Official Cert Guide

Advance your IT career with hands-on learning

Flash Cards

Review Exercises

Study Planner

CCNP Enterprise Advanced Routing

ENARSI 300-410

RAYMOND LACOSTE
BRAD EDGEWORTH, CCIE® NO. 31574

ciscopress.com

CCNP Enterprise Advanced Routing ENARSI 300-410 Official Cert Guide Special Offers

Enhance Your Exam Preparation

Save 70% on Complete Video Course

The *CCNP Enterprise Advanced Routing ENARSI 300-410 Complete Video Course*, available for both streaming and download, provides you with hours of expert-level instruction mapped directly to exam objectives. Put your knowledge to the test with full practice exams powered by the Pearson Test Prep practice test software, module quizzes, and more.

Save 80% on Premium Edition eBook and Practice Test

The *CCNP Enterprise Advanced Routing ENARSI 300-410 Premium Edition eBook and Practice Test* provides three eBook files (PDF, EPUB, and MOBI/Kindle) to read on your preferred device and an enhanced edition of the Pearson Test Prep practice test software. You also receive two additional practice exams with links for every question mapped to the PDF eBook.

See the card insert in the back of the book for your Pearson Test Prep activation code and special offers.

CCNP Enterprise Advanced Routing

ENARSI 300-410

Official Cert Guide

RAYMOND LACOSTE

BRAD EDGEWORTH, CCIE No. 31574

Cisco Press

221 River Street

Hoboken, NJ 07030 USA

CCNP Enterprise Advanced Routing ENARSI 300-410 Official Cert Guide

Raymond Lacoste, Brad Edgeworth

Published by:
Cisco Press
221 River Street
Hoboken, NJ 07030 USA

ScoutAutomatedPrintCode

Library of Congress Control Number: 2019919828

ISBN-13: 978-1-58714-525-4

ISBN-10: 1-58714-525-1

Warning and Disclaimer

This book is designed to provide information about the Implementing Cisco Enterprise Advanced Routing and Services (ENARSI) exam. Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied.

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

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

Trademark Acknowledgments

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

Special Sales

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

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

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

Feedback Information

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

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

We greatly appreciate your assistance.

Editor-in-Chief: Mark Taub

Alliances Manager, Cisco Press: Arezou Gol

Director, Product Manager: Brett Bartow

Managing Editor: Sandra Schroeder

Development Editor: Marianne Bartow

Project Editor: Mandie Frank

Copy Editor: Kitty Wilson

Technical Editors: Hector Mendoza, Jr, Russ Long

Editorial Assistant: Cindy Teeters

Designer: Chuti Prasertsith

Composition: codeMantra

Indexer: Cheryl Ann Lenser

Proofreader: Abigail Bass

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

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

Europe Headquarters
Cisco Systems International BV Amsterdam,
The Netherlands

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

Credits

Figure 7-1 Screenshot of wireshark ©2019 wireshark

Contents at a Glance

Contents

About the Authors

Raymond Lacoste has dedicated his career to developing the skills of those interested in IT. In 2001, he began to mentor hundreds of IT professionals pursuing their Cisco certification dreams. This role led to teaching Cisco courses full time. Raymond is currently master instructor for Cisco Enterprise Routing and Switching, AWS, and ITIL at StormWind Studios. Raymond treats all technologies as an escape room, working to uncover every mystery in the protocols he works with. Along this journey, Raymond has passed more than 110 exams, and his office wall includes certificates from Microsoft, Cisco, ISC2, ITIL, AWS, and CompTIA. If you were visualizing Raymond's office, you'd probably expect the usual network equipment, certifications, and awards. Those certainly take up space, but they aren't his pride and joy. Most impressive, at least to Raymond, is his gemstone and mineral collection; once he starts talking about it, he just can't stop. Who doesn't get excited by a wondrous barite specimen in a pyrite matrix? Raymond presently resides with his wife and two children in eastern Canada, where they experience many adventures together.

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

About the Technical Reviewers

Hector Mendoza, Jr., No. 10687 (R&S, SP, and Security) has spent the past 14 years at Cisco Systems and is currently a solutions integration architect supporting large SP customers. Prior to this proactive role in CX, he spent nearly a decade providing reactive support in High Touch Technical Services in the Security Group, where he provided escalation support for some of the largest customers for Cisco. A four-time Cisco Live speaker and an Alpha reviewer of Cisco Security courseware, he is a huge advocate of continuing education and knowledge sharing. Hector has a passion for technology, enjoys solving complex problems, and loves working with customers. In his spare time, he tech reviews his esteemed colleagues' Cisco Press books.

Russ Long was introduced to computers and networking at a very young age, when he tried to save the world from digital monsters and aliens, an endeavor that keeps him busy to this day. Russ started his career in enterprise-level IT work splicing fiber-optic networks in the Pacific Northwest. His career has taken a long and winding path from there: from systems administrator, to IT consultant and computer shop owner, to IT instructor. Roughly the last decade of his career has focused solely on instruction and consulting in IT environments. Some of his favorite topics include Cisco routing and switching, real-world security, storage solutions, and virtualization.

Dedications

Raymond Lacoste:

This book is dedicated to my wife, Melanie, who has dedicated her life to making me a better person, which is the hardest job in the world. Thank you, Melanie, for being the most amazing wife and mother in the world.

Brad Edgeworth:

This book is dedicated to my daughter, Teagan. I know that you want to write a book with wizards and princesses, but I don't know how to do that. However, these are your words in a book:

> I can speak in Spanish, English, French, Chinese, and Parseltongue!
>
> —*Teagan Edgeworth*

Acknowledgments

Raymond Lacoste:
A huge thank you goes out to Brad for joining me on this writing adventure. Putting our knowledge together to create this work of art was the best decision. Thank you so much for sharing this with me.

To my wife and children for allowing me to avoid many family adventures while this book was being developed and supporting me though the entire process. Love you guys!

To Russ Long, a long-time friend and a man whom I can trust. Thank you for finding my mistakes before the readers do. You have always been there to make me look my best. (*The R&R Show* for life!)

To Hector Mendoza, Jr.: I don't know you personally, but you found those little things that make a huge difference to the readers, and for that I thank you!

To Brett Bartow, thanks for trusting us to put this book together and put our knowledge on paper.

To MJB, thank you for keeping me on task and making sure nothing slipped through the cracks.

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

Brad Edgeworth:
To Raymond and Brett, thanks for letting me write this book. I am privileged to be able to share my knowledge with others, and I'm grateful. To the rest of the Cisco Press team, thanks for taking my block of stone and turning it into a work of art.

To the technical editors: Hector and Russ, thank you for finding our mistakes before everyone else found them. If any slipped by, I completely blame the both of you.

Many people within Cisco have shared their knowledge with me and taken a chance on me with various projects over the years. For that I'm forever indebted. Special gratitude goes to Craig Smith, Aaron Foss, Ramiro Garza Rios, Vinit Jain, Richard Furr, David Prall, Dustin Schuemann, Tyson Scott, Denise Fishbourne, Tyler Creek, and Mohammad Ali.

Icons Used in This Book


ASA Firewall


LAN Segment


Serial


Switched Circuit


Radio Tower


Routing Domain


Router

Command Syntax Conventions

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

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

Introduction

Congratulations! If you are reading this Introduction, then you have probably decided to obtain your Cisco CCNP Enterprise certification. Obtaining a Cisco certification will ensure that you have a solid understanding of common industry protocols along with Cisco's device architecture and configuration. Cisco has a high market share of routers and switches, with a global footprint.

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 credibility. All other considerations held equal, a certified employee/consultant/job candidate is considered more valuable than one who is not certified.

Cisco provides three primary certifications:

Cisco Certified Network Associate (CCNA), Cisco Certified Network Professional (CCNP), and Cisco Certified Internetwork Expert (CCIE).

Cisco announced changes to all three certifications to take effect in February 2020. The announcement included many changes, but these are the most notable:

- The exams will include additional topics, such as programming.
- The CCNA certification is not a prerequisite for obtaining the CCNP certification. CCNA specializations will not be offered anymore.
- The exams will test a candidate's ability to configure and troubleshoot network devices in addition to answering multiple-choice questions.
- The CCNP is obtained by taking and passing a Core exam and a Concentration exam, like the Implementing Cisco Enterprise Advanced Routing and Services (ENARSI).

CCNP Enterprise candidates need to take and pass the CCNP and CCIE Enterprise Core ENCOR 350-401 examination. Then they need to take and pass one of the following Concentration exams to obtain their CCNP Enterprise:

- 300-410 ENARSI to obtain Implementing Cisco Enterprise Advanced Routing and Services (ENARSI)
- 300-415 ENSDWI to obtain Implementing Cisco SD-WAN Solutions (SDWAN300)
- 300-420 ENSLD to obtain Designing Cisco Enterprise Networks (ENSLD)
- 300-425 ENWLSD to obtain Designing Cisco Enterprise Wireless Networks (ENWLSD)
- 300-430 ENWLSI to obtain Implementing Cisco Enterprise Wireless Networks (ENWLSI)
- 300-435 ENAUTO to obtain Implementing Automation for Cisco Enterprise Solutions (ENAUI)

Goals and Methods

The most important and somewhat obvious goal of this book is to help you pass the CCNP Implementing Cisco Enterprise Advanced Routing and Services (ENARSI) 300-410 exam. In fact, if the primary objective of this book were different, then the book's title would be misleading; however, the methods used in this book to help you pass the exam are designed to also make you much more knowledgeable about how to do your job.

One key methodology used in this book is to help you discover the exam topics that you need to review in more depth, to help you fully understand and remember those details, and to help you prove to yourself that you have retained your knowledge of those topics. This book does not try to help you pass by memorization but helps you truly learn and understand the topics. The ENARSI 300-410 exam covers foundation topics in the CCNP certification, and the knowledge contained within is vitally important for a truly skilled routing/switching engineer or specialist. This book would do you a disservice if it didn't attempt to help you learn the material. To that end, the book will help you pass the exam by using the following methods:

- Helping you discover which test topics you have not mastered
- Providing explanations and information to fill in your knowledge gaps
- Supplying exercises and scenarios that enhance your ability to recall and deduce the answers to test questions
- Providing practice exercises on the topics and the testing process via test questions on the companion website

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 ENARSI 300-410 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.

So why should you want to pass the ENARSI 300-410 exam? Because it's one of the milestones toward getting the CCNP Enterprise certification, which is no small feat. What would getting the CCNP Enterprise certification mean to you? A raise, a promotion, recognition? How about enhancing your resume? Demonstrating that you are serious about continuing the learning process and that you're not content to rest on your laurels? Pleasing your reseller-employer, who needs more certified employees for a higher discount from Cisco? You might have one of these reasons for getting the CCNP Enterprise certification or one of many others.

Strategies for Exam Preparation

The strategy you use for taking the ENARSI 300-410 exam might be slightly different from strategies used by other readers, depending on the skills, knowledge, and

experience you already have obtained. For instance, if you have attended the CCNP Implementing Cisco Enterprise Advanced Routing and Services (ENARSI) 300-410 course, you might take a different approach than someone who learned routing through on-the-job training.

Regardless of the strategy you use or the background you have, this book is designed to help you get to the point where you can pass the exam with the least amount of time required. For instance, there is no need for you to practice or read about IP addressing and subnetting if you fully understand it already. However, many people like to make sure that they truly know a topic and thus read over material that they already know. Several book features will help you gain the confidence you need to be convinced that you know some material already and to also help you know what topics you need to study more.

How This Book Is Organized

Although this book could be read cover-to-cover, it is designed to be flexible and allow you to easily move between chapters and sections of chapters to cover just the material that you need more work with. If you intend to read the entire book, the order in the book is an excellent sequence to use.

The chapters cover the following topics:

- **Chapter 1, "IPv4/IPv6 Addressing and Routing Review":** This chapter provides a review of IPv4 and IPv6 addressing, DHCP, and routing, as well as details about how to troubleshoot these topics.
- **Chapter 2, "EIGRP":** This chapter explains the underlying mechanics of the EIGRP routing protocol, the path metric calculations, and how to configure EIGRP.
- **Chapter 3, "Advanced EIGRP":** This chapter explains the a variety of advanced concepts, such as failure detection, network summarization, router filtering, and techniques to optimize WAN sites.
- **Chapter 4, "Troubleshooting EIGRP for IPv4":** This chapter focuses on how to troubleshoot EIGRP neighbor adjacency issues as well as EIGRP route issues.
- **Chapter 5, "EIGRPv6":** This chapter explains how EIGRP advertises IPv6 networks and guides you through configuring, verifying, and troubleshooting EIGRPv6.
- **Chapter 6, "OSPF":** This chapter explains the core concepts of OSPF, the exchange of routes, OSPF network types, failure detection, and OSPF authentication.
- **Chapter 7, "Advanced OSPF":** This chapter expands on Chapter 6 by explaining the OSPF database and how it builds the topology. It also explains OSPF path selection, router summarization, and techniques to optimize an OSPF environment.
- **Chapter 8, "Troubleshooting OSPFv2":** This chapter explores how to troubleshooting OSPFv2 neighbor adjacency issues as well as route issues.

- **Chapter 9, "OSPFv3":** This chapter explains how the OSPF protocol has changed to accommodate support of the IPv6 protocol.
- **Chapter 10, "Troubleshooting OSPFv3":** This chapter explains how you can troubleshooting issues that may arise with OSPFv3.
- **Chapter 11, "BGP":** This chapter explains the core concepts of BGP, its path attributes, and configuration for IPv4 and IPv6 network prefixes.
- **Chapter 12, "Advanced BGP":** This chapter expands on Chapter 11 by explaining BGP communities and configuration techniques for routers with lots of BGP peerings.
- **Chapter 13, "BGP Path Selection":** This chapter explains the BGP path selection process, how BGP identifies the best BGP path, and methods for load balancing across equal paths.
- **Chapter 14, "Troubleshooting BGP":** This chapter explores how you can identify and troubleshoot issues relating to BGP neighbor adjacencies, BGP routes, and BGP path selection. It also covers MP-BGP (BGP for IPv6).
- **Chapter 15, "Route Maps and Conditional Forwarding":** This chapter explains route maps, concepts for selecting a network prefix, and how packets can be conditionally forwarded out different interfaces for certain network traffic.
- **Chapter 16, "Route Redistribution":** This chapter explains the rules of redistribution, configuration for route redistribution, and behaviors of redistribution based on the source or destination routing protocol.
- **Chapter 17, "Troubleshooting Redistribution":** This chapter focuses on how to troubleshoot issues related to redistribution, including configuration issues, suboptimal routing issues, and routing loop issues.
- **Chapter 18, "VRF, MPLS, and MPLS Layer 3 VPNs":** This chapter explores how to configure and verify VRF and introduces you to MPLS operations and MPLS Layer 3 VPNs.
- **Chapter 19, "DMVPN Tunnels":** This chapter covers GRE tunnels, NHRP, DMVPN, and techniques to optimize a DMVPN deployment.
- **Chapter 20, "Securing DMVPN Tunnels":** This chapter explains the importance of securing network traffic on the WAN and techniques for deploying IPsec tunnel protection for DMVPN tunnels.
- **Chapter 21, "Troubleshooting ACLs and Prefix Lists":** This chapter shows how to troubleshoot issues related to IPv4 and IPv6 access control lists and prefix lists.
- **Chapter 22, "Infrastructure Security":** This chapter covers how to troubleshoot AAA issues, uRPF issues, and CoPP issues. In addition, it introduces various IPv6 First-Hop Security features.
- **Chapter 23, "Device Management and Management Tools Troubleshooting":** This chapter explores how to troubleshoot issues that you might experience with local or

remote access, remote transfers, syslog, SNMP, IP SLA, Object Tracking, NetFlow, and Flexible NetFlow. In addition, it introduces the troubleshooting options available with Cisco DNA Center Assurance.

- The last chapter, **Chapter 24, "Final Preparation,"** provides tips and strategies for studying for the ENARSI 300-410 exam.

Certification Exam Topics and This Book

The questions for each certification exam are a closely guarded secret. However, we do know which topics you must know to *successfully* complete the ENARSI 300-410 exam. Cisco publishes them as an exam blueprint. Table I-1 lists the exam topics from the blueprint along with references to the book chapters that cover each topic. These are the same topics you should be proficient in when working with enterprise technologies in the real world.

Table I-1 Enterprise Core Topics and Chapter References

Implementing Cisco Enterprise Advanced Routing (ENARSI) (300-410) Exam Topic	Chapter(s) in Which Topic Is Covered
1.0 Layer 3 Technologies	
1.1 Troubleshoot administrative distance (all routing protocols)	1
1.2 Troubleshoot route map for any routing protocol (attributes, tagging, filtering)	17
1.3 Troubleshoot loop prevention mechanisms (filtering, tagging, split horizon, route poisoning)	17
1.4 Troubleshoot redistribution between any routing protocols or routing sources	16, 17
1.5 Troubleshoot manual and auto-summarization with any routing protocol	3, 4, 5, 7, 8, 9, 10, 12
1.6 Configure and verify policy-based routing	15
1.7 Configure and verify VRF-Lite	18
1.8 Describe Bidirectional Forwarding Detection	23
1.9 Troubleshoot EIGRP (classic and named mode)	4, 5
1.9.a Address families (IPv4, IPv6)	2, 3, 4, 5
1.9.b Neighbor relationship and authentication	2, 4, 5
1.9.c Loop-free path selections (RD, FD, FC, successor, feasible successor, stuck in active)	3, 4
1.9.d Stubs	4
1.9.e Load balancing (equal and unequal cost)	2
1.9.f Metrics	2
1.10 Troubleshoot OSPF (v2/v3)	6, 7, 8, 9, 10
1.10.a Address families (IPv4, IPv6)	8, 10
1.10.b Neighbor relationship and authentication	6, 8, 10

Implementing Cisco Enterprise Advanced Routing (ENARSI) (300-410) Exam Topic	Chapter(s) in Which Topic Is Covered
1.10.c Network types, area types, and router types	8, 10
1.10.c (i) Point-to-point, multipoint, broadcast, nonbroadcast	6, 8, 10
1.10.c (ii) Area type: backbone, normal, transit, stub, NSSA, totally stub	7, 8, 10
1.10.c (iii) Internal router, backbone router, ABR, ASBR	6, 8, 10
1.10.c (iv) Virtual link	7, 8
1.10.d Path preference	7
1.11 Troubleshoot BGP (Internal and External)	11, 12, 13, 14
1.11.a Address families (IPv4, IPv6)	10, 14
1.11.b Neighbor relationship and authentication (next-hop, mulithop, 4-byte AS, private AS, route refresh, synchronization, operation, peer group, states and timers)	10, 14
1.11.c Path preference (attributes and best-path)	13, 14
1.11.d Route reflector (excluding multiple route reflectors, confederations, dynamic peer)	10
1.11.e Policies (inbound/outbound filtering, path manipulation)	11, 14
2.0 VPN Technologies	
2.1 Describe MPLS operations (LSR, LDP, label switching, LSP)	18
2.2 Describe MPLS Layer 3 VPN	18
2.3 Configure and verify DMVPN (single hub)	19, 20
2.3.a GRE/mGRE	19
2.3.b NHRP	19
2.3.c IPsec	20
2.3.d Dynamic neighbor	19
2.3.e Spoke-to-spoke	19
3.0 Infrastructure Security	
3.1 Troubleshoot device security using IOS AAA (TACACS+, RADIUS, local database)	22
3.2 Troubleshoot router security features	
3.2.a IPv4 access control lists (standard, extended, time-based)	21
3.2.b IPv6 traffic filter	21
3.2.c Unicast reverse path forwarding (uRPF)	22
3.3 Troubleshoot control plane policing (CoPP) (Telnet, SSH, HTTP(S), SNMP, EIGRP, OSPF, BGP)	22
3.4 Describe IPv6 First Hop Security features (RA Guard, DHCP Guard, binding table, ND inspection/snooping, Source Guard)	22
4.0 Infrastructure Services	
4.1 Troubleshoot device management	23
4.1.a Console and VTY	23

Implementing Cisco Enterprise Advanced Routing (ENARSI) (300-410) Exam Topic	Chapter(s) in Which Topic Is Covered
4.1.b Telnet, HTTP, HTTPS, SSH, SCP	23
4.1.c (T)FTP	23
4.2 Troubleshoot SNMP (v2c, v3)	23
4.3 Troubleshoot network problems using logging (local, syslog, debugs, conditional debugs, timestamps)	23
4.4 Troubleshoot IPv4 and IPv6 DHCP (DHCP client, IOS DHCP server, DHCP relay, DHCP options)	1
4.5 Troubleshoot network performance issues using IP SLA (jitter, tracking objects, delay, connectivity)	23
4.6 Troubleshoot NetFlow (v5, v9, flexible NetFlow)	23
4.7 Troubleshoot network problems using Cisco DNA Center assurance (connectivity, monitoring, device health, network health)	23

Each version of the exam can have topics that emphasize different functions or features, and some topics can be rather broad and generalized. The goal of this book is to provide the most comprehensive coverage to ensure that you are well prepared for the exam. Although some chapters might not address specific exam topics, they provide a foundation that is necessary for a clear understanding of important topics.

It is also important to understand that this book is a "static" reference, whereas the exam topics are dynamic. Cisco can and does change the topics covered on certification exams often.

This exam guide should not be your only reference when preparing for the certification exam. You can find a wealth of information at Cisco.com that covers each topic in great detail. If you think that you need more detailed information on a specific topic, read the Cisco documentation that focuses on that topic.

Note that as technologies continue to evolve, Cisco reserves the right to change the exam topics without notice. Although you can refer to the list of exam topics in Table I-1, always check Cisco.com to verify the actual list of topics to ensure that you are prepared before taking the exam. You can view the current exam topics on any current Cisco certification exam by visiting https://www.cisco.com/c/en/us/training-events/training-certifications/next-level-certifications.html. Note also that, if needed, Cisco Press might post additional preparatory content on the web page associated with this book: http://www.ciscopress.com/title/9781587145254. It's a good idea to check the website a couple weeks before taking your exam to be sure that you have up-to-date content.

Learning in a Lab Environment

This book is an excellent self-study resource for learning the technologies. However, reading is not enough, and any network engineer can tell you that you must implement a technology to fully understand it. We encourage the reader to re-create the topologies and technologies and follow the examples in this book.

A variety of resources are available for practicing the concepts in this book. Look online for the following:

- Cisco VIRL (Virtual Internet Routing Lab) provides a scalable, extensible network design and simulation environment. For more information about VIRL, see http://virl.cisco.com.
- Cisco dCloud provides a huge catalog of demos, training, and sandboxes for every Cisco architecture. It offers customizable environments and is free. For more information, see http://dcloud.cisco.com.
- Cisco Devnet provides many resources on programming and programmability, along with free labs. For more information, see http://developer.cisco.com.

CHAPTER 1

IPv4/IPv6 Addressing and Routing Review

This chapter covers the following topics:

- **IPv4 Addressing:** This section provides a review of IPv4 addressing and covers issues you might face and how to troubleshoot them.
- **DHCP for IPv4:** This section reviews DHCP for IPv4 operations, explores potential DHCP issues, and examines the output of various DHCP **show** commands.
- **IPv6 Addressing:** This section provides a brief review of IPv6 addressing.
- **IPv6 SLAAC, Stateful DHCPv6, and Stateless DHCPv6:** This section explores how clients obtain IPv6 addressing information using SLACC, stateful DHCPv6, and stateless DHCPv6.
- **Packet-Forwarding Process:** This section discusses the packet-forwarding process and the commands 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 useful when troubleshooting related issues.
- **Routing Information Sources:** This section explains which sources of routing information are the most believable and how the routing table interacts with various data structures to populate itself with the best information.
- **Static Routes:** This section reviews how to configure and verify IPv4 and IPv6 static routes.
- **Trouble Tickets:** This section provides a number of trouble tickets that demonstrate how a structured troubleshooting process is used to solve a reported problem.

IPv6 is currently being deployed, but that deployment is occurring at a slow pace. Most networks still rely on IPv4, and many new networks and network additions are being deployed with IPv4. Therefore, you still need the skills to successfully configure, verify, and troubleshoot IPv4 addressing. Therefore, this chapter provides a review of IPv4 addressing.

Typically, when deploying IPv4 addresses, Dynamic Host Configuration Protocol (DHCP) is used so that addresses can be dynamically assigned. However, with this dynamic process, issues may arise that prevent a device from successfully obtaining an IPv4 address from a DHCP server. Therefore, this chapter reviews how DHCP operates and how to identify the issues that may prevent a client from obtaining an IP address from a DHCP server.

Sooner or later, organizations will have to switch to IPv6. There is a whole lot more to IPv6 than just having a larger address space than IPv4. This chapter reminds you how

IPv6-enabled devices determine whether a destination is local or remote and explores the various options for address assignment and what to look out for when troubleshooting.

Before you dive into the advanced routing topics such as Enhanced Interior Gateway Routing Protocol (EIGRP), Open Shortest Path First (OSPF), and Border Gateway Protocol (BGP), you need to review 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. Either way, the router goes through the same steps to successfully take a packet from an ingress interface and packet switch it to the egress interface. You also need to review how a router populates the routing table with "the best" routes. What classifies those routes as the best? Is an EIGRP-learned route better than a static route? What about an OSPF-learned route or a BGP-learned route? How do they compare to the other sources of routing information? When multiple sources provide the same routing information, 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; therefore, this chapter reviews IPv4 and IPv6 static routing configuration and verification.

