 32-bit number, written in dotted decimal, that uniquely identifies each router.

Router LSA Another name for an OSPF Type 1 LSA. See *Type 1 LSA*.

router solicitation (RS) An IPv6 message, part of the Neighbor Discovery Protocol (NDP), used by a host to request that the routers on the same data link announce their presence, IPv6 addresses, and all prefix/length combinations using a router advertisement (RA) message.

router-on-a-trunk/router-on-a-stick A router with subinterfaces that is used to route traffic between multiple VLANs.

Routing Information Base (RIB) A term referring to the IP routing table.

routing loop When traffic is routed back in the direction that it came from or in a circular pattern through the network never reaching the intended destination.

routing protocol A set of messages and processes with which routers can exchange information about routes to reach subnets in a particular network. Examples of routing protocols include Enhanced Interior Gateway Routing Protocol (EIGRP), Open Shortest Path First (OSPF), and Routing Information Protocol (RIP), routing protocols to analyze the LSDB and find the least-cost routes from that router to each subnet.

routing table The table a router uses to determine the most appropriate way to forward a packet.

RPVST+ Also known as Rapid PVST+, where RSTP is used on a per-VLAN basis; in effect, RSTP replaces traditional 802.1D STP in the PVST+ operation.

RSPAN Also known as Remote Switched Port Analyzer, where a SPAN session is split across two independent switches and mirrored data is transported over a special purpose VLAN between them.

RSTP The Rapid Spanning Tree Protocol, based on the IEEE 802.1w standard.

running configuration Contains the current configuration that is running in RAM on the router or switch.

running config See *running configuration*.

SDM Switching Database Manager; a Cisco IOS Software function that configures or tunes memory table space on a LAN switch platform.

secondary VLAN A VLAN used with PVLANS that can pass traffic to and from its associated primary VLAN, but not with any other secondary VLAN associated with the same primary VLAN.

Secure Hash Algorithm (SHA) An authentication algorithm, considered to be more secure than MD5, which can provide neighbor authentication for Named EIGRP and OSPFv3.

Secure Sockets Layer (SSL) A security protocol integrated into commonly used web browsers that provides encryption and authentication services between the browser and a website.

seed metric When redistributing routes, the metric set for routes injected into another routing protocol.

segment (multiple definitions) 1) In TCP, a term used to describe a TCP header and its encapsulated data (also called an L4PDU). 2) Also in TCP, the set of bytes formed when TCP breaks a large chunk of data given to it by the application layer into smaller pieces that fit into TCP segments. 3) In Ethernet, either a single Ethernet cable or a single collision domain (no matter how many cables are used).

sequence number (OSPF) In OSPF, a number assigned to each LSA, ranging from 0x80000001 and wrapping back around to 0x7FFFFFFF, which determines which LSA is most recent.

set (route-map) A clause that defines the action that will be taken to the traffic or routes that match the match clause in the route map.

shoot from the hip The shoot from the hip troubleshooting method occurs when a troubleshooter bypasses examined information and eliminates potential causes, based on the troubleshooter's experience and insight.

shortest path first (SPF) The name of the algorithm OSPF uses to analyze the LSDB. The analysis determines the best (lowest cost) route for each prefix/length. Also known as Dijkstra's SPF algorithm.

shutdown violation mode A port security violation mode that will prevent access to all devices connected to the port by placing the port in the err-disabled state.

SIA query An EIGRP hello specially used halfway through a router's active timer for a route in which a router queries the downstream neighbor to discover if that neighbor is still working.

Simple Network Management Protocol (SNMP) A network management protocol that can allow a network management system (NMS) to query a managed device (that is, an SNMP client) for information found in the device's Management Information Base (MIB), and can also allow a managed device to proactively send notifications (called traps) to an NMS in response to specific events.

single-point redistribution When redistribution occurs at a single point between two different routing protocols.

site prefix In IPv6, the prefix that describes a public globally unique IPv6 address block that has been assigned to an end-user organization (for example, an enterprise or government agency). An ISP or Internet registry typically makes the assignment.

site-to-site VPN A site-to-site VPN typically terminates in a router at a headquarters and a router at the remote site. Such an arrangement does not require clients at the remote site to have VPN client software installed.