Notice that this chapter is mostly a review of IPv4/IPv6 addressing, DHCP for IPv4/IPv6, the packet-forwarding process, administrative distance, and static routing that you learned in CCNA or ENCORE. I encourage you not to skip this chapter as it is a great place to warm up for what is to come in the rest of this book, which prepares you for the Implementing Cisco Enterprise Advanced Routing and Services (ENARSI) exam.

"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?' Quiz Questions."

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

Foundation Topics Section	Questions
IPv4 Addressing	1–3
DHCP for IPv4	4–6
IPv6 Addressing	7–8
IPv6 SLAAC, Stateful DHCPv6, and Stateless DHCPv6	9–12
Packet-Forwarding Process	13–15
Routing Information Sources	16–17
Static Routes	18–19

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 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 occurs when a PC with the IP address 10.1.1.27/28 needs to communicate with a PC that has IP address 10.1.1.18? (Choose two.)
   a. It sends the frame to its default gateway.
   b. It sends the frame directly to the destination PC.
   c. It uses ARP to get the MAC address of the default gateway.
   d. It uses ARP to get the MAC address of the destination PC.
2. What occurs when a PC with the IP address 10.1.1.27/29 needs to communicate with a PC that has IP address 10.1.1.18? (Choose two.)
   a. It sends the frame to its default gateway.
   b. It sends the frame directly to the destination PC.
   c. It uses ARP to get the MAC address of the default gateway.
   d. It uses ARP to get the MAC address of the destination PC.
3. 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**
4. 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
5. 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**
6. Which command enables 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**

7. 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
8. Which of the following are true when using EUI-64? (Choose two.)
   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.
9. 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**
10. Which of the following are requirements for stateless address autoconfiguration to function? (Choose three.)
    a. The prefix must be /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.
11. 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**
12. 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**
13. Which two data structures reside at the router's data plane?
    a. IP routing table
    b. ARP cache
    c. Forwarding Information Base
    d. Adjacency table

14. 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**
15. Which of the following populate a routing protocol's data structure, such as the EIGRP topology table? (Choose three.)
    a. Updates from a neighbor
    b. Redistributed routes
    c. Interfaces enabled for the routing process
    d. Static routes
16. Which of the following has the lowest default administrative distance?
    a. OSPF
    b. EIGRP (internal)
    c. RIP
    d. eBGP
17. What is the default administrative distance of an OSPF intra-area route?
    a. 90
    b. 110
    c. 115
    d. 120
18. 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.
19. What occurs when you create an IPv4 static route with an Ethernet interface designated instead of a next-hop IP address?
    a. The router uses ARP to get the MAC address of the directly connected router's IP address.
    b. The router forwards the packet with the destination MAC address FFFF:FFFF:FFFF.
    c. The router uses ARP to get the MAC address of the IP address in the source of the packet.
    d. The router uses ARP to get the MAC address of the IP address in the destination of the packet.

Foundation Topics

IPv4 Addressing

Just as 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 is unique within your city/town. As a result, a pizza delivery person can bring your pizza to your house in 30 minutes, or it is free. If your house is addressed incorrectly, you may not get your pizza, and you do not want that to happen.

Similarly, with IPv4 addressing, if devices are addressed incorrectly, they 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 a network. This section provides a review of IPv4 addressing and discusses issues you might face and how to troubleshoot them.

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 exactly the same network/subnet portion. If they are not the same, the PC could end up addressing the Layer 2 frame incorrectly and sending the packet in the wrong direction. Figure 1-1 shows a sample subnet (10.1.1.0/26) with two PCs and their default gateway, R1.

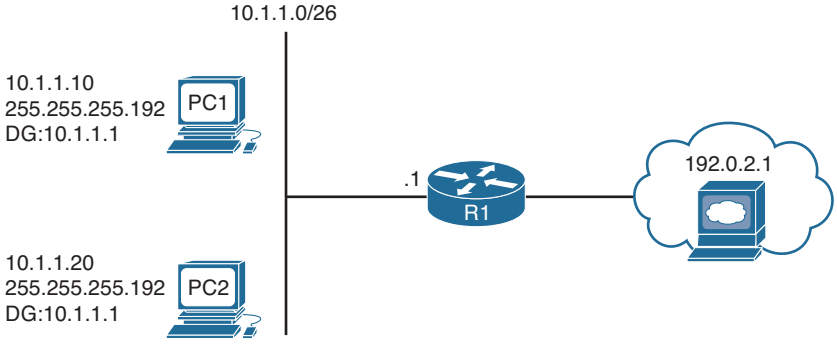

Figure 1-1 *Correct IPv4 Addressing Example*

Key Topic

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 determines whether the frame has the MAC address of PC2 or the MAC address 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 exactly the 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 communicates directly with it and does not need to send the data to its default gateway. PC1 creates a frame with its own source MAC address and the MAC address 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 determines whether the frame has the MAC address of the web server or the MAC address 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 exactly the 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 creates a frame with its own source MAC address and the MAC address of R1 as the destination.

As you can see, accurate IP addressing is paramount for successful communication. Let's look at what happens if PC1 is configured with the wrong subnet mask (255.255.255.240), as shown in Figure 1-2.

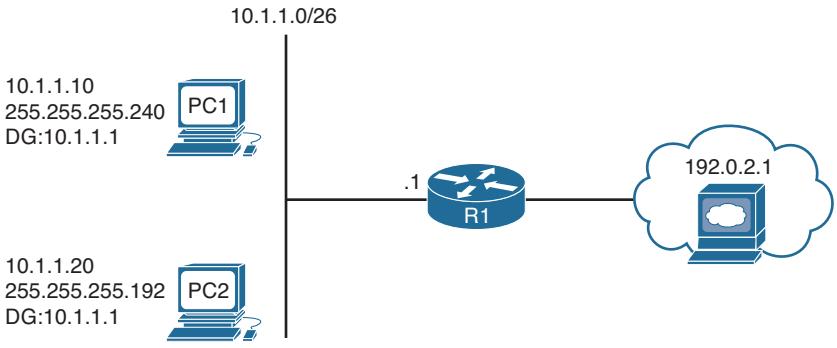

Figure 1-2 *Incorrect IPv4 Addressing Example*

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 exactly the 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 the same. Therefore, it cannot communicate directly with it and needs 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, there is an IPv4 addressing and connectivity issue.

Not only does an *improper subnet mask* cause issues, but an *inappropriate IP address combined with the correct subnet mask* also causes issues. In addition, if the *default gateway is not configured correctly* on the PCs, packets are not forwarded to the correct device when packets need to be sent to a different subnet.

As a troubleshooter, you must recognize these issues and eliminate them as possible issues quickly. You verify the IP addressing information on a Windows PC by using the **ipconfig** command, as shown in Example 1-1. On an IOS router or IOS switch, you verify IP addressing information by using the **show ip interface** *interface_type interface_number* command, as also shown in Example 1-1.

Example 1-1 *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::4107:2cfb:df25:5124%7
 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

This section describes a quick way to determine all the IP addresses that will be in a particular subnet. Refer to Figure 1-3 as you are exploring this method.

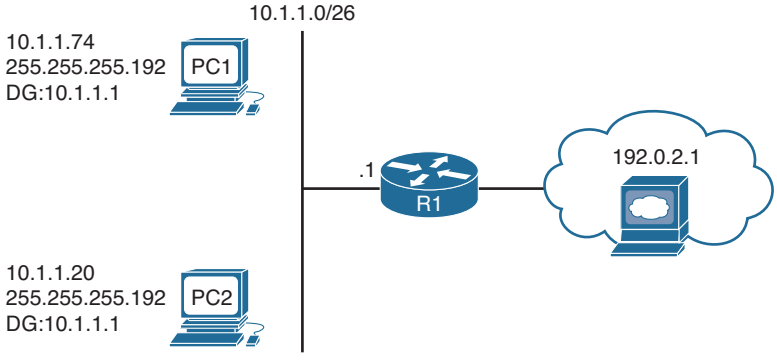

Figure 1-3 *Determining IP Addresses Within a Subnet*

In the subnet mask, find the most interesting octet. In binary, it's the octet with the last binary 1. In decimal, it's the last octet that is greater than 0. In this case, for 255.255.255.192, the fourth octet is the last octet with a value great than 0. The value of this octet is 192. If your subnet mask were 255.255.192.0, then it would be the third octet. Consider the subnet mask 255.255.255.0. Because the fourth octet is a 0, it would be the third octet, as it's the last octet with a value greater than 0.

Now, subtract 192 from 256. The result is 64. The number 64 represents the block size or the number you are counting by in that octet. The subnet in this case is 10.1.1.0/26, and because the block size is 64, this subnet begins at 10.1.1.0/26 and ends at 10.1.1.63/26. The next subnet is 10.1.1.64/26 to 10.1.1.127/26. The third subnet is 10.1.1.128/26 to 10.1.1.191/26, and so on.

Now 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 1-2, 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, but it should be in the same subnet, according to Figure 1-3. You must fix the address on PC1 so that it is within the correct network/subnet.

Example 1-2 *Verifying IP Addressing on a 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::4107:2cfb:df25:5124%7
 Default Gateway . . . . . . . . . : 10.1.1.1
```

DHCP for IPv4

Dynamic Host Configuration Protocol (DHCP) is commonly used for 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 addressing 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 using DHCP 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 through 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 1-4 illustrates the exchange of messages (Discover, Offer, Request, Acknowledgment [DORA] process) that occurs as a DHCP client obtains IP addressing information from a DHCP server.

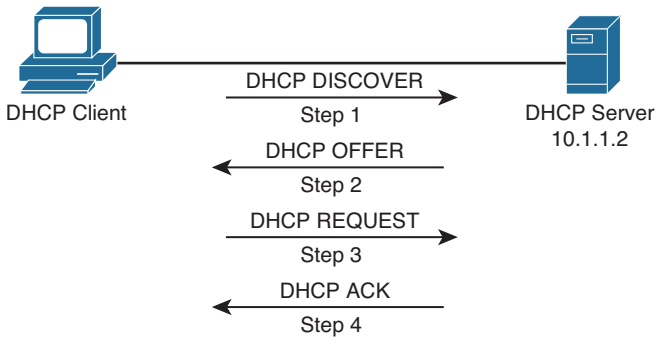

Figure 1-4 *DHCP DORA Process*

Key Topic

The DORA process works as follows:

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 destination IP address 255.255.255.255 and destination MAC address FFFF:FFFF:FFFF in an attempt to discover a DHCP server. The source IP address is 0.0.0.0, and the source MAC address is 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, such as the lease duration.

Notice that in step 1, the DHCPDISCOVER message is sent as a broadcast. The broadcast cannot cross a router boundary. Therefore, if a client resides on a different network from the DHCP server, you need to configure the default gateway of the client as a DHCP relay agent to forward the broadcast packets as unicast packets to the server. You 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 1-5 and Example 1-3. 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, using the syntax shown in Example 1-3.

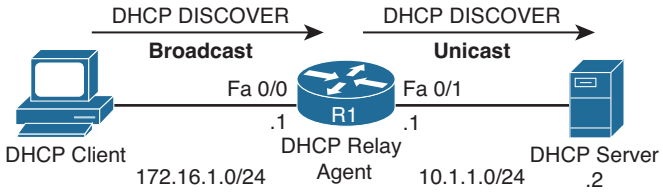

Figure 1-5 *DHCP Relay Agent*

Key Topic

Example 1-3 *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 are 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 it isn't, 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 1-2 provides a comprehensive list of DHCP message types you might encounter while troubleshooting a DHCP issue.

Table 1-2 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 broadcast IP address 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 DHCPOFFER 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 1-6 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 1-4 provides the configuration for the router in the topology (that is, router R1). Notice that the **dhcp** option is used in the **ip address** command, instead of the usual IP address and subnet mask information.

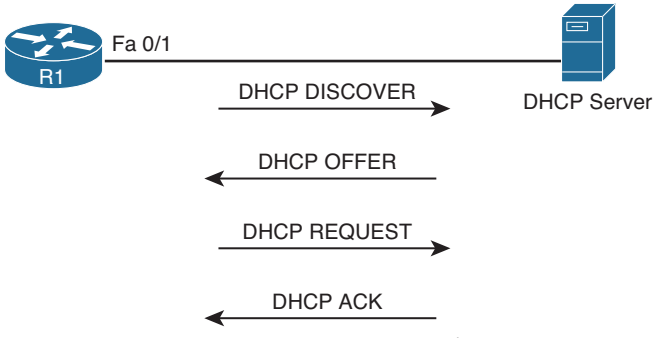

Figure 1-6 *Router Acting as a DHCP Client*

Key Topic

The following snippet shows a 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 may also act as a DHCP server. Figure 1-7 shows a router acting as a DHCP server, and Example 1-4 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 never hands out its own interface IP address. The **ip dhcp pool POOL-A** command creates a DHCP pool named POOL-A. This pool hands 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.

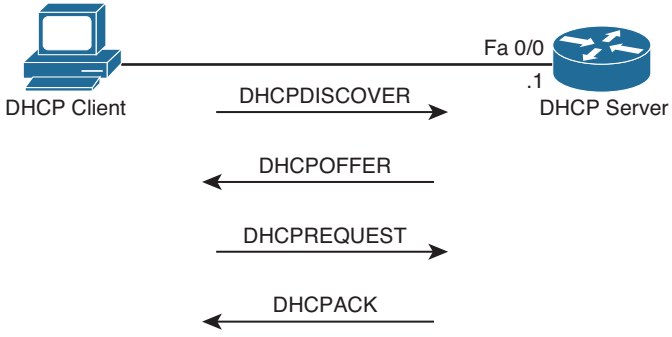

Figure 1-7 *Router Acting as a DHCP Server*

Example 1-4 *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 shown on the client is an Automatic Private IP Addressing (APIPA) address (169.254.x.x) because of autoconfiguration, as shown in Example 1-5, conclude that the client could not obtain an IP address from the DHCP server. However, 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, that is, for example, preventing the client's DHCPDISCOVER message from reaching the DHCP server.

Example 1-5 *Verifying DHCP-Assigned IP Address on a PC*

```
C:\>ipconfig /all
Windows IP Configuration

...output omitted...

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
 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:

Key Topic

- **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 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 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 was statically configured for the IP address.
- **Redundant services not communicating:** Some DHCP servers 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 hand out overlapping IP addresses to their client's.

- **The "pull" nature of DHCP:** When a DHCP client wants an IP address, it requests 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, the DHCP server cannot push information changes 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 only hands 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.

DHCP Troubleshooting Commands

The following snippet provides sample output from the **show ip dhcp conflict** command:

```
R1# show ip dhcp conflict

IP address    Detection method    Detection time

172.16.1.3    Ping                Oct 15 2018 8:56 PM
```

The output indicates a duplicate 172.16.1.3 IP address on the network, which the router discovered via a ping. You clear the information displayed by issuing the **clear ip dhcp conflict** * command after resolving the duplicate address issue on the network.

Example 1-6 shows sample output from the **show ip dhcp binding** command. The output indicates that IP address 10.1.1.10 was assigned to a DHCP client. You can release this DHCP lease with the **clear ip dhcp binding** * command.

Example 1-6 *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 2018 07:53 PM   Automatic
10.1.1.10     0108.0027.5d06.d6   Oct 17 2018 07:53 PM   Automatic
```

Example 1-7 shows sample output from the **debug ip dhcp server events** command. The output shows updates to the DHCP database.

Example 1-7 *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 1-8 shows sample output from the **debug ip dhcp server packet** command. The output shows a DHCPRELEASE message being received when a DHCP client with IP address 10.1.1.3 is shut down. You can also see the four-step process of a DHCP client obtaining IP address 10.1.1.4 with the following messages: DHCPDISCOVER, DHCPOFFER, DHCPREQUEST, and DHCPACK.

Example 1-8 *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.
```

IPv6 Addressing

Just as 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 parts is unique. Similarly, an IPv6 address is 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 are successfully routed to and from them. Refer to Figure 1-8, 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 interface IPv6 address 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 the default gateway address of R1's Gig0/0 interface, which is 2001:db8:a:a::1.

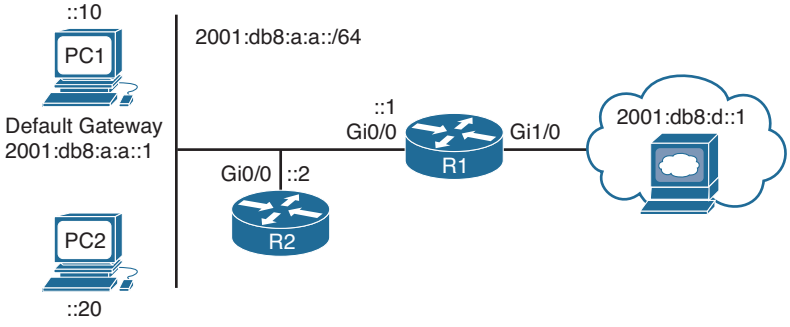

Figure 1-8 *IPv6 Addressing Example*

Key Topic

Just as with IPv4, when a host wants to communicate with another host, it compares its subnet bits to exactly the same bits in the destination IP address. If they match, both devices are in the same subnet; if they do not match, the 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 need to communicate through the default gateway.

For example, when PC1 in Figure 1-8 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. If PC1 wants to communicate with PC2, it determines it is in the same subnet and communicates directly with it.

You verify the IPv6 address of a Windows PC by using the **ipconfig** command, as shown in Example 1-9. In this example, PC1 has the link-local address fe80::a00:27ff:fe5d:6d6 and the global unicast address 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, and it is needed so that the system knows which interface to send the packets out of; keep in mind that you can have multiple interfaces on the same device with the same link-local address assigned to it.

Example 1-9 *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 an 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, it 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 randomly or based on the IEEE EUI-64 standard.

EUI-64 takes the client's 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 1-9. Notice that the link-local address is fe80::a00:27ff:fe5d:6d6. The subnet ID is FE80::, and the interface ID is a00:27ff:fe5d:6d6. If you fill in the missing leading 0s, the address is 0a00:27ff:fe5d:06d6. This is an EUI-64 interface ID because it has FFFE in it. Let's look at how it is derived.

Example 1-10 shows 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 to get 08-00-27-FF-FE-5D-06-D6. Now group the hex values into groups of four and replace each dash (-) with a colon, like this: 0800:27FF:FE5D:06D6. This looks very close to what is listed in the link-local address, but it is not exactly the same. 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. Flip it. 08 hex in binary is 00001000. The seventh bit from left to right is a 0, so make it a 1. Now you have 00001010. Convert to hex, and you get 0a. So, your interface ID is 0A00:27FF:FE5D:06D6.

Example 1-10 *Using* ***ipconfig /all*** *to Verify IPv6 Addressing*

```
C:\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 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 use EUI-64 when generating the interface portion of the link-local address of an interface. Modern Windows PCs 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, use the **eui-64** keyword at the end of the **ipv6 address** command, as shown in Example 1-11.

Example 1-11 *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 verify the global unicast address and the EUI-64 interface ID assigned to an interface by using the **show ipv6 interface** command, as shown in Example 1-12. In this case, R2's Gig0/0 interface has a global unicast address that obtained the interface ID from the EUI-64 standard.

Key Topic

Example 1-12 *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...
```

IPv6 SLAAC, Stateful DHCPv6, and Stateless DHCPv6

Manually assigning IP addresses (either IPv4 or IPv6) is not a scalable option. With IPv4, DHCP provides a dynamic addressing option. With IPv6, you have three dynamic options to choose from: stateless address autoconfiguration (SLAAC), stateful DHCPv6, or stateless DHCPv6. This section looks at the issues that might arise for each and how to troubleshoot them.

SLAAC

SLAAC is designed to enable a device to configure its own IPv6 address, prefix, and default gateway without a DHCPv6 server. Windows PCs automatically have SLAAC enabled and generate their own IPv6 addresses, as shown in Example 1-13, which displays the output of **ipconfig /all** on PC1.

Example 1-13 *Using **ipconfig /all** to Verify That IPv6 SLAAC Is Enabled*

```
C:\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 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 1-14.

Key Topic

Example 1-14 *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 send a Router Solicitation (RS) message to determine whether there are any routers connected to the local link. They then wait for a router to send a Router Advertisement (RA) that identifies the prefix being used by the router (the default gateway) connected to the same network they are on. They 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 uses EUI-64 for the interface portion, and the PC randomly generates the interface portion unless it is configured to use EUI-64. In addition, the PC uses the IPv6 link-local address of the device that sent the RA as the default gateway address.

Key Topic

Figure 1-9 shows 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 FF02::1. The destination MAC address is the all-nodes destination MAC address 33:33:00:00:00:01, which 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.

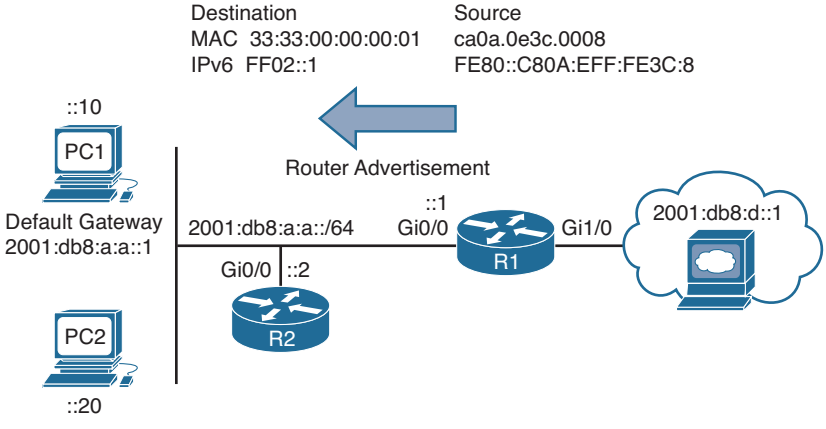

Figure 1-9 *Router Advertisement Example*

When PC1 in Figure 1-9 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 1-15, which displays the output of **ipconfig** on PC1.

Example 1-15 *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 1-16, 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 occurs only if IPv6 unicast routing was not enabled on the router and, as a result, the router is acting as an end device.

Example 1-16 *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
...output omitted...
 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:

Key Topic

- 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 the following snippet:

```
R1# show run | include ipv6 unicast-routing
ipv6 unicast-routing
```

- Make sure that the appropriate interface is enabled for IPv6 by using the **show ipv6 interface** command, as shown in Example 1-17.
- Make sure that the router interface advertising RAs has a /64 prefix by using the **show ipv6 interface** command, as shown in Example 1-17. (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 1-18 (where they are being suppressed).

Key Topic

Example 1-17 *Verifying That 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 1-18 *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
 ...output omitted...
 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 a subnet generating RAs, which is normal when you have redundant default gateways, the clients learn about multiple default gateways from the RAs, as shown in Example 1-19. 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 1-8. If PC1 uses R2 as the default gateway, the packets to the web server are dropped because R2 does not have a way to route packets to the web server, as shown in the ping output of Example 1-20, 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.

Key Topic

Example 1-19 *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: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
```

Example 1-20 *Failed Ping from PC1 to 2001:db8:d::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 information such as Network Time Protocol (NTP) server information, domain name information, DNS server information, and Trivial File Transfer Protocol (TFTP) server information. To hand out the IPv6 addressing information along with all optional information, use a DHCPv6 server. Both Cisco routers and multilayer switches may act as DHCP servers. Example 1-21 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 are receiving wrong IPv6 addressing information from a router or multilayer switch acting as a DHCPv6 server, check the interface and make sure it was associated with the correct pool.

Key Topic

Example 1-21 *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 cisco.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
```

Example 1-22 provides examples of the **show ipv6 dhcp binding** command, which displays the IPv6 addresses used by clients, the **show ipv6 dhcp interface** command, which displays the interface to DHCPv6 pool associations, and the **show ipv6 dhcp pool** command, which displays the configured pools.

Key Topic

Example 1-22 *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 2018 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: cisco.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 the IPv6 address, prefix, and default gateway. Included in the RA is a flag that tells the client to get other non-addressing information from a DHCPv6 server, such as the address of a DNS server or a TFTP server. To accomplish this, 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 1-23, notice this command configured under the Gigabit Ethernet 0/0 interface. Also, in Example 1-23, the output of **show ipv6 interface gigabitEthernet 0/0** states that hosts obtain IPv6 addressing from *stateless autoconfig* and other information from a *DHCP server*.

Key Topic

Example 1-23 *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.
```

DHCPv6 Operation

DHCPv6 has a four-step negotiation process, like IPv4. However, DHCPv6 uses the following messages:

Key Topic

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 1-3 provides a comprehensive list of DHCPv6 message types you might encounter while troubleshooting a DHCPv6 issue.

Table 1-3 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 that it is available.
REQUEST	This message is a request for IP configuration parameters sent from a client to a specific DHCPv6 server.
CONFIRM	A client sends this message to a server to determine whether the address it was assigned is still appropriate.
RENEW	A client sends this message to the server that assigned the address in order to extend the lifetime of the addresses assigned.
REBIND	When there is no response to a RENEW, a client sends a REBIND message to a server to extend the lifetime on the address assigned.
REPLY	A server sends this message to a client containing assigned address and configuration parameters in response to a SOLICIT, REQUEST, RENEW, or REBIND message received from a client.
RELEASE	A client sends this message to a server to inform the server that the assigned address is no longer needed.
DECLINE	A client sends this message to a server to inform the server that the assigned address is already in use.
RECONFIGURE	A server sends this message to a client when the server has new or updated information.
INFORMATION-REQUEST	A client sends this message to a server when the client only needs additional configuration information without any IP address assignment.
RELAY-FORW	A relay agent uses this message to forward messages to DHCP server.
RELAY-REPL	A DHCP server uses this message to reply to the relay agent.

DHCPv6 Relay Agents

All the DHCPv6 examples so far have included the DHCP server within the same local network. However, in most networks, the DHCP server is located in a different network, which creates an issue. If you review the multicast address of the SOLICIT message, notice that it

is a link-local scope multicast address. It starts with FF02. Therefore, the multicast does not leave the local network, and the client is not 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 to needs to be configured as a relay agent with the **ipv6 dhcp relay destination** interface configuration command. Example 1-24 shows interface Gigabit Ethernet 0/0 configured with the command **ipv6 dhcp relay destination 2001:db8:a:b::7**, which is used to forward SOLICIT messages to a DHCPv6 server at the address listed.

Key Topic

Example 1-24 *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
```

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 through 7). However, if the pings fail, you should focus your troubleshooting efforts on Layers 1 through 3. If you ultimately determine that there is a problem at Layer 3, your efforts might be centered on the packet-forwarding process of a router.

This section discusses the packet-forwarding process and the commands used 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 are useful when troubleshooting related issues.

Reviewing the Layer 3 Packet-Forwarding Process

To review basic routing processes, consider Figure 1-10. In this topology, PC1 needs to access HTTP resources on Server1. Notice that PC1 and Server1 are on different networks. So how does a packet from source IP address 192.168.1.2 get routed to destination IP address 192.168.3.2?

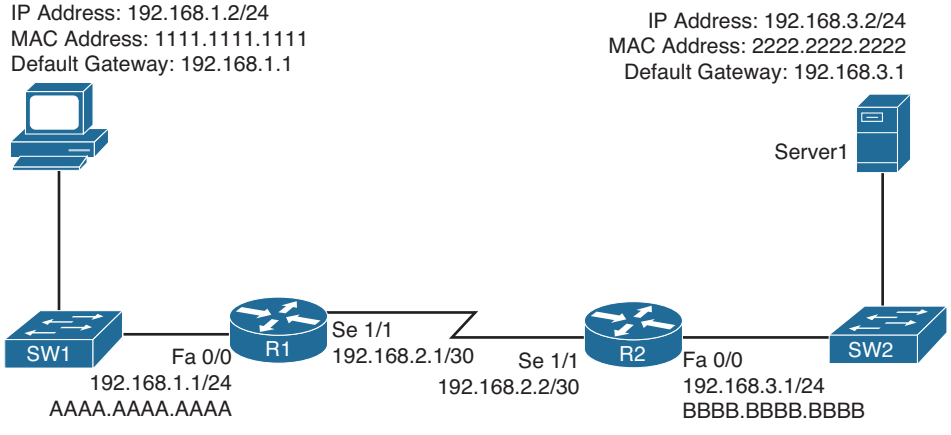

Figure 1-10 *Basic Routing Topology*

Consider the following step-by-step walkthrough of this process:

Step 1. PC1 compares its IP address and subnet mask 192.168.1.2/24 with the destination IP address 192.168.3.2, as discussed earlier in the chapter. 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, it knows the destination is on the same subnet. If they differ, it knows 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 DHCP. In this example, PC1 has the default gateway address 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 then sends its data destined for Server1 in a frame addressed to R1, as shown in Figure 1-11.

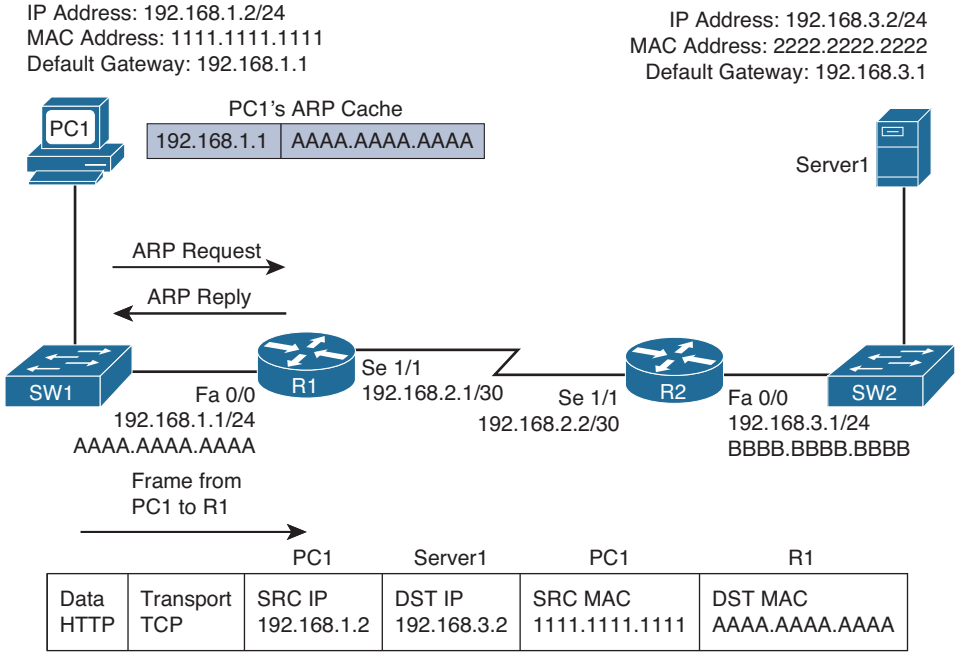

Figure 1-11 *Basic Routing, Step 1*

Step 2. Router R1 receives the frame sent from PC1, and because the destination MAC address 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 that 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 through interface Serial 1/1. Note that ARP is not required for serial interfaces because these interface types do not have MAC addresses. Therefore, router R1 forwards the frame out its Serial 1/1 interface, as shown in Figure 1-12, using the Point-to-Point Protocol (PPP) Layer 2 framing header.

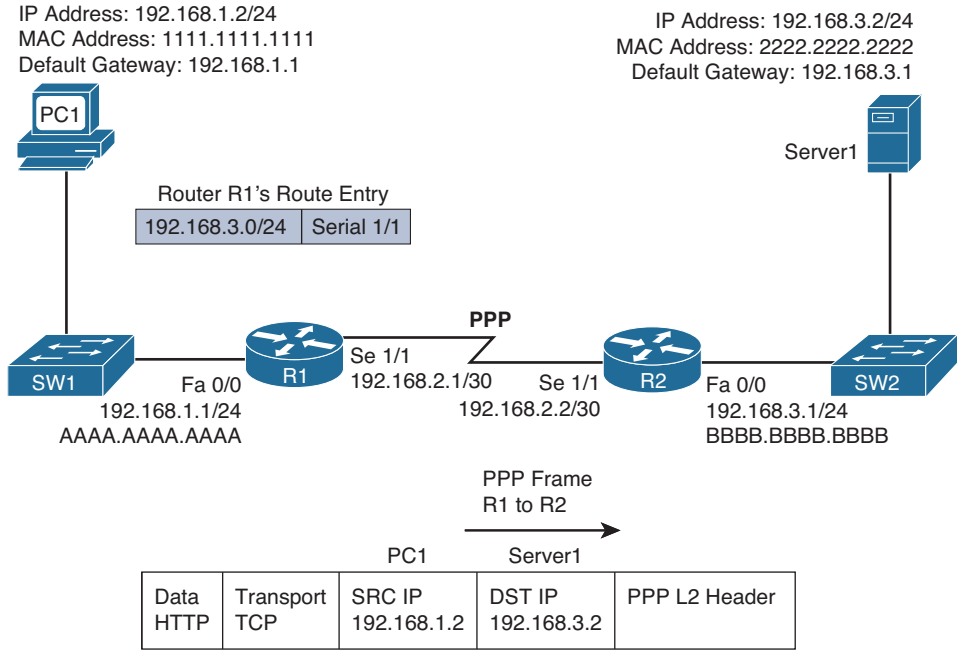

Figure 1-12 *Basic Routing, Step 2*

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 that 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 192.168.3.0/24 is directly attached to router R2's Fast Ethernet 0/0 interface. Much the way 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 stores the results of the ARP reply in the ARP cache and forwards the frame out its Fast Ethernet 0/0 interface to Server1, as shown in Figure 1-13.

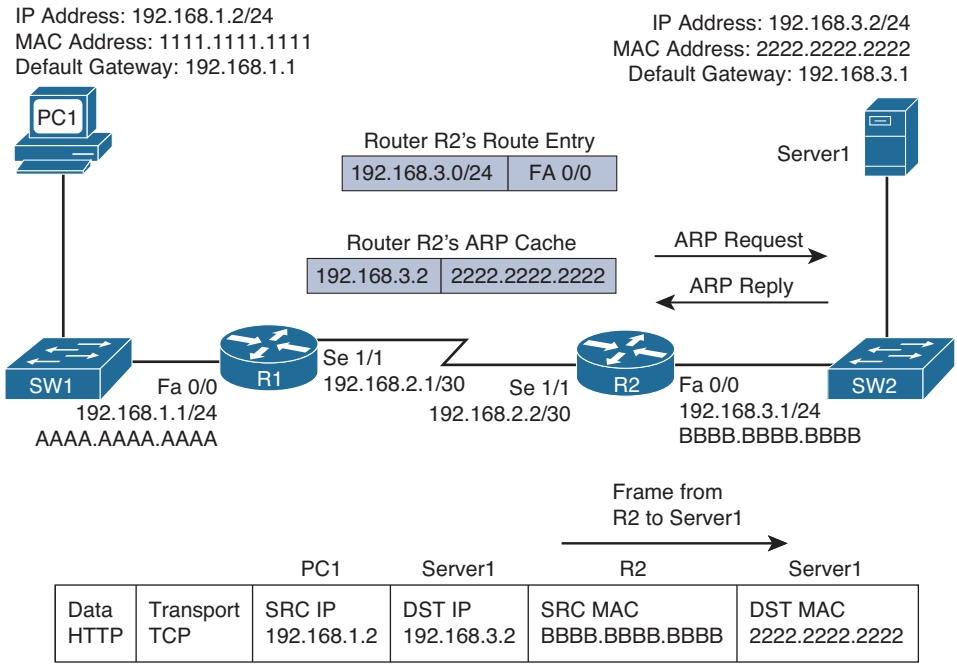

Figure 1-13 *Basic Routing, Step 3*

Key Topic

The previous steps identified two router data structures:

- **IP routing table:** When a router needs to route an IP packet, it consults 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 networks 10.0.0.0/8, 10.1.1.0/24, and 10.1.1.0/26. Also, suppose that the router is trying to forward a packet with the destination IP address 10.1.1.10. The router selects the 10.1.1.0/26 route entry as the best match for 10.1.1.10 because that route entry has the longest prefix, /26 (so it matches the most number of bits).
- **Layer 3-to-Layer 2 mapping table:** In Figure 1-13, router R2's ARP cache contains Layer 3-to-Layer 2 mapping information. Specifically, the ARP cache has a mapping that says MAC address 2222.2222.2222 corresponds to IP address 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 Network (DMVPN) networks. However, for point-to-point links such as PPP or High-Level Data Link Control (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) gleans its information from the router's IP routing table and Layer 3-to-Layer 2 mapping tables. Then, CEF's data structures in hardware can be referenced when forwarding packets.

Key Topic

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 is in an adjacency entry for a multipoint Ethernet interface, whereas a point-to-point interface requires only egress interface information.

As a reference, Figure 1-14 shows the router data structures.

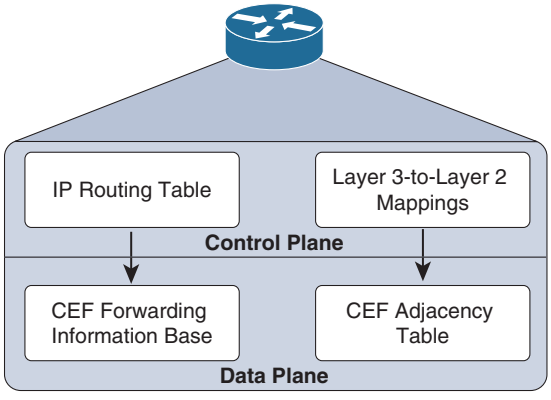

Figure 1-14 *A Router's Data Structures*

Troubleshooting the Packet-Forwarding Process

When troubleshooting packet-forwarding issues, you need to examine a router's IP routing table. If the observed behavior of the traffic 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. You need 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 1-25 provides sample output from the **show ip route** *ip_address* command. The output shows that the next-hop IP address to reach IP address 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.

Key Topic

Example 1-25 *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 1-26 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 interface Fast Ethernet 0/0, with next-hop IP address 192.168.0.11.

Example 1-26 *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 1-27 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, with the **longer-prefixes** option added, two routes are displayed, because these routes are subnets of the 172.16.0.0/16 network.

Example 1-27 *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 1-28 provides sample output from the **show ip cef** *ip_address* command. The output indicates that, according to CEF, IP address 192.168.1.11 is accessible out interface Fast Ethernet 0/0, with the next-hop IP address 192.168.0.11.

Key Topic

Example 1-28 *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 1-29 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 interface Fast Ethernet 0/0, with the next-hop IP address 192.168.0.11.

Example 1-29 *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
```

The following snippet provides sample output from the **show ip cef exact-route** *source_address destination_address* command:

```
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)
```

The output indicates that a packet sourced from IP address 10.2.2.2 and destined for IP address 192.168.1.11 will be sent out interface Fast Ethernet 0/0 to next-hop IP address 192.168.0.11.

For a multipoint interface such as point-to-multipoint Frame Relay or Ethernet, when 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 1-30 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 1-30 *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 1-31 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 uses the PVC associated with DLCI 406 to get there.

Example 1-31 *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 1-32 provides sample output from the **show ip nhrp** command. This command displays the Next Hop Resolution Protocol cache that is used with DMVPN networks. In this example, if a packet needs to be sent to the 192.168.255.2 next-hop IP address, the nonbroadcast multiaccess (NBMA) address 198.51.100.2 is used to reach it.

Example 1-32 *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 1-33 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 405. Notice the value CA1B01C4001CCA1C164000540800 for Fast Ethernet 3/0. This is the destination MAC address, the source MAC address, and the EtherType code for an Ethernet frame. The first 12 hex values are the destination MAC address, the next 12 are the source MAC address, and 0800 is the IPv4 EtherType code.

Example 1-33 *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
```

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, and BGP, to name a few. With all these different options, you need to be able to recognize what is most 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 for a troubleshooter to understand how the best source of routing information is determined and how that source's information is placed in the routing table.

This section explains which sources of routing 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 the data structures of dynamic routing protocols interact with a router's IP routing table. Figure 1-15 shows the interaction between the data structures of an IP routing protocol and a router's IP routing table.

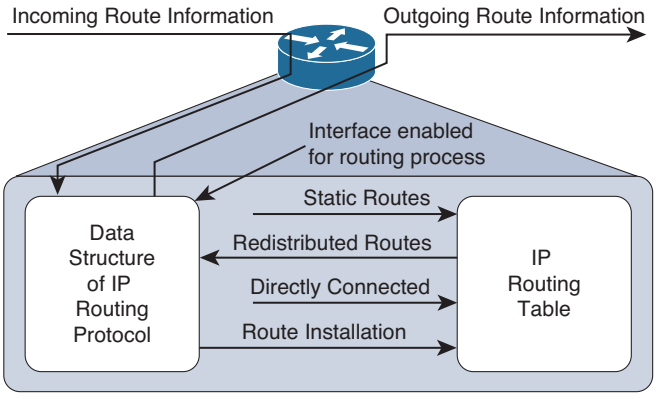

Figure 1-15 *Interaction Between the IP Routing Table and a Routing Protocol Data Structure*

As a router receives routing 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 routing information is redistributed from the routing table into the IP routing protocol's data structure. The router might be configured to have specific interfaces participate in an IP routing protocol process. In that case, 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 1-15 again, notice 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 Routing Information

A router could conceivably receive routing information from the following routing sources all at the same time:

- Connected interface
- Static route
- RIP
- EIGRP
- OSPF
- BGP

If the routing information received from all these sources is for different destination networks, each one is used for its respectively learned destination networks and placed in the routing table. However, what if the route received from Routing Information Protocol (RIP) and OSPF is exactly the same? For example, say that 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 must pick one and install that information in the routing table.

Routing information sources are each assigned an *administrative distance* (AD). Think of an administrative distance of a routing information source as the *believability* or *trustworthiness* of that routing source when comparing it to the other routing information sources. Table 1-4 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 (for example, 10.1.1.0/24), the OSPF route is 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 injected into the routing table only if the router 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.

Key Topic

Table 1-4 Default Administrative Distance of Route Sources

Source of Routing information	AD
Connected interface	0
Static route	1
EIGRP summary route	5
eBGP (External Border Gateway Protocol)	20
EIGRP (internal)	90
OSPF	110
IS-IS (Intermediate System to Intermediate System)	115
RIP	120
ODR (On-Demand Routing)	160
EIGRP (external)	170
iBGP (Internal Border Gateway Protocol)	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 1-34. Notice 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.

Key Topic

Example 1-34 *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 routing information or subset of routing information received from a particular source is never used, change the AD of specific routes or all routes from that source to 255, which means "do not believe."

AD is also used to manipulate path selection. For example, you might 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 suboptimal routing. But you are not allowed to remove the static route. To solve this issue, create a floating static route. This static route has a higher AD than the preferred route. Because you want EIGRP to be preferred, 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 is installed only if the EIGRP-learned route goes away.

Static Routes

Static routes are manually configured by administrators, and by default they 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. The following snippet displays the configuration of a static route on R1. The static route is training R1 about the 10.1.3.0/24 network:

```
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
```

The network is reachable via the next-hop address 10.1.12.2, which is R2, and is assigned an AD of 8. (The default is 1.)

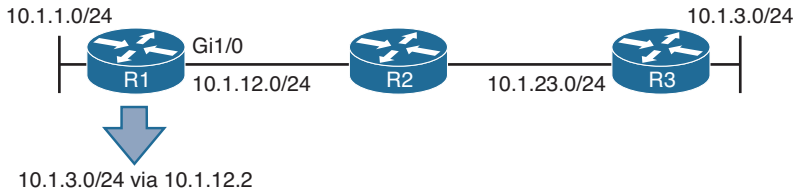

Figure 1-16 *Configuring a Static Route on R1 with the Next-Hop Option*

Example 1-35, 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 10.1.12.2, it has an AD of 8, and the metric is 0 because there is no way to know how far away the destination truly is (as there is with a dynamic routing protocol).

Example 1-35 *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 them 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 using 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 static route **ip route 10.1.3.0 255.255.255.0 10.1.12.1** on R2 in Figure 1-16, packets destined to 10.1.3.0 would be sent to R1, which is the wrong way. However, notice in Example 1-35 that R1 points to R2 (10.1.12.2) for the network 10.1.3.0/24. Therefore, R1 and R2 simply bounce packets that are destined for 10.1.3.0/24 back and forth until the TTL expires.

Key Topic

Notice that 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 1-35, 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 1-36. This example displays the output of the **show ip route 10.1.12.2** command on R1. Notice that 10.1.12.2 is directly connected out Gigabit Ethernet 1/0.

Example 1-36 *Recursive Lookup on R1 for the 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 source and destination MAC addresses. As a result, R1 looks in its ARP cache, as shown in Example 1-37, and finds that the MAC address for 10.1.12.2 is ca08.0568.0008.

Example 1-37 *MAC Address Lookup in the ARP Cache*

```
R1# show ip arp
Protocol Address     Age (min) Hardware Addr   Type Interface
Internet 10.1.1.1       -      ca07.0568.0008  ARPA GigabitEthernet0/0
Internet 10.1.12.1      -      ca07.0568.001c  ARPA GigabitEthernet1/0
Internet 10.1.12.2      71     ca08.0568.0008  ARPA GigabitEthernet1/0
```

Notice in this case that 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 when using the ARP process, and then it can store the results in the ARP cache. Then, any packet that has to go to the next hop address 10.1.12.2 does not require an ARP request to be sent; it needs just a lookup in the ARP cache, which makes the overall routing process more efficient.

Now that you understand the next-hop IP address, there is another option you need to know about. As you saw earlier in the **ip route** syntax, you can specify an exit interface instead of a next-hop IP address. There is a right time to use the exit interface, and there is a wrong time to use it. 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 but are still multiaccess, and because they are Ethernet, they require source and destination MAC addresses. 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.

Say that you configure the following static route on R1: **ip route 10.1.3.0 255.255.255.0 gigabit Ethernet 1/0**. Example 1-38 shows how the static route appears in the routing table. It states that 10.1.3.0/24 is directly connected to Gigabit Ethernet 1/0. But is it? Refer to Figure 1-17 to know for sure. It is clear in Figure 1-17 that 10.1.3.0/24 is not directly connected. But because of the way the static route is configured, R1 thinks that it is directly connected.

Example 1-38 *Static Route with an 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
```

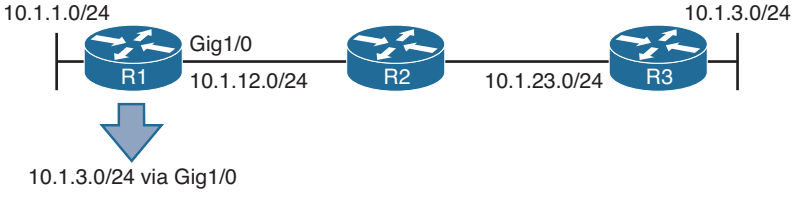

Figure 1-17 *Configuring a Static Route on R1 with Exit Interface Option*

Imagine that 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 finds that the longest match is the following entry:

```
S 10.1.3.0/24 is directly connected, GigabitEthernet1/0
```

R1 believes the network is directly connected; therefore, the destination IP address in the packet is on the network connected to Gig1/0. However, you know better because Figure 1-17 shows that it is not. So, because it is an Ethernet interface, R1 uses ARP to determine the MAC address 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 address of the next-hop address was used.) Example 1-39 shows the ARP cache on R1. Notice that every destination IP address has an entry in the ARP cache. How can that be if ARP requests are not forwarded by routers? It is because of proxy ARP, which is on by default on the 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 that 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.

Example 1-39 *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 1-40 shows how to use the **show ip interface** command to verify whether proxy ARP is enabled.

Example 1-40 *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 is not enabled, the ARP cache on R1 appears as shown in Example 1-41. Notice that 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 1-41 *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 uses 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. Specifying an Ethernet interface in a static route results in 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.

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 causes traffic to be misrouted or suboptimally routed. In addition, remember that static routes have an AD of 1; therefore, they are 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.

The following snippet displays the configuration of an IPv6 static route on R1, as shown in Figure 1-18:

```
R1# config t

R1(config)# ipv6 route 2001:DB8:0:3::/64 gigabitEthernet 1/0
FE80::2 8
```

The static route is training R1 about the 2001:DB8:0:3::/64 network. The network is reachable using the next-hop address FE80::2, which is R2's link-local address, and it was assigned an AD of 8. (The default is 1.) Notice that 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, 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.

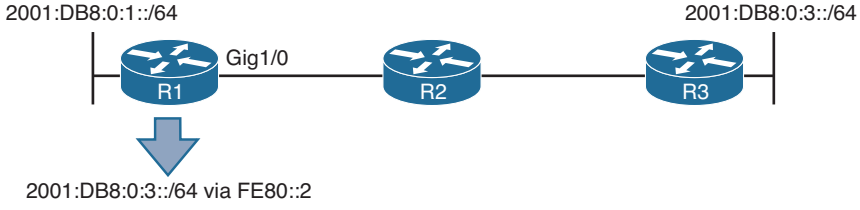

Figure 1-18 *Configuring an IPv6 Static Route on R1 with the Next-Hop Option*

Example 1-42, 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 FE80::2, it has an AD of 8, and the metric is 0 because there is no way to know how far away the destination truly is (as there is with a dynamic routing protocol).

Example 1-42 *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
```

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 address FE80::2, which is out Gig1/0. Therefore, it consults its IPv6 neighbor table, as shown in the following snippet, to determine whether there is a MAC address for FE80::2 out Gig 1/0:

```
R1# show ipv6 neighbors

IPv6 Address      Age      Link-layer Addr    State Interface

FE80::2            0       ca08.0568.0008     REACH Gi1/0
```

It is imperative that the table have an entry that maps 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 address FE80::2 on Gig1/0.

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 proxy ARP consumes 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 works 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 needs to be discovered using NDP. If the destination is not in the directly connected network, neighbor discovery fails, and Layer 2 encapsulation ultimately fails. Consider Figure 1-18 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
```

Key Topic

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 Figure 1-18). Therefore, R1 sends an Neighbor Solicitation (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 causes 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.