SLA (service level agreement) operation A configuration construct used by the IP SLA feature inside router Cisco IOS that defines a type of packet to be sent, plus a set of measurements to be made about the packet. (Did a reply occur? What delay occurred, jitter, and so on?)

smoothed round-trip time With EIGRP, a purposefully slowly changing measurement of round-trip time between neighbors from which the EIGRP RTO is calculated.

sniffer A device on the network that is designed to capture packets that are sent to it or passed through it so that they can be analyzed. Wireshark is an example of a packet sniffer.

SNMP See *Simple Network Management Protocol (SNMP)*.

SNMP agent A process that runs on the network device being monitored and uses SNMP to provide data to an SNMP manager.

SNMP inform A message that a network device sends to alert an SNMP manager about an event or a failure. The SNMP manager must acknowledge receipt of the inform by echoing the message back to the SNMP agent in the device.

SNMP manager A network management system that uses SNMP to poll network devices for operational and configuration data.

SNMP trap A message that a network device sends to alert an SNMP manager about an event or a failure. The SNMP manager does not need to acknowledge a trap that it receives.

SNMP view A grouping of objects within the MIB that defines which objects a user will be able to access.

SNMPv2c A version of SNMP that uses community strings.

SNMPv3 A version of SNMP that can use hashing algorithms and encryption algorithms to enhance SNMP security.

socket A three-tuple consisting of an IP address, port number, and transport layer protocol. TCP connections exist between a pair of sockets.

soft reconfiguration A BGP process by which a router reapplies routing policy configuration (route maps, filters, and the like) based on stored copies of sent and received BGP updates.

SOLICIT message A DHCPv6 client sends this message to locate DHCPv6 servers using the multicast address FF02::1:2 which is the all DHCPv6 servers multicast address.

solicited node multicast address In IPv6, an address used in the Neighbor Discovery (ND) process. The format for these addresses is FF02::1:FF00:0000/104, and each IPv6 host must join the corresponding group for each of its unicast and anycast addresses.

source MAC address (source MAC) The MAC address of the sender of a frame.

SPAN Also known as Switched Port Analyzer, where a switch mirrors traffic from a source interface or VLAN onto a different interface for monitoring or analysis purposes.

Spanning Tree Protocol (STP) A protocol communicated between Layer 2 switches that attempts to detect a loop in the topology before it forms, thus preventing a bridging loop from occurring.

SPF calculation The process of running the SPF algorithm against the OSPF LSDB, with the result being the determination of the current best route(s) to each subnet.

split-horizon (iBGP) A loop prevention mechanism that prevents iBGP routers from advertising BGP learned routes to other iBGP neighbors.

split horizon (RIP and EIGRP) A loop prevention mechanism that prevents routers from advertising routes out the interfaces they were originally learned on.

SSH (Secure Shell) A secure protocol that can be used to remotely manage a Cisco IOS device.

SSL See *Secure Sockets Layer*.

standard ACL A list of IOS global configuration commands that can match only a packet's source IP address for the purpose of deciding which packets to discard and which to allow.

standby router See *HSRP standby router*.

stateful DHCPv6 A term used in IPv6 to contrast with stateless DHCP. Stateful DHCP keeps track of which clients have been assigned which IPv6 addresses (state information).

stateless address autoconfiguration (SLAAC) A method used by an IPv6 host to determine its own IP address, without DHCPv6, by using Neighbor Discovery Protocol (NDP) and the modified EUI-64 address format. See also stateful autoconfiguration.

stateless DHCPv6 A term used in IPv6 to contrast with stateful DHCP. Stateless DHCP servers don't lease IPv6 addresses to clients. Instead, they supply other useful information, such as DNS server IP addresses, but with no need to track information about the clients (state information).

static default route A default route configured in IOS using the `ip route` command.

Static NAT (SNAT) A version of Network Address Translation (NAT) where there is a static assignment of an inside global address to an inside local address.

static route A route manually configured by an administrator using the **ip route** or **ipv6 route** command.

sticky secure MAC address MAC addresses dynamically learned by the port security feature and entered into the running configuration like a static entry would be.

stratum A number that indicates in which layer in the NTP hierarchy a time source is located; stratum 1 represents the most authoritative and accurate time source.

structured maintenance task A structured maintenance task is a network maintenance task that is performed as part of a predefined plan.

stub See *EIGRP stub router*.

stub area An OSPF area into which external (Type 5) LSAs are not introduced by its ABRs; instead, the ABRs originate and inject default routes into the area.

stub network (OSPF) A network/subnet to which only one OSPF router is connected.

stub router (EIGRP) A router that should not be used to forward packets between other routers. Other routers will not send Query messages to a stub router.

stub router (OSPF) A router that should either permanently or temporarily not be used as a transit router. Can wait a certain time after OSPF process starts, or after BGP notifies OSPF that BGP has converged, before ceasing to be a stub router.

stubby area The same as stub area. See *stub area*.

stuck in active The condition in which a route has been in an EIGRP active state for longer than the router's active timer.

subinterface One of the virtual interfaces on a single physical interface.

subnet broadcast address A single address in each subnet for which packets sent to this address will be broadcast to all hosts in the subnet. It is the highest numeric value in the range of IP addresses implied by a subnet number and prefix/mask.

subnet prefix In IPv6, a term for the prefix that is assigned to each data link, acting like a subnet in IPv4.

subnet A subdivision of a Class A, B, or C network, as configured by a network administrator. Subnets allow a single Class A, B, or C network to be used and still allow for a large number of groups of IP addresses, as is required for efficient IP routing.

subnets keyword (OSPF) Used when redistributing into OSPF so that classful and classless networks are redistributed.

successor route With EIGRP, the route to each destination for which the metric is the lowest of all known routes to that network.

successor In EIGRP, the route to reach a subnet that has the best metric and should be placed in the IP routing table.

Summary LSA In OSPF, a Type 3 LSA. See *Type 3 LSA*.

summary route A route that is created to represent one or more smaller component routes, typically to reduce the size of routing and topology tables.

superior BPDU A received BPDU that contains a better bridge ID than the current root bridge.

SVI Switched virtual interface; a logical interface used to assign a Layer 3 address to an entire VLAN.

syslog severity level An indicator of how important or severe a logged event is.

syslog System message logs that are generated by a switch and can be collected locally or sent to and collected on a remote server.

TACACS+ A Cisco proprietary protocol used to communicate with AAA servers.

TCAM Ternary content-addressable memory; a switching table found in Catalyst switches that is used to evaluate packet forwarding decisions based on policies or access lists. TCAM evaluation is performed simultaneously with the Layer 2 or Layer 3 forwarding decisions.

TCN Topology Change Notification; a message sent out the root port of a switch when it detects a port moving into the forwarding state or back into the blocking state. The TCN is sent toward the root bridge to inform it of the topology change, where it is reflected and propagated to every other switch in the Layer 2 network.

TCP Timer process The TCP Timer process runs for each of the TCP connections for a router. Therefore, a router with many simultaneous TCP connections could have a high CPU utilization due to the resources being consumed by the TCP Timer.

Telnet An unsecure protocol that sends data in clear-text which can be used to remotely manage a Cisco IOS device.

TFTP (Trivial File Transfer Protocol) A protocol that can be used to copy files (such as configuration files or the IOS) from a router or switch to a TFTP server.

time-based ACL An access control list that can permit or deny defined traffic based on time of day and day of week.

time-to-live (TTL) (BGP) Identifies the lifetime of a BGP message in router hops. For eBGP peers, it is set to 1 by default, and for iBGP peers it is set to 255 by default.

time-to-live (TTL) A field in the IP header that is decremented at each pass through a Layer 3 forwarding device.

top-down method A method of troubleshooting where troubleshooting starts at the top (that is, Layer 7) of the OSI model and works its way down.

topology database The structured data that describes the network topology to a routing protocol. Link-state and advanced distance-vector protocols use topology tables, from which they build the entries in the routing table.