Trouble Tickets

This section presents various trouble tickets related to the topics discussed earlier in the chapter. The purpose of this section is to show you a process you can follow when troubleshooting in the real world or in an exam environment.

IPv4 Addressing and Addressing Technologies Trouble Tickets

Trouble Tickets 1-1 and 1-2 are based on the topology shown in Figure 1-19.

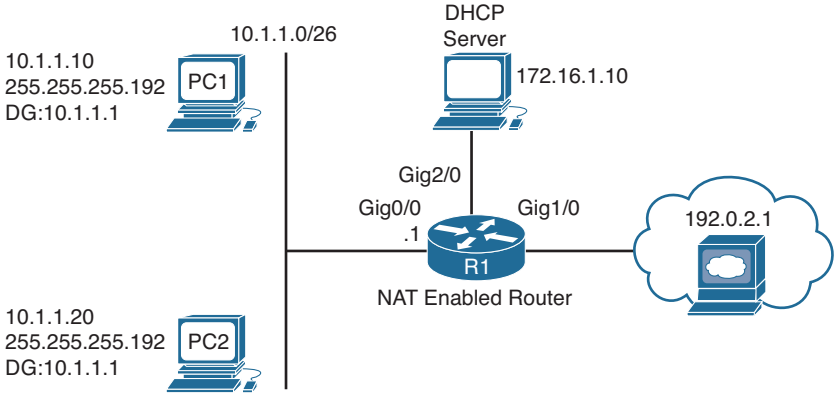

Figure 1-19 *IPv4 Addressing Trouble Tickets Topology*

Trouble Ticket 1-1

Problem: PC1 is not able to access resources on 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 1-43, the ping fails.

Example 1-43 *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 1-44, the ping is successful.

Example 1-44 *Successful Ping from PC1 to the 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 1-45.

Example 1-45 *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 have also confirmed that PC2 can reach Internet resources even though PC1 cannot. There are many reasons this situation might exist. One of the big ones is that an access control list (ACL) on Gig0/0 or Gig1/0 is denying PC1 from accessing resources on the Internet. Alternatively, a NAT issue could be preventing 10.1.1.10 from being translated. However, before you 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. If you review the output of **ipconfig** on PC1, as shown in Example 1-46, you see that the default gateway is configured as 10.1.1.100, which is not the IP address of R1's interface.

Example 1-46 *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 1-47.

Example 1-47 *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 1-2

Problem: PC1 is not able to access resources on 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 1-48, the ping fails.

Example 1-48 *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 1-49, it fails as well.

Example 1-49 *Failed Ping from PC1 to the 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 1-50, both pings fail as well, indicating that the problem is not isolated.

Example 1-50 *Failed Ping from PC2 to 192.0.2.1 and the 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. This can be caused by many different factors. For example, VLANs, VLAN access control lists (VACLs), trunks, VLAN Trunking Protocol (VTP), and Spanning Tree Protocol (STP) could all possibly cause this issue to occur. However, always remember to check the basics first; start with IP addressing on the client. On PC1, you issue the **ipconfig** command, and as shown in Example 1-51, 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 VLAN, trunk, VTP, STP, and so on as causes. However, it helps you narrow the focus.

Example 1-51 *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 . . . . . . . . .:
```

Notice in the trouble ticket topology in Figure 1-19 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. You can start there to eliminate this as the issue and then focus elsewhere if need be. On R1, you issue the command **show run interface gigabitEthernet 0/0**, as shown in Example 1-52. The output indicates that the IP helper address is 172.16.1.100, which is not correct according to the network diagram.

Example 1-52 *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 1-53.

Example 1-53 *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 PC1, the ping to 192.0.2.1 is successful, as shown in Example 1-54.

Example 1-54 *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
```

IPv6 Addressing Trouble Tickets

Trouble Tickets 1-3 and 1-4 are based on the topology shown in Figure 1-20.

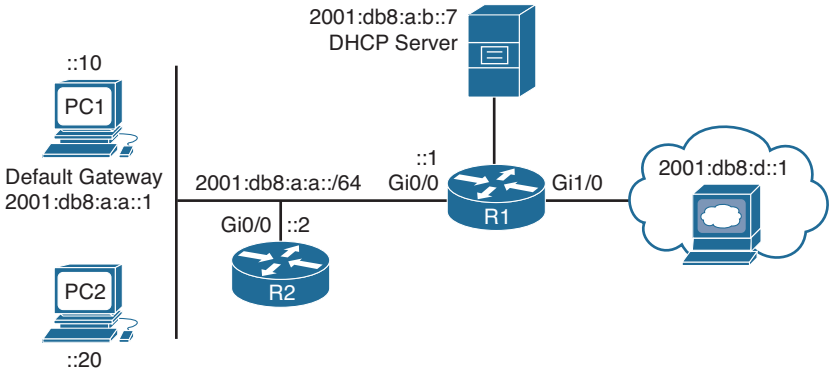

Figure 1-20 *IPv6 Addressing Trouble Tickets Topology*

Trouble Ticket 1-3

Problem: PC1 is not able to access resources on the web server 2001:db8:d::1.

Your network uses stateless address 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 1-55, the ping fails.

Example 1-55 *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 1-56.

Example 1-56 *Failed Ping from PC1 to the 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 by using the **ipconfig** command. Example 1-57 indicates that PC1 is not generating its own global unicast address using stateless address autoconfiguration or identifying a default gateway on the network.

Example 1-57 *Verifying IPv6 Addressing on PC1*

```
C:\PC1>ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

 Connection-specific DNS Suffix .  : cisco.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 1-58, and notice that it is also not generating an IPv6 address or identifying a default gateway.

Example 1-58 *Verifying IPv6 Addressing on PC2*

```
C:\PC2>ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

 Connection-specific DNS Suffix .  : cisco.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 gigabitEthernet 0/0** on R1, as shown in Example 1-59. The output indicates that hosts use SLAAC for addresses, and DHCP is used for other configuration values. However, it also indicates that RAs are suppressed. Therefore, PC1 and PC2 do not receive RAs that provide the prefix information necessary to perform autoconfiguration.

Example 1-59 *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):
 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 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 gigabitEthernet 0/0** to verify the configuration commands on the interface. As shown in Example 1-60, the interface is configured with the command **ipv6 nd ra suppress all**, which stops R1 from sending RAs.

Example 1-60 *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 1-61.

Example 1-61 *Verifying IPv6 Addressing on PC1*

```
C:\PC1>ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

 Connection-specific DNS Suffix .  : cisco.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, as shown in Example 1-62, 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 can.

Example 1-62 *Successful Ping from PC1 to the 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 1-4

Problem: PC1 is not able to access resources on the web server 2001:db8:d::1.

Your network uses stateless address 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 1-63, the ping fails.

Example 1-63 *Failed Ping from PC1 to the 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 1-64.

Example 1-64 *Failed Ping from PC1 to the 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 by using the **ipconfig** command. Example 1-65 indicates that PC1 is not generating its own global unicast address using stateless address autoconfiguration; however, it is identifying a default gateway on the network at the link-local address fe80::c80a:eff:fe3c:8.

Example 1-65 *Verifying IPv6 Addressing on PC1*

```
C:\PC1>ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

 Connection-specific DNS Suffix .  : cisco.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 1-66, and notice that it's experiencing the same issues as PC1.

Example 1-66 *Verifying IPv6 Addressing on PC2*

```
C:\PC2>ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

 Connection-specific DNS Suffix .  : cisco.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 must send RAs on the link for PC1 and PC2 to generate their own IPv6 address using SLAAC. You issue the command **show ipv6 interface gigabitEthernet 0/0** on R1, as shown in Example 1-67. The output indicates that hosts use SLAAC for addresses, and DHCP is 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 1-65 and 1-66, the default gateway is fe80::c80a:eff:fe3c:8. Based on Example 1-67, this is correct. If you review Example 1-67 further, can you see the issue?

Example 1-67 *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 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.
 Hosts use DHCP to obtain other configuration.
```

If you did not spot it, look at the global prefix assigned to interface Gig0/0. It is 2001:db8:a::/60. SLAAC works only if the prefix is /64.

You issue the command **show run interface gigabitEthernet 0/0** to verify the configuration commands on the interface. As shown in Example 1-68, the interface is configured with the command **ipv6 address 2001:db8:a:a::1/60**. RAs are still generated, but SLAAC does not work unless the prefix is /64.

Example 1-68 *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
```

You confirm with your network design plans that the prefix should be /64. 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 1-69.

Example 1-69 *Verifying IPv6 Addressing on PC1*

```
C:\PC1>ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

 Connection-specific DNS Suffix .  : cisco.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, as shown in Example 1-70, and the ping is successful. In addition, you contact the user at PC2, and she indicates that everything is fine now.

Example 1-70 *Successful Ping from PC1 to the 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
```

Static Routing Trouble Tickets

Trouble Tickets 1-5 and 1-6 are based on the topology shown in Figure 1-21.

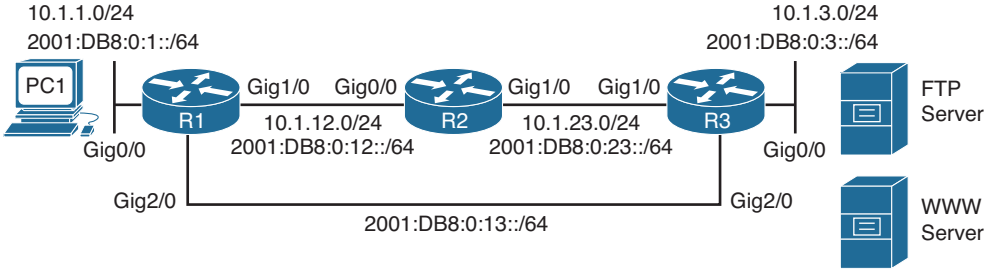

Figure 1-21 *Static Routing Trouble Tickets Topology*

Trouble Ticket 1-5

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 the static IPv4 address 10.1.3.10. Users have also indicated that they are able to access the web server at 10.1.3.5. (Note that this network uses only 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 1-71, 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 1-71 as well.

Example 1-71 *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 1-72, the closest entry that matches 10.1.3.10 is the entry for 10.1.3.0/29. However, does 10.1.3.10 fall within that subnet?

Example 1-72 *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 has a range of addresses from 10.1.3.0 to 10.1.3.7, and 10.1.3.10 does not fall within that subnet; however, 10.1.3.5 does fall within that range. 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 verifies this further. As shown in Example 1-73, there is no match for 10.1.3.10, but there is a match for 10.1.3.5.

Example 1-73 *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 1-21 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 the following snippet:

```
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
```

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.

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 1-74.

Example 1-74 *Verifying an 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 1-75, and see that the IP address 10.1.3.10 now matches an entry in the routing table.

Example 1-75 *Verifying That 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 1-76.

Example 1-76 *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 1-6

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 be used to forward traffic from 2001:DB8:0:1::/64 to 2001:DB8:0:3::/64 only 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 that this network uses only static routes.)

You confirm the problem with a trace, as shown in Example 1-77, 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 1-77 *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:DB8:0:12::2
3 5 ms 1 ms 2 ms 2001:DB8:0:23::3

Trace complete.
```

Next, you issue the **show ipv6 route 2001:DB8:0:3::/64** command on R1, as shown in Example 1-78, 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 R2's Gig0/0 interface. The next-hop IPv6 address should be 2001:DB8:0:13::3, which is R3's Gig2/0 interface.

Example 1-78 *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 by using the **show run | include ipv6 route** command, as shown in Example 1-79. You notice that there are two commands for 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 1-79 *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 1-80, you can see 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 is therefore the one used.

To solve this issue, you need to configure the static route with a 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 1-80 confirms that the change was successful.

Example 1-80 *Verifying the 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
```

Next, you perform a trace from PC1 to 2001:DB8:0:3::3, as shown in Example 1-81, and it confirms that R2 is no longer being used. The traffic is now flowing across the link between R1 and R3.

Example 1-81 *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.
```

Exam Preparation Tasks

As mentioned in the section "How to Use This Book" in the Introduction, you have a couple choices for exam preparation: the exercises here, Chapter 24, "Final Preparation," and the exam simulation questions in the Pearson Test Prep software. The questions that follow present a bigger challenge than the exam itself because they use an open-ended question format. By using this more difficult format, you can exercise your memory better and prove your conceptual and factual knowledge of this chapter. You can find the answers to these questions in the appendix.

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 these key topics and the page number on which each is found.

Table 1-5 Key Topics for Chapter 1

Key Topic Element	Description	Page Number
Paragraph	The process used by a device to determine whether the packet will be sent to a local or remote device	7
Paragraph	What occurs when IPv4 addressing is not correct	9
Example 1-1	Verifying IP Addressing on a PC and on a Router	10
Section	Determining IP Addresses Within a Subnet	10
Step list	The DHCPv4 DORA process	12
Example 1-3	DHCP relay agent configuration	13
Snippet	DHCP client configuration	15
Paragraph	How a router can be configured as a DHCP server	15

Key Topic Element	Description	Page Number
List	Items to look out for while troubleshooting DHCP-related issues	16
Section	DHCP troubleshooting commands	17
Paragraphs	The process used by a device to determine whether the packet will be sent to a local or remote device when using IPv6	19
Paragraph	The EUI-64 process	20
Example 1-12	Verifying EUI-64 on a router interface	22
Example 1-14	Enabling SLAAC on a router interface	23
Paragraph	The router advertisement process	23
Paragraph	Verifying SLAAC-generated IPv6 addresses	24
List	Issues that may occur while using SLAAC	24
Example 1-17	Verifying that an interface is enabled for IPv6	25
Example 1-18	Verifying that RAs are not suppressed	25
Example 1-19	Verifying default gateways configured on a PC	26
Example 1-21	Sample DHCPv6 configuration on R1	27
Example 1-22	Verifying DHCPv6 information on R1	27
Example 1-23	Verifying stateless DHCPv6	28
Step list	The four-way negotiation process of DHCPv6	29
Example 1-24	Configuring R1 as a DHCPv6 relay agent	30
List	The routing table and Layer 3-to-Layer 2 mapping table	33
List	The FIB and adjacency table	34
Example 1-25	**show ip route** *ip_address* command output	34
Example 1-28	**show ip cef** *ip_address* command output	36
Example 1-30	**show ip arp** command output	36
Table 1-4	Administrative distance of route sources	40
Example 1-34	Verifying the administrative distance of a route in the routing table	40
Paragraph	The importance of the next-hop address in an IPv4 static route	42
Paragraph	Using an Ethernet interface in an IPv4 static route	44
Paragraph	Using an Ethernet interface in an IPv6 static route	47

Define Key Terms

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

DHCP, DORA, DHCPDISCOVER, DHCPOFFER, DHCPREQUEST, DHCPACK, DHCP relay agent, APIPA, Neighbor Discovery, EUI-64, stateless address 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, packet forwarding, ARP, TTL, routing table, ARP cache, CEF, FIB, adjacency table, control plane, data plane, administrative distance, static route, proxy ARP

Command Reference to Check Your Memory

This section includes the most important configuration and verification 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 1-6 with a piece of paper, read the description on the left side, and then see how much of the command you can remember.

The ENARSI 300-410 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, verify, and troubleshoot the topics covered in this chapter.

Table 1-6 Configuration and Verification Commands

Task	Command Syntax
Display the IP address, subnet mask, and default gateway of a Windows PC	**ipconfig**
Display the IP address, subnet mask, and default gateway of a Windows PC, in addition to DNS servers, domain name, MAC address, and whether autoconfiguration is enabled	**ipconfig /all**
Display various IP-related parameters for a router interface, including the IP address and subnet mask that have been assigned	**show ip interface** *interface_type interface_number*
Identify 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**
Display IP addresses that an IOS DHCP server assigns, their corresponding MAC addresses, and lease expirations	**show ip dhcp binding**
Determine whether IPv6 is enabled on an interface, display the multicast groups the router interface is a member of, display the global and link-local unicast addresses associated with an interface, indicate whether EUI-64 was used or stateless autoconfiguration was used to obtain the IPv6 address for the interface, display whether RAs are suppressed for the interface, and display 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** *interface_type interface_number*
Display the IPv6 addresses that are being used by each of the DHCPv6 clients	**show ipv6 dhcp binding**

Task	Command Syntax
Display which DHCPv6 pool is assigned to which interface on the router	**show ipv6 dhcp interface**
Display the configured DHCPv6 pools on the router	**show ipv6 dhcp pool**
Display a router's best route to the specified IP address	**show ip route** *ip_address*
Display only the static routes in a router's routing table	**show ip route static**
Display 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** *ip_address subnet_mask*
Display all routes in a router's IP routing table that are encompassed by the specified network address and subnet mask (This command is often useful when troubleshooting route summarization issues.)	**show ip route** *ip_address subnet_mask* **longer- prefixes**
Display information (for example, next-hop IP address and egress interface) required to forward a packet, similar to the output of the **show ip route** *ip_address* command (The output of this command comes from CEF. Therefore, routing protocol information is not presented in the output.)	**show ip cef** *ip_address*
Display 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** *ip_address subnet_mask*
Display 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** *source_address destination_address*
Display the static IPv6 routes configured on a device	**show ipv6 route static**
Display the Layer 3 IPv6 address-to-Layer 2 MAC address mappings	**show ipv6 neighbors**
Display a router's ARP cache, containing IPv4 address-to-MAC address mappings	**show ip arp**

CHAPTER 2

EIGRP

This chapter covers the following topics:

- **EIGRP Fundamentals:** This section explains how EIGRP establishes a neighborship with other routers and how routes are exchanged with other routers.
- **EIGRP Configuration Modes:** This section defines the two methods of configuring EIGRP with a baseline configuration.
- **Path Metric Calculation:** This section explains how EIGRP calculates the path metric to identify the best and alternate loop-free paths.

Enhanced Interior Gateway Routing Protocol (*EIGRP*) is an enhanced distance vector routing protocol commonly found in enterprise networks. EIGRP is a derivative of Interior Gateway Routing Protocol (IGRP) but includes support for variable-length subnet masking (VLSM) and metrics capable of supporting higher-speed interfaces. Initially, EIGRP was a Cisco proprietary protocol, but it was released to the Internet Engineering Task Force (IETF) through RFC 7868, which was ratified in May 2016.

This chapter explains the underlying mechanics of the EIGRP routing protocol and the path metric calculations, and it demonstrates how to configure EIGRP on a router. This is the first of several chapters in the book that discuss EIGRP:

- **Chapter 2, "EIGRP":** This chapter describes the fundamental concepts of EIGRP.
- **Chapter 3, "Advanced EIGRP":** This chapter describes EIGRP's failure detection mechanisms and techniques to optimize the operations of the routing protocol. It also includes topics such as route filtering and traffic manipulation.
- **Chapter 4, "Troubleshooting EIGRP for IPv4":** This chapter reviews common problems with the routing protocols and the methodology to troubleshoot EIGRP from an IPv4 perspective.
- **Chapter 5, "EIGRPv6":** This chapter demonstrates how IPv4 EIGRP concepts carry over to IPv6 and the methods to troubleshoot common problems.

"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?' Quiz Questions."

Table 2-1 "Do I Know This Already?" Foundation Topics Section-to-Question Mapping

Foundation Topics Section	Questions
EIGRP Fundamentals	1–6
EIGRP Configuration Modes	7–9
Path Metric Calculation	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 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. EIGRP uses protocol number ____ for inter-router communication.
   - **a.** 87
   - **b.** 88
   - **c.** 89
   - **d.** 90
2. How many packet types does EIGRP use for inter-router communication?
   - **a.** Three
   - **b.** Four
   - **c.** Five
   - **d.** Six
   - **e.** Seven
3. Which of the following is not required to match to form an EIGRP adjacency?
   - **a.** Metric K values
   - **b.** Primary subnet
   - **c.** Hello and hold timers
   - **d.** Authentication parameters
4. What is an EIGRP successor?
   - **a.** The next-hop router for the path with the lowest path metric for a destination prefix
   - **b.** The path with the lowest metric for a destination prefix
   - **c.** The router selected to maintain the EIGRP adjacencies for a broadcast network
   - **d.** A route that satisfies the feasibility condition where the reported distance is less than the feasible distance

5. What attributes does the EIGRP topology table contain? (Choose all that apply.)
   a. Destination network prefix
   b. Hop Count
   c. Total path delay
   d. Maximum path bandwidth
   e. List of EIGRP neighbors
6. What destination addresses does EIGRP use when feasible? (Choose two.)
   a. IP address 224.0.0.9
   b. IP address 224.0.0.10
   c. IP address 224.0.0.8
   d. MAC address 01:00:5E:00:00:0A
   e. MAC address 0C:15:C0:00:00:01
7. The EIGRP process is initialized by which of the following technique? (Choose two.)
   a. Using the interface command **ip eigrp** *as-number* **ipv4 unicast**
   b. Using the global configuration command **router eigrp** *as-number*
   c. Using the global configuration command **router eigrp** *process-name*
   d. Using the interface command **router eigrp** *as-number*
8. True or false: The EIGRP router ID (RID) must be configured for EIGRP to be able to establish neighborship.
   a. True
   b. False
9. True or false: When using MD5 authentication between EIGRP routers, the key-chain sequence number can be different, as long as the password is the same.
   a. True
   b. False
10. Which value can be modified on a router to manipulate the path taken by EIGRP but does not have impacts on other routing protocols, like OSPF?
    a. Interface bandwidth
    b. Interface MTU
    c. Interface delay
    d. Interface priority

Foundation Topics

EIGRP Fundamentals

EIGRP overcomes the deficiencies of other distance vector routing protocols, such as Routing Information Protocol (RIP), with features such as unequal-cost load balancing, support for networks 255 hops away, and rapid convergence features. EIGRP uses a *diffusing update algorithm* (*DUAL*) to identify network paths and provides for fast convergence using precalculated loop-free backup paths. Most distance vector routing protocols use hop count as the metric for routing decisions. Using hop count for path selection does not take into account link speed and total delay. EIGRP adds logic to the route-selection algorithm that uses factors besides hop count.

Autonomous Systems

A router can run multiple EIGRP processes. Each process operates under the context of an autonomous system, which represents a common routing domain. Routers within the same domain use the same metric calculation formula and exchange routes only with members of the same autonomous system. Do not confuse an EIGRP autonomous system with a Border Gateway Protocol (BGP) autonomous system.

In Figure 2-1, EIGRP autonomous system (AS) 100 consists of R1, R2, R3, R4, and EIGRP AS 200 consists of R3, R5, and R6. Each EIGRP process correlates to a specific autonomous system and maintains an independent EIGRP topology table. R1 does not have knowledge of routes from AS 200 because it is different from its own autonomous system, AS 100. R3 is able to participate in both autonomous systems and, by default, does not transfer routes learned from one autonomous system into a different autonomous system.

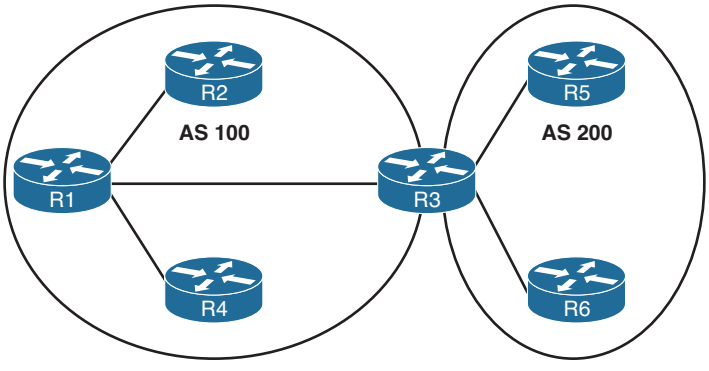

Figure 2-1 *EIGRP Autonomous Systems*

EIGRP uses *protocol-dependent modules* (*PDMs*) to support multiple network protocols, such as IPv4, IPv6, AppleTalk, and IPX. EIGRP is written so that the PDM is responsible for the functions to handle the route selection criteria for each communication protocol. In theory, new PDMs can be written as new communication protocols are created. Current implementations of EIGRP support only IPv4 and IPv6.

EIGRP Terminology

This section explains some of the core concepts of EIGRP, along with the path selection process. Figure 2-2 is used as a reference topology for R1 calculating the best path and alternative loop-free paths to the 10.4.4.0/24 network. The values in parentheses represent the link's calculated metric for a segment based on bandwidth and delay.

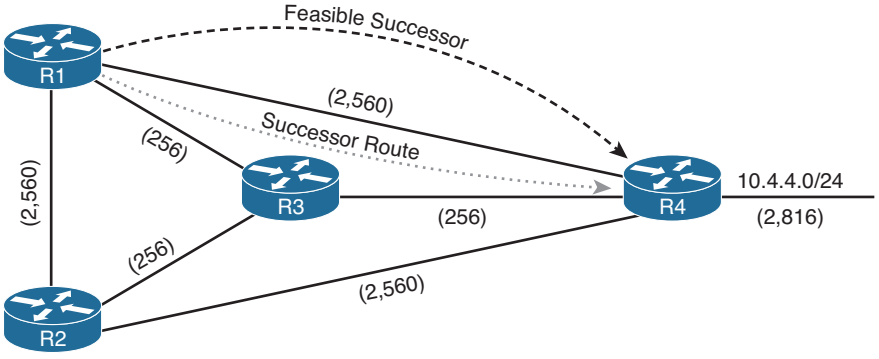

Figure 2-2 *EIGRP Reference Topology*

Table 2-2 defines important terms related to EIGRP and correlates them to Figure 2-2.

Key Topic

Table 2-2 EIGRP Terminology

Term	Definition
Successor route	The route with the lowest path metric to reach a destination. The successor route for R1 to reach 10.4.4.0/24 on R4 is R1→R3→R4.
Successor	The first next-hop router for the successor route. The successor for 10.4.4.0/24 is R3.
Feasible distance (FD)	The metric value for the lowest-metric path to reach a destination. The feasible distance is calculated locally using the formula shown in the "Path Metric Calculation" section, later in this chapter. The FD calculated by R1 for the 10.4.4.0/24 network is 3328 (that is, 256 + 256 + 2816).
Reported distance (RD)	Distance reported by a router to reach a prefix. The reported distance value is the feasible distance for the advertising router. R3 advertises the 10.4.4.0/24 prefix with an RD of 3072. R4 advertises the 10.4.4.0/24 to R1 and R2 with an RD of 2816.
Feasibility condition	For a route to be considered a backup route, the RD received for that route must be less than the FD calculated locally. This logic guarantees a loop-free path.
Feasible successor	A route with that satisfies the feasibility condition is maintained as a backup route. The feasibility condition ensures that the backup route is loop free. The route R1→R4 is the feasible successor because the RD of 2816 is lower than the FD of 3328 for the R1→R3→R4 path.

Topology Table

Key Topic

EIGRP contains a topology table, which makes it different from a true distance vector routing protocol. EIGRP's topology table is a vital component of DUAL and contains information to identify loop-free backup routes. The topology table contains all the network prefixes advertised within an EIGRP autonomous system. Each entry in the table contains the following:

- Network prefix
- EIGRP neighbors that have advertised that prefix
- Metrics from each neighbor (reported distance and hop count)
- Values used for calculating the metric (load, reliability, total delay, and minimum bandwidth)

The command **show ip eigrp topology** [**all-links**] provides the topology table. By default, only the successor and feasible successor routes are displayed, but the optional **all-links** keyword shows the paths that did not pass the feasibility condition.

Figure 2-3 shows the topology table for R1 from Figure 2-2. This section focuses on the 10.4.4.0/24 network when explaining the topology table.

```
R1#show ip eigrp topology
EIGRP-IPv4 Topology Table for AS (100)/ID(192.168.1.1)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 10.12.1.0/24, 1 successors, FD is 2816
        via Connected, GigabitEthernet0/3
P 10.13.1.0/24, 1 successors, FD is 2816
        via Connected, GigabitEthernet0/1
P 10.14.1.0/24, 1 successors, FD is 5120
        via Connected, GigabitEthernet0/2
P 10.23.1.0/24, 2 successors, FD is 3072
        via 10.12.1.2 (3072/2816), GigabitEthernet0/3
        via 10.13.1.3 (3072/2816), GigabitEthernet0/1
P 10.34.1.0/24, 1 successors, FD is 3072
        via 10.13.1.3 (3072/2816), GigabitEthernet0/1
        via 10.14.1.4 (5376/2816), GigabitEthernet0/2
P 10.24.1.0/24, 1 successors, FD is 5376
        via 10.12.1.2 (5376/5120), GigabitEthernet0/3
        via 10.14.1.4 (7680/5120), GigabitEthernet0/2
P 10.4.4.0/24, 1 successors, FD is 3328
        via 10.13.1.3 (3328/3072), GigabitEthernet0/1
        via 10.14.1.4 (5376/2816), GigabitEthernet0/2
```

Feasible Distance

Successor Route

Path Metric

Reported Distance

Feasible Successor

Passes Feasibility Condition
2816<3328

Figure 2-3 *EIGRP Topology Output*

Examine the network 10.4.4.0/24 and notice that R1 calculates an FD of 3328 for the successor route. The successor (upstream router) advertises the successor route with an RD of 3072. The second path entry has a metric of 5376 and has an RD of 2816. Because 2816 is less than 3072, the second entry passes the feasibility condition and classifies the second entry as the feasible successor for the prefix.

The 10.4.4.0/24 route is passive (P), which means the topology is stable. During a topology change, routes go into an active (A) state when computing a new path.

EIGRP Neighbors

EIGRP does not rely on periodic advertisement of all the network prefixes in an autonomous system, which is done with routing protocols such as Routing Information Protocol (RIP), Open Shortest Path First (OSPF), and Intermediate System-to-Intermediate System (IS-IS). EIGRP neighbors exchange the entire routing table when forming an adjacency, and they advertise incremental updates only as topology changes occur within a network. The neighbor adjacency table is vital for tracking neighbor status and the updates sent to each neighbor.

Inter-Router Communication

EIGRP uses five different packet types to communicate with other routers, as shown in Table 2-3. EIGRP uses its own IP protocol number (88) and uses multicast packets where possible; it uses unicast packets when necessary. Communication between routers is done with multicast using the group address 224.0.0.10 or the MAC address 01:00:5e:00:00:0a when possible.

Key Topic

Table 2-3 EIGRP Packet Types

Packet Type	Packet Name	Function
1	Hello	Used for discovery of EIGRP neighbors and for detecting when a neighbor is no longer available
2	Request	Used to get specific information from one or more neighbors
3	Update	Used to transmit routing and reachability information with other EIGRP neighbors
4	Query	Sent out to search for another path during convergence
5	Reply	Sent in response to a query packet

NOTE EIGRP uses multicast packets to reduce bandwidth consumed on a link (one packet to reach multiple devices). While broadcast packets are used in the same general way, all nodes on a network segment process broadcast packets, whereas with multicast, only nodes listening for the particular multicast group process the multicast packets.

EIGRP uses *Reliable Transport Protocol* (*RTP*) to ensure that packets are delivered in order and to ensure that routers receive specific packets. A sequence number is included in each EIGRP packet. The sequence value zero does not require a response from the receiving EIGRP router; all other values require an ACK packet that includes the original sequence number.

Ensuring that packets are received makes the transport method reliable. All update, query, and reply packets are deemed reliable, and hello and ACK packets do not require acknowledgment and could be unreliable.

If the originating router does not receive an ACK packet from the neighbor before the retransmit timeout expires, it notifies the non-acknowledging router to stop processing its multicast packets. The originating router sends all traffic by unicast until the neighbor is fully synchronized. Upon complete synchronization, the originating router notifies the destination router to start processing multicast packets again. All unicast packets require acknowledgment. EIGRP retries up to 16 times for each packet that requires confirmation, and it resets the neighbor relationship when the neighbor reaches the retry limit of 16.

NOTE In the context of EIGRP, do not confuse RTP with the Real-Time Transport Protocol (RTP), which is used for carrying audio or video over an IP network. EIGRP's RTP allows for confirmation of packets while supporting multicast. Other protocols that require reliable connection-oriented communication, such as TCP, cannot use multicast addressing.

Forming EIGRP Neighbors

Unlike other distance vector routing protocols, EIGRP requires a neighbor relationship to form before routes are processed and added to the Routing Information Base (RIB). Upon hearing an EIGRP hello packet, a router attempts to become the neighbor of the other router. The following parameters must match for the two routers to become neighbors:

- Metric formula K values
- Primary subnet matches
- Autonomous system number (ASN) matches
- Authentication parameters

Figure 2-4 shows the process EIGRP uses for forming neighbor adjacencies.

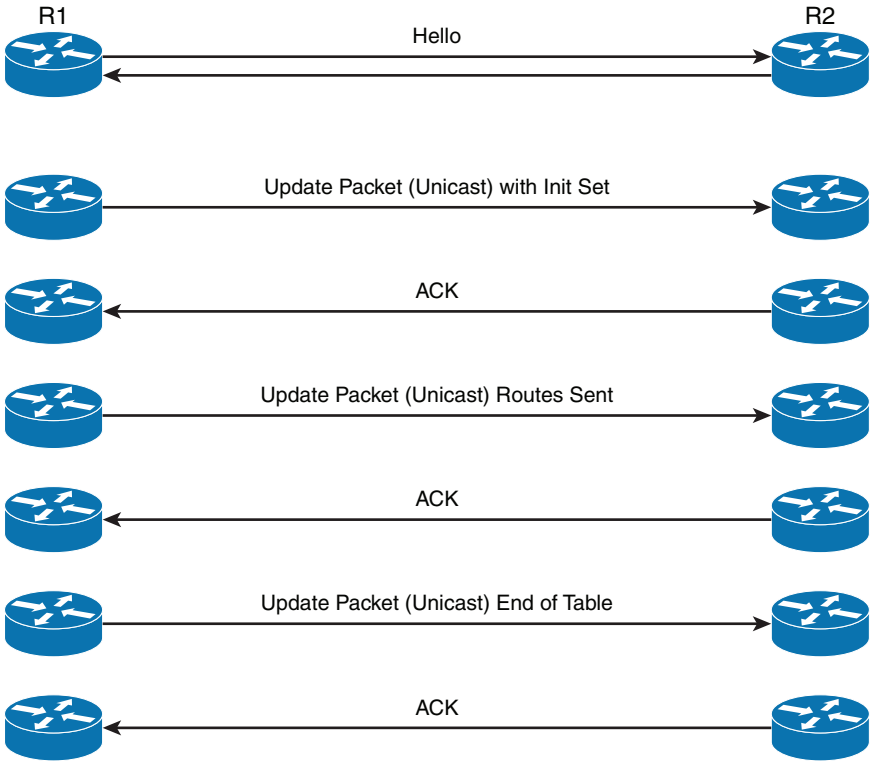

Figure 2-4 *EIGRP Neighbor Adjacency Process from R1's Perspective*

EIGRP Configuration Modes

This section describes the two methods of EIGRP configuration: classic mode and named mode.

Classic Configuration Mode

With classic EIGRP configuration mode, most of the configuration takes place in the EIGRP process, but some settings are configured under the interface configuration submode. This can add complexity for deployment and troubleshooting as users must scroll back and forth between the EIGRP process and individual network interfaces. Some of the settings set individually are hello advertisement interval, split-horizon, authentication, and summary route advertisements.

Key Topic

Classic configuration requires the initialization of the routing process with the global configuration command **router eigrp** *as-number* to identify the ASN and initialize the EIGRP process. The second step is to identify the network interfaces with the command **network** *ip-address* [*mask*]. The network statement is explained in the following sections.

EIGRP Named Mode

Key Topic

EIGRP named mode configuration was released to overcome some of the difficulties network engineers have with classic EIGRP autonomous system configuration, including scattered configurations and unclear scope of commands.

EIGRP named configuration provides the following benefits:

- All the EIGRP configuration occurs in one location.
- It supports current EIGRP features and future developments.
- It supports multiple address families (including Virtual Routing and Forwarding [VRF] instances). EIGRP named configuration is also known as *multi-address family configuration mode*.
- Commands are clear in terms of the scope of their configuration.

EIGRP named mode provides a hierarchical configuration and stores settings in three subsections:

- **Address Family:** This submode contains settings that are relevant to the global EIGRP AS operations, such as selection of network interfaces, EIGRP K values, logging settings, and stub settings.
- **Interface:** This submode contains settings that are relevant to the interface, such as hello advertisement interval, split-horizon, authentication, and summary route advertisements. In actuality, there are two methods of the EIGRP interface section's configuration. Commands can be assigned to a specific interface or to a *default* interface, in which case those settings are placed on all EIGRP-enabled interfaces. If there is a conflict between the default interface and a specific interface, the specific interface takes priority over the default interface.
- **Topology:** This submode contains settings regarding the EIGRP topology database and how routes are presented to the router's RIB. This section also contains route redistribution and administrative distance settings.

EIGRP named configuration makes it possible to run multiple instances under the same EIGRP process. The process for enabling EIGRP interfaces on a specific instance is as follows:

Step 1. Initialize the EIGRP process by using the command **router eigrp** *process-name*. (If a number is used for *process-name*, the *number* does not correlate to the autonomous system number.)

Step 2. Initialize the EIGRP instance for the appropriate address family with the command **address-family {IPv4 | IPv6} {unicast | vrf** *vrf-name*} *autonomous-system as-number*.

Step 3. Enable EIGRP on interfaces by using the command **network** *network mask.*

EIGRP Network Statement

Both configuration modes use a network statement to identify the interfaces that EIGRP will use. The network statement uses a wildcard mask, which allows the configuration to be as specific or ambiguous as necessary.

NOTE The two styles of EIGRP configuration are independent. Using the configuration options from classic EIGRP autonomous system configuration does not modify settings on a router running EIGRP named configuration.

The syntax for the network statement, which exists under the EIGRP process, is **network** *ip-address* [*mask*]. The optional *mask* can be omitted to enable interfaces that fall within the classful boundaries for that network statement.

A common misconception is that the **network** statement adds the networks to the EIGRP topology table. In reality, the **network** statement identifies the interface to enable EIGRP on, and it adds the interface's connected network to the EIGRP topology table. EIGRP then advertises the topology table to other routers in the EIGRP autonomous system.

EIGRP does not add an interface's secondary connected network to the topology table. For secondary connected networks to be installed in the EIGRP routing table, they must be redistributed into the EIGRP process. Chapter 16, "Route Redistribution," provides additional coverage of route redistribution.

To help illustrate the concept of the wildcard mask, Table 2-4 provides a set of IP addresses and interfaces for a router. The following examples provide configurations to match specific scenarios.

Table 2-4 Table of Sample Interface and IP Addresses

Router Interface	IP Address
Gigabit Ethernet 0/0	10.0.0.10/24
Gigabit Ethernet 0/1	10.0.10.10/24
Gigabit Ethernet 0/2	192.0.0.10/24
Gigabit Ethernet 0/3	192.10.0.10/24

The configuration in Example 2-1 enables EIGRP only on interfaces that explicitly match the IP addresses in Table 2-4.

Example 2-1 *EIGRP Configuration with Explicit IP Addresses*

```
Router eigrp 1
    network 10.0.0.10 0.0.0.0
    network 10.0.10.10 0.0.0.0
    network 192.0.0.10 0.0.0.0
    network 192.10.0.10 0.0.0.0
```

Example 2-2 shows the EIGRP configuration using **network** statements that match the subnets used in Table 2-4. Setting the last octet of the IP address to 0 and changing the wildcard mask to 255 causes the network statements to match all IP addresses within the /24 network range.

Example 2-2 *EIGRP Configuration with Explicit Subnet*

```
Router eigrp  1
    network 10.0.0.0 0.0.0.255
    network 10.0.10.0 0.0.0.255
    network 192.0.0.0 0.0.0.255
    network 192.10.0.0 0.0.0.255
```

The following snippet shows the EIGRP configuration using **network** statements for interfaces that are within the 10.0.0.0/8 or 192.0.0.0/8 network ranges:

```
router eigrp  1
     network 10.0.0.0 0.255.255.255
     network 192.0.0.0 0.255.255.255
```

The following snippet shows the configuration to enable all interfaces with EIGRP:

```
router eigrp  1
     network 0.0.0.0 255.255.255.255
```

NOTE A key topic with wildcard network statements is that large ranges simplify configuration; however, they may possibly enable EIGRP on unintended interfaces.

Sample Topology and Configuration

Figure 2-5 shows a sample topology for demonstrating EIGRP configuration in classic mode for R1 and named mode for R2.

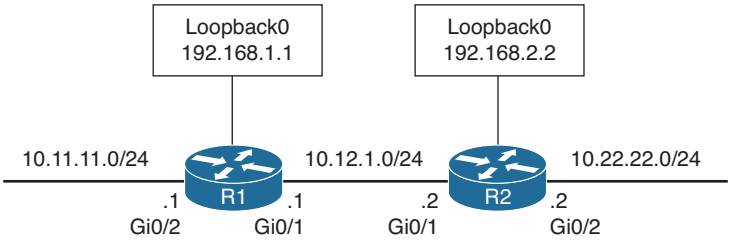

Figure 2-5 *EIGRP Sample Topology*

R1 and R2 enable EIGRP on all of their interfaces. R1 configures EIGRP using multiple specific network interface addresses, and R2 enables EIGRP on all network interfaces with one command. Example 2-3 provides the configuration that is applied to R1 and R2.

Example 2-3 *Sample EIGRP Configuration*

```
R1 (Classic Configuration)
interface Loopback0
 ip address 192.168.1.1 255.255.255.255
!
interface GigabitEthernet0/1
    ip address 10.12.1.1 255.255.255.0
!
interface GigabitEthernet0/2
    ip address 10.11.11.1 255.255.255.0
!
router eigrp 100
 network 10.11.11.1 0.0.0.0
 network 10.12.1.1 0.0.0.0
 network 192.168.1.1 0.0.0.0
```

```
R2 (Named Mode Configuration)
interface Loopback0
 ip address 192.168.2.2 255.255.255.255
!
interface GigabitEthernet0/1
    ip address 10.12.1.2 255.255.255.0
!
interface GigabitEthernet0/2
    ip address 10.22.22.2 255.255.255.0
!
router eigrp EIGRP-NAMED
 address-family ipv4 unicast autonomous-system 100
  network 0.0.0.0 255.255.255.255
```

As mentioned earlier, EIGRP named mode has three configuration submodes. The configuration from Example 2-3 uses only the EIGRP address-family submode section, which uses the **network** statement. The EIGRP topology base submode is created automatically with the command **topology base** and exited with the command **exit-af-topology**. Settings for the topology submode are listed between those two commands.

Example 2-4 demonstrates the slight difference in how the configuration is stored on the router between EIGRP classic and named mode configurations.

Example 2-4 *Named Mode Configuration Structure*

```
R1# show run | section router eigrp
router eigrp 100
 network 10.11.11.1 0.0.0.0
 network 10.12.1.1 0.0.0.0
 network 192.168.1.1 0.0.0.0
```

```
R2# show run | section router eigrp
router eigrp EIGRP-NAMED
 !
 address-family ipv4 unicast autonomous-system 100
  !
  topology base
  exit-af-topology
  network 0.0.0.0
 exit-address-family
```

NOTE The EIGRP interface submode configurations contain the command **af-interface** *interface-id* or **af-interface default** with any specific commands listed immediately. The EIGRP interface submode configuration is exited with the command **exit-af-interface**. This is demonstrated later in this chapter.

Confirming Interfaces

Upon configuring EIGRP, it is a good practice to verify that only the intended interfaces are running EIGRP. The command **show ip eigrp interfaces** [{*interface-id* [**detail**] | **detail**}] shows active EIGRP interfaces. Appending the optional **detail** keyword provides additional information, such as authentication, EIGRP timers, split horizon, and various packet counts.

Example 2-5 demonstrates R1's non-detailed EIGRP interface and R2's detailed information for the Gi0/1 interface.

Example 2-5 *Verification of EIGRP Interfaces*

```
R1# 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
Gi0/2       0       0/0        0/0          0       0/0           0           0
Gi0/1       1       0/0        0/0         10       0/0          50           0
Lo0         0       0/0        0/0          0       0/0           0           0
```

```
R2# show ip eigrp interfaces gi0/1 detail
```

```
EIGRP-IPv4 VR(EIGRP-NAMED) 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
Gi0/1        1        0/0       0/0        1583       0/0       7912           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: 186/2
  Un/reliable mcasts: 0/2  Un/reliable ucasts: 2/2
  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 not set
  Topologies advertised on this interface:  base
  Topologies not advertised on this interface:
```

Table 2-5 provides a brief explanation to the key fields shown with the EIGRP interfaces.

Table 2-5 EIGRP Interface Fields

Field	Description
Interface	Interfaces running EIGRP.
Peers	Number of peers detected on that interface.
Xmt Queue Un/Reliable	Number of unreliable/reliable packets remaining in the transmit queue. The value zero is an indication of a stable network.
Mean SRTT	Average time for a packet to be sent to a neighbor and a reply from that neighbor to be received, in milliseconds.
Multicast Flow Timer	Maximum time (seconds) that the router sent multicast packets.
Pending Routes	Number of routes in the transmit queue that need to be sent.

Verifying EIGRP Neighbor Adjacencies

Each EIGRP process maintains a table of neighbors to ensure that they are alive and processing updates properly. Without keeping track of a neighbor state, an autonomous system could contain incorrect data and could potentially route traffic improperly. EIGRP must form a neighbor relationship before a router advertises update packets containing network prefixes.

The command **show ip eigrp neighbors** [*interface-id*] displays the EIGRP neighbors for a router. Example 2-6 shows the EIGRP neighbor information using this command.

Example 2-6 *EIGRP Neighbor Confirmation*

```
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   10.12.1.2               Gi0/1                    13 00:18:31   10   100  0  3
```

Table 2-6 provides a brief explanation of the key fields shown in Example 2-6.

Table 2-6 EIGRP Neighbor Columns

Field	Description
Address	IP address of the EIGRP neighbor
Interface	Interface the neighbor was detected on
Holdtime	Time left to receive a packet from this neighbor to ensure that it is still alive
SRTT	Time for a packet to be sent to a neighbor and a reply to be received from that neighbor, in milliseconds
RTO	Timeout for retransmission (waiting for ACK)
Q Cnt	Number of packets (update/query/reply) in queue for sending
Seq Num	Sequence number that was last received from this router

Displaying Installed EIGRP Routes

You can see EIGRP routes that are installed into the RIB by using the command **show ip route eigrp**. EIGRP routes originating within the autonomous system have an administrative distance (AD) of 90 and are indicated in the routing table with a D. Routes that originate from outside the autonomous system are external EIGRP routes. External EIGRP routes have an AD of 170 and are indicated in the routing table with D EX. Placing external EIGRP routes into the RIB with a higher AD acts as a loop-prevention mechanism.

Example 2-7 displays the EIGRP routes from the sample topology in Figure 2-5. The metric for the selected route is the second number in brackets.

Example 2-7 *EIGRP Routes for R1 and R2*

```
R1# 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
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR
```

```
Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
D        10.22.22.0/24 [90/3072] via 10.12.1.2, 00:19:25, GigabitEthernet0/1
      192.168.2.0/32 is subnetted, 1 subnets
D        192.168.2.2 [90/2848] via 10.12.1.2, 00:19:25, GigabitEthernet0/1
```

```
R2# show ip route eigrp
! Output omitted for brevity
Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
D        10.11.11.0/24 [90/15360] via 10.12.1.1, 00:20:34, GigabitEthernet0/1
      192.168.1.0/32 is subnetted, 1 subnets
D        192.168.1.1 [90/2570240] via 10.12.1.1, 00:20:34, GigabitEthernet0/1
```

NOTE The metrics for R2's routes are different from the metrics from R1's routes. This is because R1's classic EIGRP mode uses classic metrics, and R2's named mode uses wide metrics by default. This topic is explained in depth in the "Path Metric Calculation" section, later in this chapter.

Router ID

The router ID (RID) is a 32-bit number that uniquely identifies an EIGRP router and is used as a loop-prevention mechanism. The RID can be set dynamically, which is the default, or manually.

The algorithm for dynamically choosing the EIGRP RID uses the highest IPv4 address of any *up* loopback interfaces. If there are not any *up* loopback interfaces, the highest IPv4 address of any active *up* physical interfaces becomes the RID when the EIGRP process initializes.

IPv4 addresses are commonly used for the RID because they are 32 bits and are maintained in dotted-decimal format. You use the command **eigrp router-id** *router-id* to set the RID, as demonstrated in Example 2-8, for both classic and named mode configurations.

Example 2-8 *Static Configuration of EIGRP Router ID*

```
R1(config)# router eigrp 100
R1(config-router)# eigrp router-id 192.168.1.1
```

```
R2(config)# router eigrp EIGRP-NAMED
R2(config-router)# address-family ipv4 unicast autonomous-system 100
R2(config-router-af)# eigrp router-id 192.168.2.2
```

Key Topic

Passive Interfaces

Some network topologies must advertise a network segment into EIGRP but need to prevent neighbors from forming adjacencies with other routers on that segment. This might be the case, for example, when advertising access layer networks in a campus topology. In such a scenario, you need to put the EIGRP interface in a passive state. Passive EIGRP interfaces do not send out or process EIGRP hellos, which prevents EIGRP from forming adjacencies on that interface.

To configure an EIGRP interface as passive, you use the command **passive-interface** *interface-id* under the EIGRP process for classic configuration. Another option is to configure all interfaces as passive by default with the command **passive-interface default** and then use the command **no passive-interface** *interface-id* to allow an interface to process EIGRP packets, preempting the global passive interface default configuration.

Example 2-9 demonstrates making R1's Gi0/2 interface passive and also the alternative option of making all interfaces passive but setting Gi0/1 as non-passive.

Example 2-9 *Passive EIGRP Interfaces for Classic Configuration*

```
R1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# router eigrp 100
R1(config-router)# passive-interface gi0/2
```

```
R1(config)# router eigrp 100
R1(config-router)# passive-interface default
04:22:52.031: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 10.12.1.2 (GigabitEther-
net0/1) is down: interface passive
R1(config-router)# no passive-interface gi0/1
*May 10 04:22:56.179: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 10.12.1.2 (Giga-
bitEthernet0/1) is up: new adjacency
```

For a named mode configuration, you place the **passive-interface** state on **af-interface default** for all EIGRP interfaces or on a specific interface with the **af-interface** *interface-id* section. Example 2-10 shows how to set the Gi0/2 interface as passive while allowing the Gi0/1 interface to be active using both configuration strategies.