totally NSSA area A type of OSPF NSSA area for which neither External (Type 5) LSAs are introduced, nor Type 3 Summary LSAs; instead, the ABRs originate and inject default routes into the area. External routes can be injected into a totally NSSA area.

totally stubby area A type of OSPF stub area for which neither External (Type 5) LSAs are introduced, nor Type 3 Summary LSAs; instead, the ABRs originate and inject default routes into the area. External routes cannot be injected into a totally stubby area.

traceroute A tool that can be used on Cisco IOS devices to identify the path a packet is taking through the network.

tracking object A concept in IOS that analyzes different conditions on a router that results in the object's state either being up or down. IOS can then use different features, or not use different features, based on the current state of the tracking object. (In this book, tracking objects watch IP SLA operations and influence static routes and policy-based routing.)

transit area The area over which an OSPF virtual link's messages flow.

transit autonomous system With BGP, an autonomous system that receives packets from one neighboring autonomous system and forwards the packet to yet another autonomous system. An enterprise typically does not want to be a transit AS.

traps See *SNMP trap*.

triggered updates A routing protocol feature for which the routing protocol sends routing updates immediately upon hearing about a changed route, even though it may normally only send updates on a regular update interval.

trunk (VLAN) A physical link that can carry traffic for multiple VLANs.

TTL See *time-to-live*.

tunnel interface In Cisco IOS, a software (virtual) interface used as a configuration construct to configure a tunnel.

tunnel A method of taking one packet and encapsulating another packet so that the original encapsulated packet can be delivered across another network—in some cases across networks through which the original packet could not have been forwarded. The tunnel might simply provide for packet delivery, and it might add other services such as encryption and authentication.

tunneling The process of using a tunnel. See *tunnel*.

two-way redistribution With route redistribution, the process of redistributing routes from one routing protocol into a second routing protocol and vice versa.

two-way state In OSPF, a neighbor state that implies that the router has exchanged hellos with the neighbor, and all required parameters match.

Type 1 LSA An OSPF LSA type that describes a router. It lists the router's OSPF ID, its interfaces, their states, and the link-state IDs of neighboring LSAs. Also called a router LSA.

Type 2 LSA An OSPF LSA type that describes a multiaccess network on which a DR has been elected and for which at least one other router connects. The LSA represents the subnet. Also called a network LSA.

Type 3 LSA An OSPF LSA type that describes a subnet in another area. Also called a Summary LSA.

Type 3 LSA Filtering The process of causing an ABR to not create and flood a Type 3 LSA into another area.

Type 4 Summary ASBR LSA An LSA type used to describe an ASBR and the cost to reach that ASBR for the purpose of allowing routers to determine the OSPF cost to reach an external subnet advertised as a Type 5 or Type 7 LSA. Also called an ASBR Summary LSA.

Type 5 LSA An LSA that represents a subnet that OSPF learned from another (external) routing source, typically through route redistribution by an ASBR. Also known as an External LSA.

Type 7 AS External LSA An LSA type that describes an external subnet as injected into an NSSA area.

UDLD Unidirectional Link Detection; a feature that enables a switch to confirm that a link is operating bidirectionally. If not, the port can be disabled automatically.

unequal-cost load balancing A feature of EIGRP in which EIGRP includes multiple routes for the same prefix in the IP routing table but with IOS forwarding packets proportionally based on the calculated integer metric for each route.

unicast MAC address Ethernet MAC address that represents a single NIC or interface.

Unicast Reverse Path Forwarding (uRPF) A Cisco IOS feature that allows an interface to check the source IP address of an arriving packet and permit or deny that packet based on whether or not that IP address is reachable, based on the router's FIB (and optionally based on whether the egress interface to get back to that source IP address is the same interface on which it is arriving).

unique local address A type of IPv6 unicast address meant as a replacement for IPv4 private addresses.

unknown unicast flooding The action taken by a switch when the destination MAC address cannot be found in the MAC address table; the frame is flooded or replicated out all switchports except the receiving port.

update (EIGRP) An EIGRP message that informs neighbors about routing information. Update messages require an acknowledgment.

update source (BGP) In BGP, a reference to the IP address used as the source address of packets that hold BGP messages. The Update source can differ from neighbor to neighbor and is important in that a BGP router may set a route's NEXT_HOP to its update source IP address.