Example 2-10 *Passive EIGRP Interfaces for Named Mode Configuration*

```
R2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)# router eigrp EIGRP-NAMED
R2(config-router)# address-family ipv4 unicast autonomous-system 100
R2(config-router-af)# af-interface gi0/2
R2(config-router-af-interface)# passive-interface
R2(config-router-af-interface)# exit-af-interface
```

```
R2(config)# router eigrp EIGRP-NAMED
R2(config-router)# address-family ipv4 unicast autonomous-system 100
R2(config-router-af)# af-interface default
R2(config-router-af-interface)# passive-interface
04:28:30.366: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 10.12.1.1
(GigabitEthernet0/1) is down: interface passiveex
R2(config-router-af-interface)# exit-af-interface
R2(config-router-af)# af-interface gi0/1
R2(config-router-af-interface)# no passive-interface
R2(config-router-af-interface)# exit-af-interface
*May 10 04:28:40.219: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 10.12.1.1
(GigabitEthernet0/1) is up: new adjacency
```

Example 2-11 shows what the named mode configuration looks like with some settings (i.e. **passive-interface** or **no passive-interface**) placed under the **af-interface default** or the **af-interface** *interface-id* setting.

Example 2-11 *Viewing the EIGRP Interface Settings with Named Mode*

```
R2# show run | section router eigrp
router eigrp EIGRP-NAMED
 !
 address-family ipv4 unicast autonomous-system 100
  !
  af-interface default
   passive-interface
  exit-af-interface
  !
  af-interface GigabitEthernet0/1
   no passive-interface
  exit-af-interface
  !
  topology base
  exit-af-topology
  network 0.0.0.0
 exit-address-family
```

A passive interface does not appear in the output of the command **show ip eigrp interfaces** even though it was enabled. Connected networks for passive interfaces are still added to the EIGRP topology table so that they are advertised to neighbors.

Example 2-12 shows that the Gi0/2 interface on R1 no longer appears; compare this to Example 2-5, where it does exist.

Example 2-12 *Passive Interfaces do not Appear*

```
R1# 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
Gi0/1       1        0/0       0/0          9       0/0           50          0
```

To accelerate troubleshooting of passive interfaces, and other settings, the command **show ip protocols** provides a lot of valuable information about all the routing protocols. With EIGRP, it displays the EIGRP process identifier, the ASN, K values that are used for path calculation, RID, neighbors, AD settings, and all the passive interfaces.

Example 2-13 provides sample output for both classic and named mode instances on R1 and R2.

Example 2-13 *IP Protocols Output*

```
R1# show ip protocols
! Output omitted for brevity
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
    Soft SIA disabled
    NSF-aware route hold timer is 240
    Router-ID: 192.168.1.1
    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:
    10.11.11.1/32
    10.12.1.1/32
    192.168.1.1/32
  Passive Interface(s):
    GigabitEthernet0/2
    Loopback0
```

```
  Routing Information Sources:
    Gateway         Distance      Last Update
    10.12.1.2              90      00:21:35
  Distance: internal 90 external 170
```

```
R2# show ip protocols
! Output omitted for brevity
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 VR(EIGRP-NAMED) 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
    Soft SIA disabled
    NSF-aware route hold timer is 240
    Router-ID: 192.168.2.2
    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

  Automatic Summarization: disabled
  Maximum path: 4
  Routing for Networks:
    0.0.0.0
  Passive Interface(s):
    GigabitEthernet0/2
    Loopback0
  Routing Information Sources:
    Gateway         Distance      Last Update
    10.12.1.1              90      00:24:26
  Distance: internal 90 external 170
```

Authentication

Authentication is a mechanism for ensuring that only authorized routers are eligible to become EIGRP neighbors. It is possible for someone to add a router to a network and introduce invalid routes accidentally or maliciously. Authentication prevents such scenarios from happening. A precomputed password hash is included with all EIGRP packets, and the receiving router decrypts the hash. If the passwords do not match for a packet, the router discards the packet.

EIGRP encrypts the password by using a Message Digest 5 (MD5) authentication, using the keychain function. The hash consists of the key number and a password. EIGRP authentication encrypts just the password rather than the entire EIGRP packet.

NOTE Keychain functionality allows a password to be valid for a specific time, so passwords can change at preconfigured times. Restricting the key sequence to a specific time is beyond the scope of this book. For more information, see Cisco.com.

To configure EIGRP authentication, you need to create a keychain and then enable EIGRP authentication on the interface. The following sections explain the steps.

Keychain Configuration

Keychain creation is accomplished with the following steps:

Step 1. Create the keychain by using the command **key chain** *key-chain-name*.

Step 2. Identify the key sequence by using the command **key** *key-number*, where *key-number* can be anything from 0 to 2147483647.

Step 3. Specify the preshared password by using the command **key-string** *password*.

NOTE Be careful not to use a space after the password because that will be used for computing the hash.

Enabling Authentication on the Interface

When using classic configuration, authentication must be enabled on the interface under the interface configuration submode. The following commands are used in the interface configuration submode:

```
ip authentication key-chain eigrp as-number key-chain-name
```

```
ip authentication mode eigrp as-number md5
```

The named mode configuration places the configurations under the EIGRP interface submode, under the **af-interface default** or the **af-interface** *interface-id*. Named mode configuration supports MD5 or *Hashed Message Authentication Code-Secure Hash*

Algorithm-256 (*HMAC-SHA-256*) authentication. MD5 authentication involves the following commands:

authentication key-chain eigrp *key-chain-name*

authentication mode md5

The HMAC-SHA-256 authentication involves the command **authentication mode hmac-sha-256** *password*.

Example 2-14 demonstrates MD5 configuration on R1 with classic EIGRP configuration and on R2 with named mode configuration. Remember that the hash is computed using the key sequence number and key string, which must match on the two nodes.

Example 2-14 *EIGRP Authentication Configuration*

```
R1(config)# key chain EIGRPKEY
R1(config-keychain)# key 2
R1(config-keychain-key)# key-string CISCO
R1(config)# interface gi0/1
R1(config-if)# ip authentication mode eigrp 100 md5
R1(config-if)# ip authentication key-chain eigrp 100 EIGRPKEY
```

```
R2(config)# key chain EIGRPKEY
R2(config-keychain)# key 2
R2(config-keychain-key)# key-string CISCO
R2(config-keychain-key)# router eigrp EIGRP-NAMED
R2(config-router)# address-family ipv4 unicast autonomous-system 100
R2(config-router-af)# af-interface default
R2(config-router-af-interface)# authentication mode md5
R2(config-router-af-interface)# authentication key-chain EIGRPKEY
```

The command **show key chain** provides verification of the keychain. Example 2-15 shows that each key sequence provides the lifetime and password.

Example 2-15 *Verification of Keychain Settings*

```
R1# show key chain
Key-chain EIGRPKEY:
    key 2 -- text "CISCO"
        accept lifetime (always valid) - (always valid) [valid now]
        send lifetime (always valid) - (always valid) [valid now]
```

The EIGRP interface detail view provides verification of EIGRP authentication on a specific interface. Example 2-16 provides detailed EIGRP interface output.

Example 2-16 *Verification of EIGRP Authentication*

```
R1# show ip eigrp interface detail
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
Gi0/1             0        0/0          0/0           0        0/0           50
0
  Hello-interval is 5, Hold-time is 15
  Split-horizon is enabled
  Next xmit serial <none>
  Packetized sent/expedited: 10/1
  Hello's sent/expedited: 673/12
  Un/reliable mcasts: 0/9  Un/reliable ucasts: 6/19
  Mcast exceptions: 0  CR packets: 0  ACKs suppressed: 0
  Retransmissions sent: 16  Out-of-sequence rcvd: 1
  Topology-ids on interface - 0
  Authentication mode is md5,  key-chain is "EIGRPKEY"
```

Key Topic

Path Metric Calculation

Metric calculation is a critical component for any routing protocol. EIGRP uses multiple factors to calculate the metric for a path. Metric calculation uses *bandwidth* and *delay* by default but can include interface load and reliability, too. The formula shown in Figure 2-6 illustrates the EIGRP classic metric formula.

$$\text{Metric} = \left[\left(K_1 * BW + \frac{K_2 * BW}{256 - Load} + K_3 * Delay\right) * \frac{K_5}{K_4 + Reliability}\right]$$

Figure 2-6 *EIGRP Classic Metric Formula*

EIGRP uses K values to define which factors the formula uses and the impact associated with a factor when calculating the metric. A common misconception is that the K values directly apply to bandwidth, load, delay, or reliability; this is not accurate. For example, K_1 and K_2 both reference bandwidth (BW).

BW represents the slowest link in the path, scaled to a 10 Gbps link (10^7). Link speed is collected from the configured interface bandwidth on an interface. Delay is the total measure of delay in the path, measured in tens of microseconds (µs).

The EIGRP formula is based on the IGRP metric formula, except the output is multiplied by 256 to change the metric from 24 bits to 32 bits. Taking these definitions into consideration, the formula for EIGRP is shown in Figure 2-7.

$$\text{Metric} = 256 * \left[\left(K_1 * \frac{10^7}{\text{Min. Bandwidth}} + \frac{K_2 * \frac{10^7}{\text{Min. Bandwidth}}}{256 - Load} + \frac{K_3 * \text{Total Delay}}{10}\right) * \frac{K_5}{K_4 + Reliability}\right]$$

Figure 2-7 *EIGRP Classic Metric Formula with Definitions*

By default, K_1 and K_3 have a value of 1, and K_2, K_4, and K_5 are set to 0. Figure 2-8 places default K values into the formula and shows a streamlined version of the formula.

$$\text{Metric} = 256 * \left[\left(1 * \frac{10^7}{\text{Min. Bandwidth}} + \frac{\frac{10^7}{0 * \text{Min. Bandwidth}}}{256 - \text{Load}} + \frac{1 * \text{Total Delay}}{10}\right) * \frac{0}{0 + \text{Reliability}}\right]$$

Equals

$$\text{Metric} = 256 * \left(\frac{10^7}{\text{Min. Bandwidth}} + \frac{\text{Total Delay}}{10}\right)$$

Figure 2-8 *EIGRP Classic Metric Formula with Default K Values*

Key Topic

The EIGRP update packet includes path attributes associated with each prefix. The EIGRP path attributes can include hop count, cumulative delay, minimum bandwidth link speed, and RD. The attributes are updated each hop along the way, allowing each router to independently identify the shortest path.

Figure 2-9 shows the information in the EIGRP update packets for the 10.1.1.0/24 prefix propagating through the autonomous system. Notice that the hop count increments, minimum bandwidth decreases, total delay increases, and the RD changes with each EIGRP update.

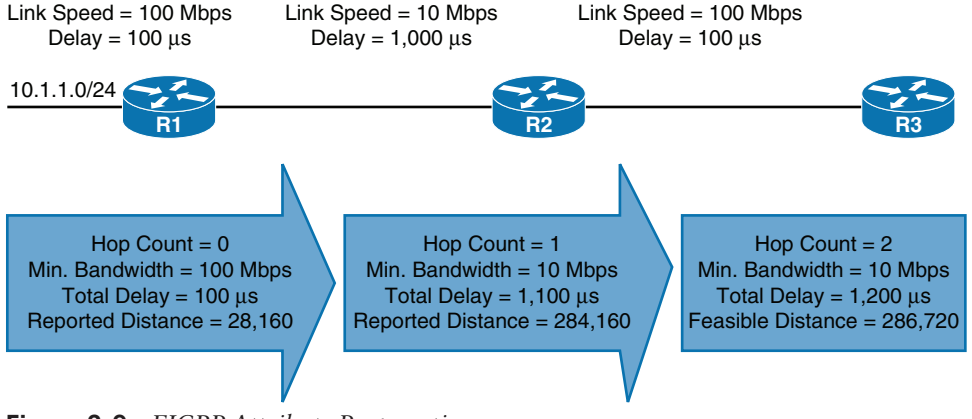

Figure 2-9 *EIGRP Attribute Propagation*

Table 2-7 shows some of the common network types, link speeds, delay, and EIGRP metric, using the streamlined formula from Figure 2-7.

Table 2-7 Default EIGRP Interface Metrics for Classic Metrics

Interface Type	Link Speed (Kbps)	Delay	Metric
Serial	64	20,000 µs	40,512,000
T1	1544	20,000 µs	2,170,031
Ethernet	10,000	1000 µs	281,600
FastEthernet	100,000	100 µs	28,160
GigabitEthernet	1,000,000	10 µs	2816
TenGigabitEthernet	10,000,000	10 µs	512

Using the topology from Figure 2-2, the metrics from R1 and R2 for the 10.4.4.0/24 network are calculated using the formula in Figure 2-10. The link speed for both routers is 1 Gbps, and the total delay is 30 µs (10 µs for the 10.4.4.0/24 link, 10 µs for the 10.34.1.0/24 link, and 10 µs for the 10.13.1.0/24 link).

$$\text{Metric} = 256 * \left(\frac{10^7}{1{,}000{,}000} + \frac{30}{10}\right) = 3{,}328$$

Figure 2-10 *EIGRP Classic Metric Formula with Default K Values*

If you are unsure of the EIGRP metrics, you can query the parameters for the formula directly from EIGRP's topology table by using the command **show ip eigrp topology** *network/prefix-length.*

Example 2-17 shows R1's topology table output for the 10.4.4.0/24 network. Notice that the output includes the successor route, any feasible successor paths, and the EIGRP state for the prefix. Each path contains the EIGRP attributes minimum bandwidth, total delay, interface reliability, load, and hop count.

Example 2-17 *EIGRP Topology for a Specific Prefix*

```
R1# show ip eigrp topology 10.4.4.0/24
! Output omitted for brevity
EIGRP-IPv4 Topology Entry for AS(100)/ID(10.14.1.1) for 10.4.4.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 3328
  Descriptor Blocks:
  10.13.1.3 (GigabitEthernet0/1), from 10.13.1.3, Send flag is 0x0
      Composite metric is (3328/3072), route is Internal
      Vector metric:
        Minimum bandwidth is 1000000 Kbit
        Total delay is 30 microseconds
        Reliability is 252/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 2
        Originating router is 10.34.1.4
  10.14.1.4 (GigabitEthernet0/2), from 10.14.1.4, Send flag is 0x0
      Composite metric is (5376/2816), route is Internal
```

```
    Vector metric:
       Minimum bandwidth is 1000000 Kbit
       Total delay is 110 microseconds
       Reliability is 255/255
       Load is 1/255
       Minimum MTU is 1500
       Hop count is 1
       Originating router is 10.34.1.4
```

Wide Metrics

The original EIGRP specifications measured delay in 10-microsecond (µs) units and bandwidth in kilobytes per second, which did not scale well with higher-speed interfaces. In Table 2-7, notice that the delay is the same for the GigabitEthernet and TenGigabitEthernet interfaces.

Example 2-18 provides some metric calculations for common LAN interface speeds. Notice that there is not a differentiation between an 11 Gbps interface and a 20 Gbps interface. The composite metric stays at 256, despite the different bandwidth rates.

Example 2-18 *Metric Calculation for Common LAN Interface Speeds*

```
GigabitEthernet:
Scaled Bandwidth = 10,000,000 / 1,000,000
Scaled Delay = 10 / 10
Composite Metric = 10 + 1 * 256 = 2816
```

```
10 GigabitEthernet:
Scaled Bandwidth = 10,000,000 / 10,000,000
Scaled Delay = 10 / 10
Composite Metric = 1 + 1 * 256 = 512
```

```
11 GigabitEthernet:
Scaled Bandwidth = 10,000,000 / 11,000,000
Scaled Delay = 10 / 10
Composite Metric = 0 + 1 * 256 = 256
```

```
20 GigabitEthernet:
Scaled Bandwidth = 10,000,000 / 20,000,000
Scaled Delay = 10 / 10
Composite Metric = 0 + 1 * 256 = 256
```

EIGRP includes support for a second set of metrics, known as *wide metrics*, that addresses the issue of scalability with higher-capacity interfaces. The original formula referenced in Figure 2-6 is known as *EIGRP classic metrics*.

Figure 2-11 shows the explicit EIGRP wide metrics formula. Notice that an additional K value (K_6) is included that adds an extended attribute to measure jitter, energy, or other future attributes.

Key Topic

$$\text{Wide Metric} = \left[\left(K_1 * BW + \frac{K_2 * BW}{256 - Load} + K_3 * Latency + K_6 * Extended\right) * \frac{K_5}{K_4 + Reliability}\right]$$

Figure 2-11 *EIGRP Wide Metrics Formula*

Just as EIGRP scaled by 256 to accommodate IGRP, EIGRP wide metrics scale by 65,535 to accommodate higher-speed links. This provides support for interface speeds up to 655 terabits per second ($65{,}535 \times 10^7$) without any scalability issues. Latency is the total interface delay measured in picoseconds (10^{-12}) instead of in microseconds (10^{-6}). Figure 2-12 shows an updated formula that takes into account the conversions in latency and scalability.

$$\text{Wide Metric} = 65{,}535 * \left[\left(\frac{K_1 * 10^7}{\text{Min. Bandwidth}} + \frac{\frac{K_2 * 10^7}{\text{Min. Bandwidth}}}{256 - Load} + \frac{K_3 * Latency}{10\text{^}6} + K_6 * Extended\right) * \frac{K_5}{K_4 + Reliability}\right]$$

Figure 2-12 *EIGRP Wide Metrics Formula with Definitions*

The interface delay varies from router to router, depending on the following logic:

- If the interface's delay was specifically set, the value is converted to picoseconds. Interface delay is always configured in tens of microseconds and is multiplied by 10^7 for picosecond conversion.
- If the interface's bandwidth was specifically set, the interface delay is configured using the classic default delay, converted to picoseconds. The configured bandwidth is not considered when determining the interface delay. If delay was configured, this step is ignored.
- If the interface supports speeds of 1 Gbps or less and does not contain bandwidth or delay configuration, the delay is the classic default delay, converted to picoseconds.
- If the interface supports speeds over 1 Gbps and does not contain bandwidth or delay configuration, the interface delay is calculated by 10^{13}/interface bandwidth.

The EIGRP classic metrics exist only with EIGRP classic configuration, while EIGRP wide metrics exist only in EIGRP named mode. The metric style used by a router is identified with the command **show ip protocols**; if a K_6 metric is present, the router is using wide-style metrics.

Example 2-19 verifies the operational mode of EIGRP on R1 and R2. R1 does not have a K_6 metric and is using EIGRP classic metrics. R2 has a K_6 metric and is using EIGRP wide metrics.

Example 2-19 *Verification of EIGRP Metric Style*

```
R1# show ip protocols | include AS|K
  EIGRP-IPv4 Protocol for AS(100)
    Metric weight K1=1, K2=0, K3=1, K4=0, K5=0
```

```
R2# show ip protocols | include AS|K
  EIGRP-IPv4 VR(EIGRP-NAMED) Address-Family Protocol for AS(100)
    Metric weight K1=1, K2=0, K3=1, K4=0, K5=0 K6=0
```

Metric Backward Compatibility

EIGRP wide metrics were designed with backward compatibility in mind. EIGRP wide metrics set K_1 and K_3 to a value of 1 and set K_2, K_4, K_5, and K_6 to 0, which allows backward compatibility because the K value metrics match with classic metrics. As long as K_1 through K_5 are the same and K_6 is not set, the two metric styles allow adjacency between routers.

EIGRP is able to detect when peering with a router is using classic metrics, and it *unscales* the metric to the formula in Figure 2-13.

$$\text{Unscaled Bandwidth} = \left(\frac{\text{EIGRP Bandwidth} * \text{EIGRP Classic Scale}}{\text{Scaled Bandwidth}} \right)$$

Figure 2-13 *Formula for Calculating Unscaled EIGRP Metrics*

This conversion results in loss of clarity if routes pass through a mixture of classic metric and wide metric devices. An end result of this intended behavior is that paths learned from wide metric peers always look better than paths learned from classic peers. Using a mixture of classic metric and wide metric devices could lead to suboptimal routing, so it is best to keep all devices operating with the same metric style.

Interface Delay Settings

If you do not remember the delay values from Table 2-7, the values can be dynamically queried with the command **show interface** *interface-id*. The output displays the EIGRP interface delay, in microseconds, after the DLY field. Example 2-20 provides sample output of the command on R1 and R2. Both interfaces have a delay of 10.

Example 2-20 *Verification of EIGRP Interface Delay*

```
R1# show interfaces gigabitEthernet 0/1 | i DLY
  MTU 1500 bytes, BW 1000000 Kbit/sec, DLY 10 usec,
```

```
R2# show interfaces gigabitEthernet 0/1 | i DLY
  MTU 1500 bytes, BW 1000000 Kbit/sec, DLY 10 usec,
```

EIGRP delay is set on an interface-by-interface basis, allowing for manipulation of traffic patterns flowing through a specific interface on a router. Delay is configured with the interface parameter command **delay** *tens-of-microseconds* under the interface.

Example 2-21 demonstrates the modification of the delay on R1 to 100, increasing the delay to 1000 µs on the link between R1 and R2. To ensure consistent routing, modify the delay on R2's Gi0/1 interface as well. Afterward, you can verify the change.

Example 2-21 *Interface Delay Configuration*

```
R1# configure terminal
R1(config)# interface gi0/1
R1(config-if)# delay 100
R1(config-if)# do show interface Gigabit0/1 | i DLY
  MTU 1500 bytes, BW 1000000 Kbit/sec, DLY 1000 usec,
```

NOTE Bandwidth modification with the interface parameter command **bandwidth** *bandwidth* has a similar effect on the metric calculation formula but can impact other routing protocols, such as OSPF, at the same time. Modifying the interface delay only impacts EIGRP.

Key Topic

Custom K Values

If the default metric calculations are insufficient, you can change them to modify the path metric formula. K values for the path metric formula are set with the command **metric weights** *TOS* K_1 K_2 K_3 K_4 K_5 [K_6] under the EIGRP process. The TOS value always has a value of 0, and the K_6 value is used for named mode configurations.

To ensure consistent routing logic in an EIGRP autonomous system, the K values must match between EIGRP neighbors to form an adjacency and exchange routes. The K values are included as part of the EIGRP hello packet. The K values are displayed with the **show ip protocols** command, as demonstrated with the sample topology in Example 2-13. Notice that both routers are using the default K values, with R1 using classic metrics and R2 using wide metrics.

Load Balancing

EIGRP allows multiple successor routes (with the same metric) to be installed into the RIB. Installing multiple paths into the RIB for the same prefix is called *equal-cost multipathing* (*ECMP*) routing. At the time of this writing, the default maximum ECMP is four routes. You change the default ECMP setting with the command **maximum-paths** *maximum-paths* under the EIGRP process in classic mode and under the topology base submode in named mode.

Example 2-22 shows the configuration for changing the maximum paths on R1 and R2 so that classic and named mode configurations are visible.

Example 2-22 *Changing the EIGRP Maximum Paths*

```
R1# show run | section router eigrp
router eigrp 100
 maximum-paths 6
 network 0.0.0.0
```

```
R2# show run | section router eigrp
router eigrp EIGRP-NAMED
 !
 address-family ipv4 unicast autonomous-system 100
  !
  topology base
   maximum-paths 6
  exit-af-topology
  network 0.0.0.0
  eigrp router-id 192.168.2.2
 exit-address-family
```

Key Topic

EIGRP supports unequal-cost load balancing, which allows installation of both successor routes and feasible successors into the EIGRP RIB. To use unequal-cost load balancing with EIGRP, change EIGRP's *variance multiplier*. The EIGRP *variance value* is the feasible distance (FD) for a route multiplied by the EIGRP variance multiplier. Any feasible successor's FD with a metric below the EIGRP variance value is installed into the RIB. EIGRP installs multiple routes where the FD for the routes is less than the EIGRP multiplier value up to the maximum number of ECMP routes, as discussed earlier.

Dividing the feasible successor metric by the successor route metric provides the variance multiplier. The variance multiplier is a whole number, and any remainders should always round up.

Using the topology shown in Figure 2-2 and output from the EIGRP topology table in Figure 2-3, the minimum EIGRP variance multiplier can be calculated so that the direct path from R1 to R4 can be installed into the RIB. The FD for the successor route is 3328, and the FD for the feasible successor is 5376. The formula provides a value of about 1.6 and is always rounded up to the nearest whole number to provide an EIGRP variance multiplier of 2. Figure 2-14 shows the calculation.

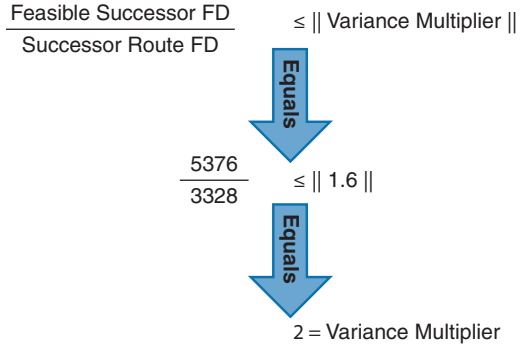

Figure 2-14 *EIGRP Variance Multiplier Formula*

The command **variance** *multiplier* configures the variance multiplier under the EIGRP process for classic configuration and under the topology base submode in named mode. Example 2-23 provides a sample configuration for both configuration modes.

Example 2-23 *EIGRP Variance Configuration*

```
R1 (Classic Configuration)
router eigrp 100
 variance 2
 network 0.0.0.0
```

```
R1 (Named Mode Configuration)
router eigrp EIGRP-NAMED
 !
 address-family ipv4 unicast autonomous-system 100
  !
  topology base
   variance 2
  exit-af-topology
  network 0.0.0.0
  exit-address-family
```

Example 2-24 provides a brief verification that both paths were installed into the RIB. Notice that the metrics for the paths are different. One path metric is 3328, and the other path metric is 5376. To see the traffic load-balancing ratios, you use the command **show ip route** *network*, as demonstrated in the second output. The load-balancing traffic share is highlighted.

Example 2-24 *Verification of Unequal-Cost Load Balancing*

```
R1# show ip route eigrp | begin Gateway
Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 10 subnets, 2 masks
D        10.4.4.0/24 [90/5376] via 10.14.1.4, 00:00:03, GigabitEthernet0/2
                     [90/3328] via 10.13.1.3, 00:00:03, GigabitEthernet0/1
```

```
R1# show ip route 10.4.4.0
Routing entry for 10.4.4.0/24
  Known via "eigrp 100", distance 90, metric 3328, type internal
  Redistributing via eigrp 100
  Last update from 10.13.1.3 on GigabitEthernet0/1, 00:00:35 ago
  Routing Descriptor Blocks:
  * 10.14.1.4, from 10.14.1.4, 00:00:35 ago, via GigabitEthernet0/2
      Route metric is 5376, traffic share count is 149
      Total delay is 110 microseconds, minimum bandwidth is 1000000 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 1
```

```
10.13.1.3, from 10.13.1.3, 00:00:35 ago, via GigabitEthernet0/1
  Route metric is 3328, traffic share count is 240
  Total delay is 30 microseconds, minimum bandwidth is 1000000 Kbit
  Reliability 254/255, minimum MTU 1500 bytes
  Loading 1/255, Hops 2
```

References in This Chapter

Edgeworth, Brad, Foss, Aaron, and Garza Rios, Ramiro. *IP Routing on Cisco IOS, IOS XE, and IOS XR*. Cisco Press: 2014.

RFC 7838, *Cisco's Enhanced Interior Gateway Routing Protocol (EIGRP)*, D. Savage, J. Ng, S. Moore, D. Slice, P. Paluch, R. White. http://tools.ietf.org/html/rfc7868, May 2016.

Cisco. *Cisco IOS Software Configuration Guides*. http://www.cisco.com.

Exam Preparation Tasks

As mentioned in the section "How to Use This Book" in the Introduction, you have a couple choices for exam preparation: the exercises here, Chapter 24, "Final Preparation," and the exam simulation questions in the Pearson Test Prep software.

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 2-8 lists these key topics and the page number on which each is found.

Table 2-8 Key Topics

Key Topic Element	Description	Page Number
Paragraph	EIGRP terminology	74
Paragraph	Topology table	75
Table 2-3	EIGRP packet types	76
Paragraph	Forming EIGRP neighbors	77
Paragraph	Classic configuration mode	78
Paragraph	EIGRP named mode	79
Paragraph	Passive interfaces	87
Paragraph	Authentication	91
Paragraph	Path metric calculation	93
Paragraph	EIGRP attribute propagation	94
Figure 2-11	EIGRP wide metrics formula	97
Paragraph	Custom K values	99
Paragraph	Unequal-cost load balancing	100

Complete Tables and Lists from Memory

There are no memory tables in this chapter.

Define Key Terms

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

autonomous system (AS), successor route, successor, feasible distance, reported distance, feasibility condition, feasible successor, topology table, EIGRP classic configuration, EIGRP named mode configuration, passive interface, K values, wide metrics, variance value

Use the Command Reference to Check Your Memory

This section includes the most important configuration and verification 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 2-9 with a piece of paper, read the description on the left side, and then see how much of the command you can remember.

The ENARSI 300-410 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, verify, and troubleshoot the topics covered in this chapter.

Table 2-9 Command Reference

Task	Command Syntax
Initialize EIGRP in classic configuration	**router eigrp** *as-number* **network** *network mask*
Initialize EIGRP in named mode configuration	**router eigrp** *process-name* *address-family* **{ipv4| ipv6} {unicast | vrf** *vrf-name}* **autonomous-system** *as-number* **network** *network mask*
Define the EIGRP router ID	**eigrp router-id** *router-id*
Configure an EIGRP-enabled interface to prevent neighbor adjacencies	Classic: (EIGRP Process) **passive-interface** *interface-id* Named Mode: **af-interface {default |** *interface-id}* **passive-interface**
Configure a keychain for EIGRP MD5 authentication	**key chain** *key-chain-name* **key** *key-number* **key-string** *password*

Task	Command Syntax
Configure MD5 authentication for an EIGRP interface	Classic: (EIGRP Process) **ip authentication key-chain eigrp** *as-number key-chain-name* **ip authentication mode eigrp** *as-number* **md5** Named Mode: **af-interface {default** | *interface-id***}** **authentication key-chain eigrp** *key-chain-name* **authentication mode md5**
Configure SHA authentication for EIGRP named mode interfaces	Named Mode: **af-interface {default** | *interface-id***}** **authentication mode hmac-sha-256** *password*
Modify the interface delay for an interface	**delay** *tens-of-microseconds*
Modify the EIGRP K values	**metric weights** *TOS* K_1 K_2 K_3 K_4 K_5 [K_6]
Modify the default number of EIGRP maximum paths that can be installed into the RIB	**maximum-paths** *maximum-paths*
Modify the EIGRP variance multiplier for unequal-cost load balancing	**variance** *multiplier*
Display the EIGRP-enabled interfaces	**show ip eigrp interface** [{*interface-id* [**detail**] | **detail**}]
Display the EIGRP topology table	**show ip eigrp topology** [**all-links**]
Display the configured EIGRP keychains and passwords	**show key chain**
Display the IP routing protocol information configured on the router	**show ip protocols**

CHAPTER 3

Advanced EIGRP

This chapter covers the following topics:

- **Failure Detection and Timers:** This section explains how EIGRP detects the absence of a neighbor and the convergence process.
- **Route Summarization:** This section explains the logic and configuration of summarizing routes on a router.
- **WAN Considerations:** This section reviews common design considerations with using EIGRP in a WAN.
- **Route Manipulation:** This section explains techniques for filtering or manipulating route metrics.

This chapter explores the mechanisms used by EIGRP during path computations for alternate routes due to network events. It also covers design concepts for accelerating convergence and increasing the scale of the EIGRP network. The last portion of the chapter reviews techniques for filtering or manipulating routes.

"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?' Quiz Questions."

Table 3-1 "Do I Know This Already?" Foundation Topics Section-to-Question Mapping

Foundation Topics Section	Questions
Failure Detection and Timers	1–4
Route Summarization	5–6
WAN Considerations	7
Route Manipulation	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 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 EIGRP hello timer for a high-speed interface?
   a. 1 second
   b. 5 seconds
   c. 10 seconds
   d. 20 seconds
   e. 30 seconds
   f. 60 seconds
2. What is the default EIGRP hello timer for a low-speed interface?
   a. 1 second
   b. 5 seconds
   c. 10 seconds
   d. 20 seconds
   e. 30 seconds
   f. 60 seconds
3. When a path is identified using EIGRP and in a stable fashion, the route is considered _____.
   a. passive
   b. dead
   c. active
   d. alive
4. How does an EIGRP router indicate that a path computation is required for a specific route?
   a. EIGRP sends out an EIGRP update packet with the topology change notification flag set.
   b. EIGRP sends out an EIGRP update packet with a metric value of zero.
   c. EIGRP sends out an EIGRP query with the delay set to infinity.
   d. EIGRP sends a route withdrawal, notifying other neighbors to remove the route from the topology table.
5. True or false: EIGRP summarization is performed with the command **summary-aggregate** *network subnet-mask* under the EIGRP process for classic mode configuration.
   a. True
   b. False
6. True or false: EIGRP automatic summarization is enabled by default and must be disabled to prevent issues with networks that cross classful network boundaries.
   a. True
   b. False
7. True or false: EIGRP stub site functions can be deployed at all branch sites, regardless of whether downstream EIGRP routers are present.
   a. True
   b. False

8. How do EIGRP offset lists manipulate a route?
   a. Completely removing a set of specific routes
   b. Reducing the total path metric to a more preferred value
   c. Adding the total path metric to a specific set of routes
   d. Adding delay to the path metric for a specific set of routes

Foundation Topics

Failure Detection and Timers

Key Topic

A secondary function of the EIGRP hello packets is to ensure that EIGRP neighbors are still healthy and available. EIGRP hello packets are sent out in intervals according to the *hello timer*. The default EIGRP hello timer is 5 seconds, but EIGRP uses 60 seconds on slow-speed interfaces (T1 or lower).

EIGRP uses a second timer called the *hold timer*, which is the amount of time EIGRP deems the router reachable and functioning. The hold time value defaults to three times the hello interval. The default value is 15 seconds (or 180 seconds for slow-speed interfaces). The hold time decrements, and upon receipt of a hello packet, the hold time resets and restarts the countdown. If the hold time reaches 0, EIGRP declares the neighbor unreachable and notifies the diffusing update algorithm (DUAL) of a topology change.

The hello timer is modified with the interface parameter command **ip hello-interval eigrp** *as-number seconds*, and the hold timer is modified with the interface parameter command **ip hold-time eigrp** *as-number seconds* when using EIGRP classic configuration mode.

For named mode configurations, the commands are placed under the **af-interface default** or the **af-interface** *interface-id* submodes. The command **hello-interval** *seconds* modifies the hello timer, and the command **hold-time** *seconds* modifies the hold timer when using named mode configuration.

Example 3-1 demonstrates changing the EIGRP hello interval to 3 seconds and the hold time to 15 seconds for R1 (in classic mode) and R2 (in named mode).

Example 3-1 *EIGRP Hello and Hold Timer Value Verification*

```
R1 (Classic Mode Configuration)
interface GigabitEthernet0/1
 ip address 10.12.1.1 255.255.255.0
 ip hello-interval eigrp 100 3
 ip hold-time eigrp 100 15
```

```
R2 (Named Mode Configuration)
router eigrp EIGRP-NAMED
 address-family ipv4 unicast autonomous-system 100
  !
 af-interface default
   hello-interval 3
   hold-time 15
  exit-af-interface
  !
```

```
  topology base
  exit-af-topology
  network 0.0.0.0
exit-address-family
```

The EIGRP hello and hold timers are verified by viewing the EIGRP interfaces with the command **show ip eigrp interfaces detail** [*interface-id*], as demonstrated in the following snippet:

```
R1# show ip eigrp interfaces detail gi0/1 | i Hello|Hold
   Hello-interval is 3, Hold-time is 15
   Hello's sent/expedited: 18348/5
```


NOTE EIGRP neighbors can still form an adjacency if the timers do not match, but the hellos must be received before the hold time reaches zero; that is, the hello interval must be less than the hold time.

Convergence

Key Topic

When a link fails, and the interface protocol moves to a down state, any neighbor attached to that interface moves to a down state, too. When an EIGRP neighbor moves to a down state, path recomputation must occur for any prefix where that EIGRP neighbor was a successor (an upstream router).

When EIGRP detects that it has lost its successor for a path, the feasible successor instantly becomes the successor route, providing a backup route. The router sends out an update packet for that path because of the new EIGRP path metrics. Downstream routers run their own DUAL algorithm for any affected prefixes to account for the new EIGRP metrics. It is possible for a change of the successor route or feasible successor to occur upon receipt of new EIGRP metrics from a successor router for a prefix.

Figure 3-1 demonstrates such a scenario when the link between R1 and R3 fails.

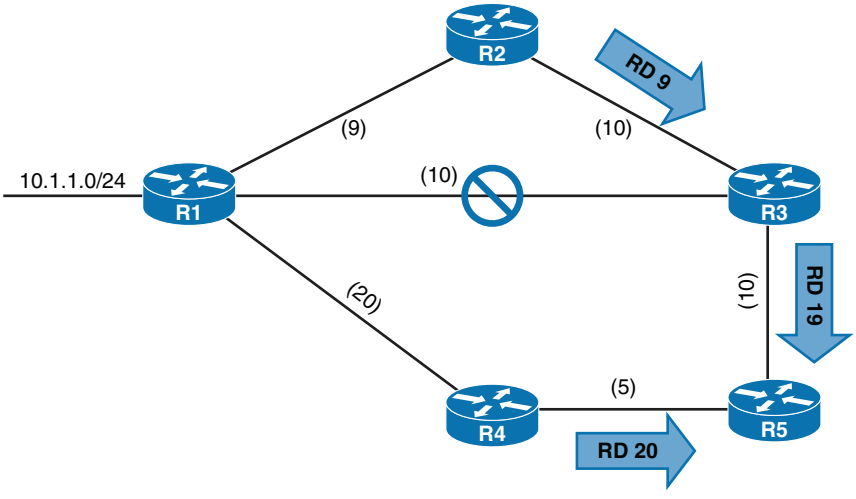

Figure 3-1 *EIGRP Topology with Link Failure*

R3 installs the feasible successor path advertised from R2 as the successor route. R3 sends an update packet with a new reported distance (RD) of 19 for the 10.1.1.0/24 prefix. R5 receives the update packet from R3 and calculates a feasible distance (FD) of 29 for the R1→R2→R3 path to 10.1.1.0/24. R5 compares that path to the one received from R4, which has a path metric of 25. R5 chooses the path through R4 as the successor route.

Example 3-2 provides simulated output of R5's EIGRP topology for the 10.1.1.0/24 prefix after the R1–R3 link fails.

Example 3-2 *Simulated EIGRP Topology for the 10.1.1.0/24 Network*

```
R5# show ip eigrp topology 10.1.1.0/24
EIGRP-IPv4 Topology Entry for AS(100)/ID(192.168.5.5) for 10.4.4.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 25
  Descriptor Blocks:
  *10.45.1.4 (GigabitEthernet0/2), from 10.45.1.4, Send flag is 0x0
      Composite metric is (25/20), route is Internal
      Vector metric:
        Hop count is 2
        Originating router is 192.168.1.1
  10.35.1.3 (GigabitEthernet0/1), from 10.35.1.3, Send flag is 0x0
      Composite metric is (29/19), route is Internal
      Vector metric:
        Hop count is 3
        Originating router is 192.168.1.1
```

Key Topic

If a feasible successor is not available for the prefix, DUAL must perform a new route calculation. The route state changes from passive (P) to active (A) in the EIGRP topology table.

The router detecting the topology change sends out query packets to EIGRP neighbors for the route. A query packet includes the network prefix with the delay set to infinity so that other routers are aware that it is now active. When the router sends EIGRP query packets, it sets the reply status flag for each neighbor on a prefix basis.

Upon receipt of a query packet, an EIGRP router does one of the following:

- It replies to the query that the router does not have a route to the prefix.
- If the query came from the successor for the route, the receiving router detects the delay set for infinity, sets the prefix as active in the EIGRP topology, and sends out a query packet to all downstream EIGRP neighbors for that route.
- If the query did not come from the successor for that route, it detects that the delay is set for infinity but ignores it because it did not come from the successor. The receiving router replies with the EIGRP attributes for that route.

The query process continues from router to router until a router establishes the query boundary. A query boundary is established when a router does not mark the prefix as active, meaning that it responds to a query as follows:

- It says it does not have a route to the prefix.
- It replies with EIGRP attributes because the query did not come from the successor.

When a router receives a reply for every downstream query that was sent out, it completes the DUAL, changes the route to passive, and sends a reply packet to any upstream routers that sent a query packet to it. Upon receiving the reply packet for a prefix, the reply packet is notated for that neighbor and prefix. The reply process continues upstream for the queries until the first router's queries are received.

Figure 3-2 shows a topology where the link between R1 and R2 failed.

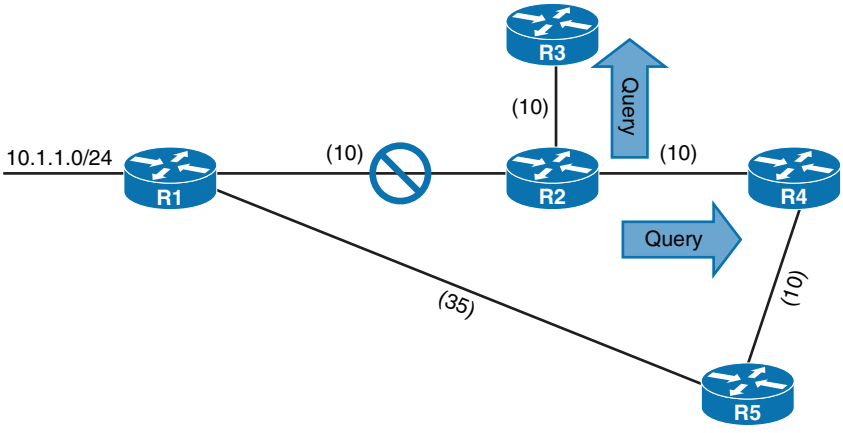

Figure 3-2 *EIGRP Convergence Topology*

The following steps are processed in order from the perspective of R2 calculating a new route to the 10.1.1.0/24 network:

Step 1. R2 detects the link failure. R2 does not have a feasible successor for the route, sets the 10.1.1.0/24 prefix as active, and sends queries to R3 and R4.

Step 2. R3 receives the query from R2 and processes the Delay field that is set to infinity. R3 does not have any other EIGRP neighbors and sends a reply to R2 that a route does not exist. R4 receives the query from R2 and processes the Delay field that is set to infinity. Because the query was received by the successor, and a feasible successor for the prefix does not exist, R4 marks the route as active and sends a query to R5.

Step 3. R5 receives the query from R4 and detects that the Delay field is set to infinity. Because the query was received by a nonsuccessor, and a successor exists on a different interface, a reply for the 10.4.4.0/24 network is sent back to R2 with the appropriate EIGRP attributes.

Step 4. R4 receives R5's reply, acknowledges the packet, and computes a new path. Because this is the last outstanding query packet on R4, R4 sets the prefix as passive. With all queries satisfied, R4 responds to R2's query with the new EIGRP metrics.

Step 5. R2 receives R4's reply, acknowledges the packet, and computes a new path. Because this is the last outstanding query packet on R4, R2 sets the prefix as passive.

Stuck in Active

Key Topic

DUAL is very efficient at finding loop-free paths quickly, and it normally finds backup paths in seconds. Occasionally, an EIGRP query is delayed because of packet loss, slow neighbors, or a large hop count. EIGRP maintains a timer, known as the active timer, which has a default value of 3 minutes (180 seconds). EIGRP waits half of the active timer value (90 seconds) for a reply. If the router does not receive a response within 90 seconds, the originating router sends a stuck in active (SIA) query to EIGRP neighbors that did not respond.

Upon receipt of an SIA query, the router should respond within 90 seconds with an SIA reply. An SIA reply contains the route information or provides information on the query process itself. If a router fails to respond to an SIA query by the time the active timer expires, EIGRP deems the router SIA. If the SIA state is declared for a neighbor, DUAL deletes all routes from that neighbor, treating the situation as if the neighbor responded with unreachable message for all routes.

NOTE Earlier versions of IOS terminated EIGRP neighbor sessions with routers that never replied to an SIA query.

You can only troubleshoot active EIGRP prefixes when the router is waiting for a reply. You show active queries with the command **show ip eigrp topology**.

To demonstrate the SIA process, Figure 3-3 illustrates a scenario in which the link between R1 and R2 failed. R2 sends out queries to R4 and R3. R4 sends a reply back to R2, and R3 sends a query on to R5.

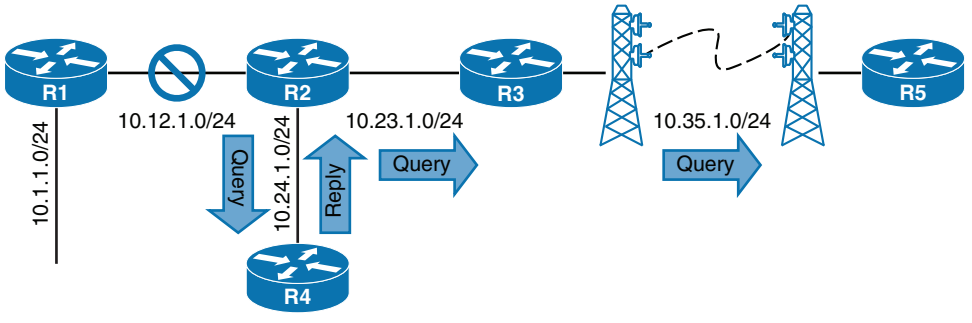

Figure 3-3 *EIGRP SIA Topology*

A network engineer who sees the syslog message and runs the **show ip eigrp topology active** command on R2 gets the output shown in Example 3-3. The r next to the peer's IP address (10.23.1.3) indicates that R2 is still waiting on the reply from R3 and that R4 responded. The command is then executed on R3, and R3 indicates that it is waiting on a response from R5. When you execute the command on R5, you do not see any active prefixes, which implies that R5 never received a query from R3. R3's query could have been dropped on the radio tower connection.

Example 3-3 *Output for SIA Timers*

```
R2# show ip eigrp topology active
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - Reply status

A 10.1.1.0/24, 0 successors, FD is 512640000, Q
    1 replies, active 00:00:01, query-origin: Local origin
         via 10.24.1.4 (Infinity/Infinity), GigabitEthernet 0/0
    1 replies, active 00:00:01, query-origin: Local origin
         via 10.23.1.3 (Infinity/Infinity), r, GigabitEthernet 0/1
    Remaining replies:
         via 10.23.1.3, r, GigabitEthernet 0/1
```

The active timer is set to 3 minutes by default. The active timer can be disabled or modified with the command **timers active-time** {**disabled** | *1-65535-minutes*} under the EIGRP process. With classic configuration mode, the command runs directly under the EIGRP process, and with named mode configuration, the command runs under the topology base. Example 3-4 demonstrates the modification of SIA to 2 minutes for R1 in classic mode and R2 in named mode.

Example 3-4 *Configuration of SIA Timers*

```
R1(config)# router eigrp 100
R1(config-router)# timers active-time 2
```

```
R2(config)# router eigrp EIGRP-NAMED
R2(config-router)# address-family ipv4 unicast autonomous-system 100
R2(config-router-af)# topology base
R2(config-router-af-topology)# timers active-time 2
```

You can see the active timer by examining the IP protocols on a router with the command **show ip protocols**. Filtering with the keyword **Active** streamlines the information, as demonstrated in the following snippet, where you see that R2's SIA timer is set to 2 minutes:

```
R2# show ip protocols | include Active

      Active Timer: 2 min
```

The SIA query now occurs after 1 minute, which is half of the configured SIA timer.

Route Summarization

EIGRP works well with minimal optimization. Scalability of an EIGRP autonomous system depends on route summarization. As the size of an EIGRP autonomous system increases, convergence may take longer. Scaling an EIGRP topology depends on summarizing routes in a hierarchical fashion. Figure 3-4 shows summarization occurring at the access, distribution, and core layers of the network topology. In addition to shrinking the routing table of all the routers, route summarization creates a query boundary and shrinks the query domain when a route goes active during convergence, thereby reducing SIA scenarios.

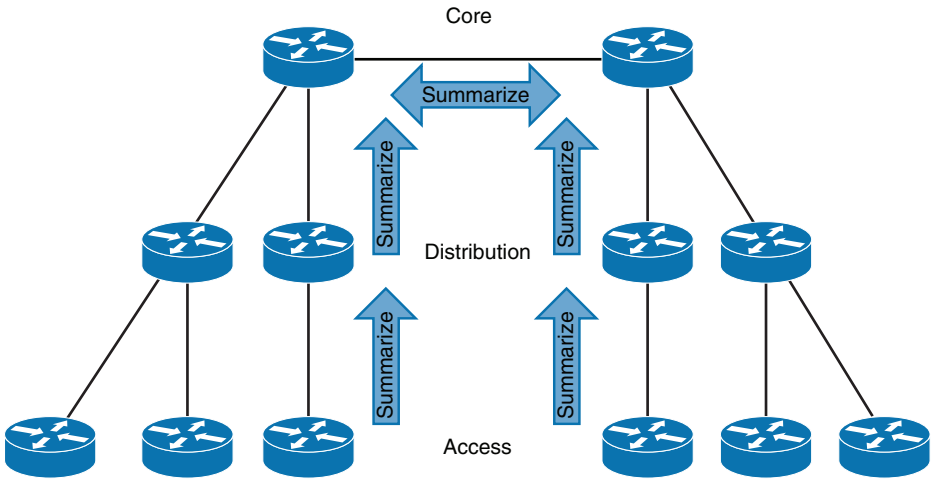

Figure 3-4 *EIGRP Hierarchical Summarization*

NOTE Route summarization on this scale requires hierarchical deployment of an IP addressing scheme.