UplinkFast An STP feature that enables access layer switches to unblock a redundant uplink when the primary root port fails.

VACL VLAN access control list; a filter that can control traffic passing within a VLAN.

variance An integer setting for EIGRP. Any FS route whose metric is less than this variance multiplier times the successor's metric is added to the routing table, within the restrictions of the `maximum-paths` command.

virtual circuit A logical concept that represents the path over which frames travel between DTEs. VCs are particularly useful when comparing Frame Relay to leased physical circuits.

virtual link With OSPF, the encapsulation of OSPF messages inside IP to a router with which no common subnet is shared for the purpose of either mending partitioned areas or providing a connection from some remote area to the backbone area.

virtual MAC address The MAC address associated with the virtual router in an HSRP/VRRP group.

virtual private network (VPN) A set of security protocols that, when implemented by two devices on either side of an unsecure network such as the Internet, can enable the devices to send data securely. VPNs provide privacy, device authentication, antireplay services, and data integrity services.

Virtual Router Redundancy Protocol (VRRP) VRRP, similar to Hot Standby Routing Protocol (HSRP), allows a collection of routers to service traffic destined for a single IP address. Unlike HSRP, the IP address serviced by a VRRP group does not have to be a virtual IP address. The IP address can be the address of a physical interface on the virtual router master, which is the router responsible for forwarding traffic destined for the IP address of the VRRP group.

virtual router The IP address of a virtual router acting as the default gateway in an HSRP/VRRP group.

Virtual Routing and Forwarding (VRF) A technology that allows a single physical router to run multiple virtual router instances.

VLAN Virtual LAN; a logical network existing on one or more Layer 2 switches, forming a single broadcast domain.

VLAN hopping A malicious host sends specially crafted frames that contain extra, spoofed 802.1Q trunking tags into an access port, while the packet payloads appear on a totally different VLAN.

VLAN number A unique index number given to a VLAN on a switch, differentiating it from other VLANs on the switch.

VLSM Variable-length subnet mask(ing). The ability to specify a different subnet mask for the same Class A, B, or C network number on different subnets. VLSM can help optimize available address space.

voice VLAN The VLAN used between a Cisco IP phone and a Catalyst switch to carry voice traffic.

VPN client Software that resides on a PC, often a laptop, so that the host can implement the protocols required to be an endpoint of a VPN.

VPN See *virtual private network*.

VRF-Lite A traditional approach to configuring Virtual Routing and Forwarding (VRF) on Cisco routers.

VRRP virtual master router The router in a VRRP group that forwards traffic sent to the virtual gateway IP and MAC address.

VRRP virtual router backup A router in a VRRP group that waits until the master router fails before taking over that role.

VTP configuration revision number An index that indicates the current version of VLAN information used in the VTP domain; a higher number is more preferable.

VTP domain name The name used to identify each unique VTP domain.

VTP domain A logical grouping of switches that share a common set of VLAN requirements.

VTP pruning A VTP feature that reduces unnecessary flooded traffic by pruning or removing VLANs from a trunk link, only when there are no active hosts associated with the VLANs.

VTP VLAN Trunking Protocol; used to communicate VLAN configuration information among a group of switches.

weight A local Cisco proprietary BGP attribute that is not advertised to any peers. A larger value is considered to be better.

weighting 1) A numeric value from 0 to 255 that is used by GLBP AVFs with object tracking to determine whether they are healthy enough to forward traffic for the GLBP virtual MAC address it has been assigned. If not healthy enough, another AVF can take over the forwarding of traffic for that virtual MAC address. 2) A numeric value from 0 to 255 assigned to AVFs that is used by the AVG to determine how the virtual MAC address will be distributed to the clients when they request the MAC address via an ARP request.

wiki A wiki (which is the Hawaiian word for fast) can act as a web-based collaborative documentation platform.