Interface-Specific Summarization

EIGRP summarizes network prefixes on an interface-by-interface basis. A summary aggregate is configured for the EIGRP interface. Prefixes within the summary aggregate are suppressed, and the summary aggregate prefix is advertised in lieu of the original prefixes. The summary aggregate prefix is not advertised until a prefix matches it. Interface-specific summarization can be performed in any portion of the network topology.

Figure 3-5 illustrates the concept of EIGRP summarization. Without summarization, R2 advertises the 172.16.1.0/24, 172.16.3.0/24, 172.16.12.0/24, and 172.16.23.0/24 networks toward R4. R2 summarizes these network prefixes to the summary aggregate 172.16.0.0/16 prefix so that only one advertisement is sent to R4.

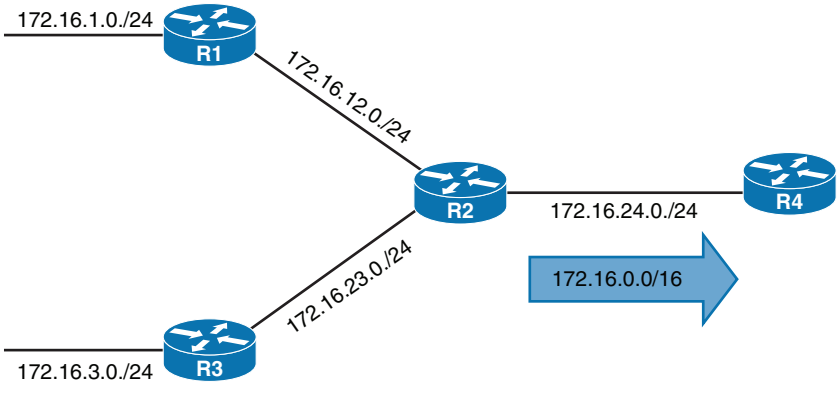

Figure 3-5 *EIGRP Summarization*

Key Topic

The advertisement of summary routes occurs on an interface-by-interface basis. For classic EIGRP configuration mode, you use the interface parameter command **ip summary-address** *eigrp as-number network subnet-mask* [**leak-map** *route-map-name*] to place an EIGRP summary aggregate on an interface. You perform summary route configuration for named mode under **af-interface** *interface-id*, using the command **summary-address** *network subnet-mask* [**leak-map** *route-map-name*].

The **leak-map** option allows the advertisement of the routes identified in the route map. Because suppression is avoided, the routes are considered leaked because they are advertised along with the summary aggregate. This allows for the use of longest-match routing to influence traffic patterns while suppressing most of the prefixes.

Example 3-5 shows R4's routing table before summarization is configured on R2. Notice that only /24 networks exist in the routing table.

Example 3-5 *R4's Routing Table Before Summarization*

```
R4# show ip route eigrp | begin Gateway
Gateway of last resort is not set

      172.16.0.0/16 is variably subnetted, 6 subnets, 2 masks
D        172.16.1.0/24 [90/3328] via 172.16.24.2, 1d01h, GigabitEthernet0/2
D        172.16.3.0/24 [90/3328] via 172.16.24.2, 1d01h, GigabitEthernet0/2
D        172.16.12.0/24 [90/3072] via 172.16.24.2, 1d01h, GigabitEthernet0/2
D        172.16.23.0/24 [90/3072] via 172.16.24.2, 1d01h, GigabitEthernet0/2
```

Example 3-6 shows the configuration for the 172.16.0.0/16 summary route that is advertised toward R4 out the Gi0/4 interface. Summary routes are always advertised based on the outgoing interface. The **af-interface default** option cannot be used with the **summary-address** command. It requires the use of a specific interface.

Example 3-6 *Configuration for EIGRP Summarization*

```
R2 (Classic Configuration)
interface gi0/4
 ip summary-address eigrp 100 172.16.0.0/16
```

```
R2 (Named Mode Configuration)
router eigrp EIGRP-NAMED
 address-family ipv4 unicast autonomous-system 100
  af-interface GigabitEthernet0/4
   summary-address 172.16.0.0 255.255.0.0
```

Example 3-7 shows R4's routing table after summarization is enabled on R2. The number of EIGRP paths has been drastically reduced, thereby reducing consumption of CPU and memory resources. Notice that all the routes are condensed into the 172.16.0.0/16 aggregate.

Example 3-7 *R4's Routing Table After Summarization*

```
R4# show ip route eigrp | begin Gateway
Gateway of last resort is not set

      172.16.0.0/16 is variably subnetted, 3 subnets, 3 masks
D        172.16.0.0/16 [90/3072] via 172.16.24.2, 00:00:24, GigabitEthernet0/2
```

NOTE Advertising a default route into EIGRP requires the summarization syntax described earlier in this section, except that the network and mask uses 0.0.0.0 0.0.0.0 (commonly referred to as double quad zeros).

Summary Discard Routes

Key Topic

EIGRP installs a discard route on the summarizing routers as a routing loop-prevention mechanism. A discard route is a route that matches the summary aggregate prefix with the destination Null0. This prevents routing loops where portions of the summarized network range do not have a more specific entry in the Routing Information Base (RIB) on the summarizing router. The AD for the Null0 route is 5 by default.

You view the discard route by using the **show ip route** *network subnet-mask* command, as shown in Example 3-8. Notice that the AD is set to 5, and it is connected to Null0, which means that packets are discarded if a longest match is not made.

Example 3-8 *Verification of AD Change for Summary Route AD*

```
R2# show ip route 172.16.0.0 255.255.0.0 | include entry|distance|via
Routing entry for 172.16.0.0/16
  Known via "eigrp 100", distance 5, metric 10240, type internal
  Redistributing via eigrp 100
  * directly connected, via Null0
```

Summarization Metrics

Key Topic

The summarizing router uses the lowest metric of the component routes in the summary aggregate prefix. The path metric for the summary aggregate is based on the path attributes of the path with the lowest metric. EIGRP path attributes such as total delay and minimum bandwidth are inserted into the summary route so that downstream routers can calculate the correct path metric for the summarized prefix.

In Figure 3-6, R2 has a path metric of 3072 for 172.16.1.0/24 prefix and a path metric of 3328 for the 172.16.3.0/24 prefix. The 172.16.0.0/16 summary aggregate is advertised with the path metric 3072 and the EIGRP path attributes received by R2 from R1.

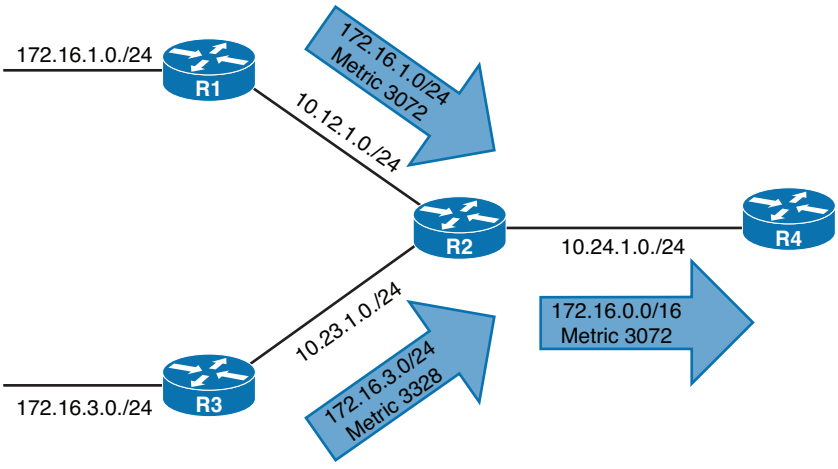

Figure 3-6 *EIGRP Summarization Metrics*

Every time a matching component route for the summary aggregate is added or removed, EIGRP must verify that the summary route is still using the attributes from the path with the lowest metric. If it is not, a new summary aggregate is advertised with updated EIGRP attributes, and downstream routes must run the DUAL again. The summary aggregate hides the smaller prefixes from downstream routers, but downstream routers are still burdened with processing updates to the summary aggregate.

The fluctuation in the path metric is resolved by statically setting the metric on the summary aggregate with the command **summary-metric** *network* {*/prefix-length* | *subnet-mask*} *bandwidth delay reliability load MTU*. Bandwidth is in kilobits per second (Kbps), delay is in 10-microsecond (µs) units, reliability and load are values between 1 and 255, and the maximum transmission unit (MTU) is the MTU for the interface.

Automatic Summarization

EIGRP supports automatic summarization, automatically summarizing network advertisements when they cross a classful network boundary. Figure 3-7 shows automatic summarization for the 10.1.1.0/24 route on R2 and the 10.5.5.0/24 network on R4. R2 and R4 only advertise the classful network 10.0.0.0/8 toward R3.

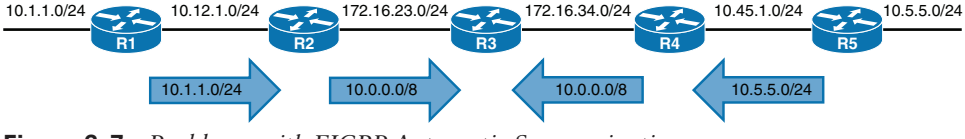

Figure 3-7 *Problems with EIGRP Automatic Summarization*

Example 3-9 shows the routing table for R3. Notice that there are no routes for the 10.1.1.0/24 or 10.5.5.0/24 networks; there is only a route for 10.0.0.0/8 with next hops of R2 and R4. Traffic sent to either network could be sent out the wrong interface. This problem affects network traffic traveling across the network in addition to traffic originating from R3.

Example 3-9 *Path Selection Problems on R3 with Automatic Summarization*

```
R3# show ip route eigrp | begin Gateway
Gateway of last resort is not set

D     10.0.0.0/8 [90/3072] via 172.16.34.4, 00:08:07, GigabitEthernet0/0
                 [90/3072] via 172.16.23.2, 00:08:07, GigabitEthernet0/1
```

Example 3-10 displays a similar behavior for the 172.16.23.0/24 and 172.16.34.0/24 networks as they are advertised as 172.16.0.0/16 networks from R2 to R1. The identical advertisement occurs from R4 to R5, too.

Example 3-10 *Automatic Summarization on R1 and R5*

```
R1# show ip route eigrp | begin Gateway
Gateway of last resort is not set

D     172.16.0.0/16 [90/3072] via 10.12.1.2, 00:09:50, GigabitEthernet0/0
```

```
R5# show ip route eigrp | begin Gateway
Gateway of last resort is not set

D     172.16.0.0/16 [90/3072] via 10.45.1.4, 00:09:50, GigabitEthernet0/1
```

Current releases of IOS XE disable EIGRP classful network automatic summarization by default. You enable automatic summarization by using the command **auto-summary** under the EIGRP process for classic configuration mode or by using the command **topology base** for named mode configurations. To disable automatic summarization, use the command **no auto-summary.**

WAN Considerations

EIGRP does not change behavior based on the media type of an interface. Serial and Ethernet interfaces are treated the same. Some WAN topologies may require special consideration for bandwidth utilization, split horizon, or next-hop self. The following sections explain each scenario in more detail.

EIGRP Stub Router

Key Topic

A proper network design provides redundancy where dictated by business requirements to ensure that a remote location always maintains network connectivity. To overcome single points of failure, you can add additional routers at each site, add redundant circuits (possibly with different service providers), use different routing protocols, or use virtual private network (VPN) tunnels across the Internet for backup transport.

Figure 3-8 shows a topology with R1 and R2 providing connectivity at two key data center locations. They are interconnected with a 10 Gbps circuit (10.12.1.0/24) and maintain backup connectivity to each other with a backup VPN tunnel. R1 and R2 connect to R3 through T1 (1.5 Mbps) circuits. R1 is advertising the 10.1.1.0/24 prefix directly to R2 and R3, and R2 advertises the 10.2.2.0/24 prefix to R1 and R3.

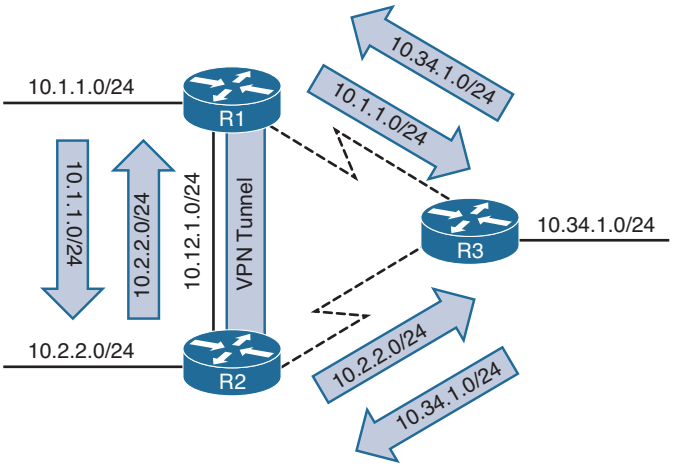

Figure 3-8 *WAN Connectivity Between Two Data Centers*

NOTE The serial WAN link network advertisements are not illustrated in Figures 3-8 to 3-12, which instead focus on advertisement of routes that are multiple hops away.

Proper network design considers traffic patterns during normal operations and throughout various failure scenarios to prevent suboptimal routing or routing loops. Figure 3-9 demonstrates the failure of the 10 Gbps network link between R1 and R2. R3 continues to advertise the 10.1.1.0/24 prefix to R2 even though R1's traffic should be taking the VPN tunnel to reach R2. The scenario happens in the same fashion with 10.2.2.0/24 traffic transiting R3 instead of going across the VPN tunnel.

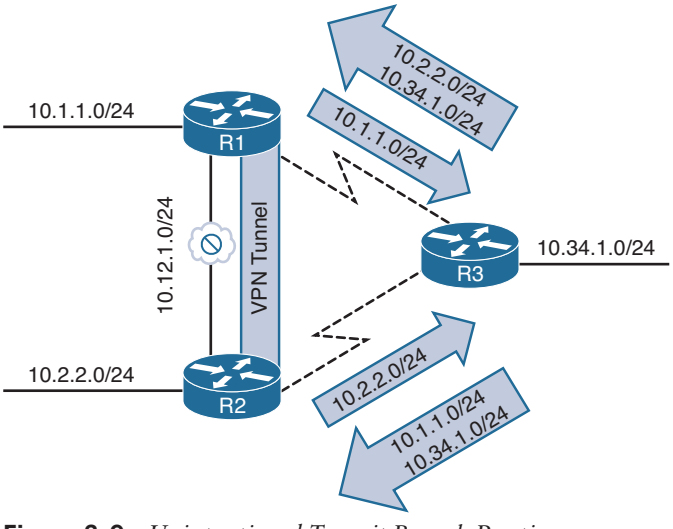

Figure 3-9 *Unintentional Transit Branch Routing*

The EIGRP stub functionality prevents scenarios like this from happening and allows an EIGRP router to conserve router resources. An EIGRP stub router does not advertise routes

that it learns from other EIGRP peers. By default, EIGRP stubs advertise only connected and summary routes, but they can be configured so that they only receive routes or advertise any combination of redistributed routes, connected routes, or summary routes.

In Figure 3-10, R3 was configured as a stub router, and the 10 Gbps link between R1 and R2 fails. Traffic between R1 and R2 uses the backup VPN tunnel and does not traverse R3's T1 circuits because R3 is only advertising its connected networks (10.34.1.0/24).

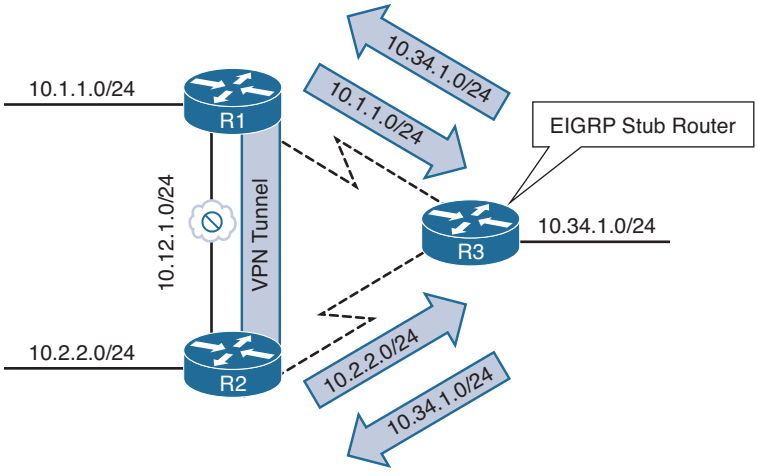

Figure 3-10 *Stopping Transit Branch Routing with an EIGRP Stub Router*

Key Topic

The EIGRP stub router announces itself as a stub within the EIGRP hello packet. Neighboring routers detect the stub field and update the EIGRP neighbor table to reflect the router's stub status. If a route goes active, EIGRP does not send EIGRP queries to an EIGRP stub router. This provides faster convergence within an EIGRP autonomous system because it decreases the size of the query domain for that prefix.

You configure a stub router by placing the command **eigrp stub** {**connected** | **receive-only** | **redistributed** | **static** | **summary**} under the EIGRP process for classic configuration and under the address family for named mode configuration. Example 3-11 demonstrates the stub configuration for EIGRP classic mode and named mode.

Example 3-11 *EIGRP Stub Configuration*

```
R3 (Classic Configuration)
router eigrp 100
  network 0.0.0.0 255.255.255.255
  eigrp stub
```

```
R3 (Named Mode Configuration)
router eigrp EIGRP-NAMED
 address-family ipv4 unicast autonomous-system 100
  eigrp stub
```

NOTE The **receive-only** option cannot be combined with other EIGRP stub options as it does not advertise any networks to its neighbors. The network design should be given special consideration to ensure bidirectional connectivity for any networks connected to an EIGRP router with the **receive-only** stub option to ensure that routers know how to send return traffic.

Stub Site Functions

A common problem with EIGRP stub routers is forgetting that they do not advertise EIGRP routes that they learn from another peer. Figure 3-11 expands on the previous topology and adds the R4 router to the branch network; R4 is attached to R3.

Key Topic

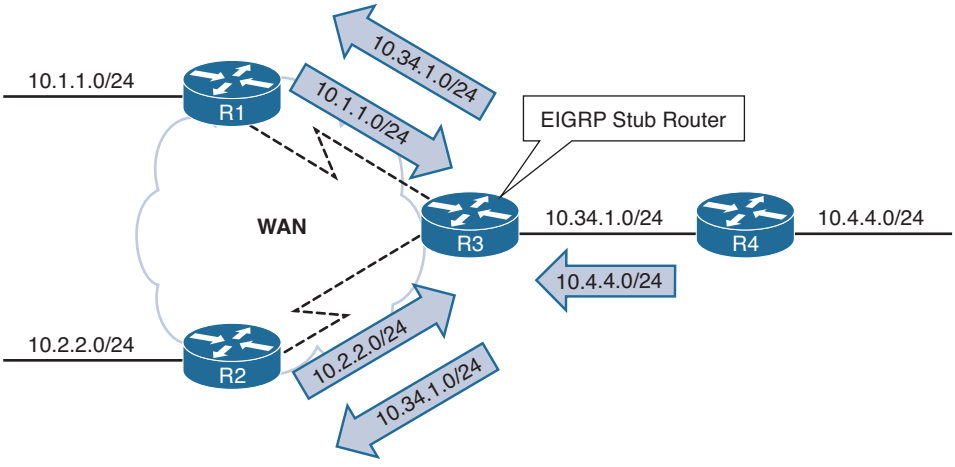

Figure 3-11 *Problems with Downstream Routing and EIGRP Stub Routers*

Say that a junior network engineer recently learned about the EIGRP stub function and configured it on R3 to prevent transient routing and reduce the size of the query domain. The users attached to R4's 10.4.4.0/24 network start to complain because they cannot access any resources attached to R1 and R2; however, they can still communicate with devices attached to R3.

Example 3-12 demonstrates the EIGRP learned routes on R1 and R4. R1 is missing the 10.4.4.0/24 prefix, and R4 is missing the 10.1.1.0/24 prefix. Both prefixes are missing because R3 is an EIGRP stub router.

Example 3-12 *Missing Routes Because of EIGRP Stub Routing*

```
R1# show ip route eigrp | begin Gateway
Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 9 subnets, 2 masks
D        10.34.1.0/24 [90/61440] via 10.13.1.3, 00:20:26, GigabitEthernet0/5
```

```
R4# show ip route eigrp | begin Gateway
Gateway of last resort is not set
```

```
      10.0.0.0/8 is variably subnetted, 6 subnets, 2 masks
! These networks are the serial links directly attached to R3
D        10.13.1.0/24 [90/61440] via 10.34.1.3, 00:19:39, GigabitEthernet0/1
D        10.23.1.0/24 [90/61440] via 10.34.1.3, 00:19:39, GigabitEthernet0/1
```

Key Topic

The EIGRP stub site feature builds on EIGRP stub capabilities that allow a router to advertise itself as a stub to peers only on the specified WAN interfaces but allow it to exchange routes learned on LAN interfaces. EIGRP stub sites provide the following key benefits:

- EIGRP neighbors on WAN links do not send EIGRP queries to the remote site when a route becomes active.
- The EIGRP stub site feature allows downstream routers to receive and advertise network prefixes across the WAN.
- The EIGRP stub site feature prevents the EIGRP stub site route from being a transit site.

The EIGRP stub site feature works by identifying the WAN interfaces and then setting an EIGRP stub site identifier. Routes received from a peer on the WAN interface are tagged with an EIGRP stub site identifier attribute. When EIGRP advertises network prefixes out a WAN-identified interface, it checks for an EIGRP stub site identifier. If one is found, the route is not advertised; if an EIGRP stub site identifier is not found, the route is advertised.

Figure 3-12 illustrates the concept further with R3 being configured as a stub site router and the serial links configured as EIGRP WAN interfaces:

Step 1. R1 advertises the 10.1.1.0/24 route to R3, and the 10.1.1.0/24 route is received on R3's WAN interface. R3 is then able to advertise that prefix to the downstream router R4.

Step 2. R2 advertises the 10.2.2.0/24 route to R3, and the 10.2.2.0/24 route is received on R3's other WAN interface. R3 is then able to advertise that prefix to the downstream router R4.

Step 3. R4 advertises the 10.4.4.0/24 network to R3. R3 checks the 10.4.4.0/24 route for the EIGRP stub site attribute before advertising that prefix out either WAN interface. R3 is able to advertise the prefix to R1 and R2 because it does not contain an EIGRP stub site identifier attribute.

Notice that R3 does not advertise the 10.1.1.0/24 prefix to R2 and that it does not advertise the 10.2.2.0/24 prefix to R1. This is because the EIGRP stub site attribute was added upon receipt of the prefix and blocked during advertisement out the other WAN interface.

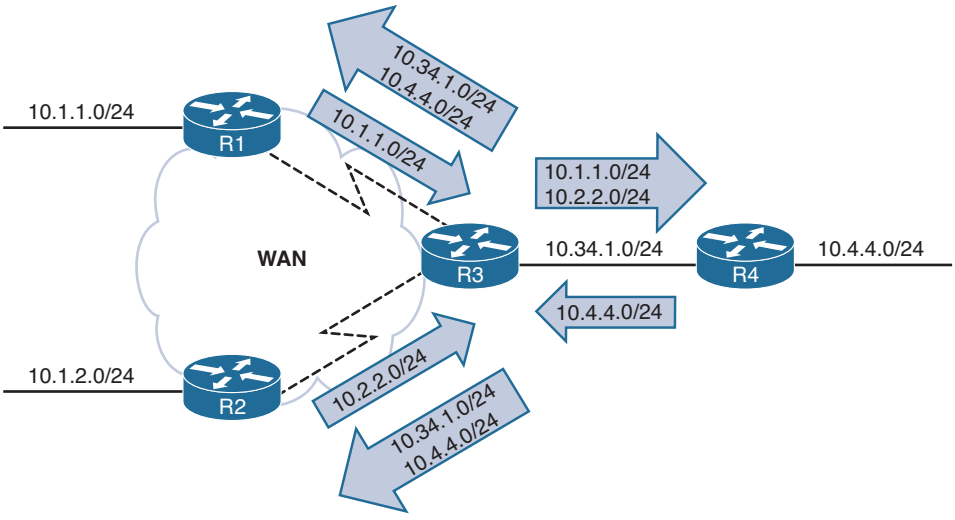

Figure 3-12 *EIGRP Stub Site Feature*

The EIGRP stub site function is available only in EIGRP named mode configuration. The WAN interfaces are identified underneath the **af-interface** *interface-id* hierarchy and use the **stub-site wan-interface** command. The stub site function and identifier are enabled with the command **eigrp stub-site** *as-number:identifier*. The *as-number:identifier* must remain the same for all devices in a site. Upon associating an interface to the EIGRP stub site, the router resets the EIGRP neighbor for that interface.

Example 3-13 provides the EIGRP stub site configuration for R3 for both serial interfaces.

Example 3-13 *EIGRP Stub Site Configuration*

```
R3
router eigrp EIGRP-NAMED
 address-family ipv4 unicast autonomous-system 100
  af-interface Serial1/0
   stub-site wan-interface
  exit-af-interface
  !
  af-interface Serial1/1
   stub-site wan-interface
  exit-af-interface
  eigrp stub-site 100:1
 exit-address-family
```

Example 3-14 verifies that the 10.1.1.0/24 route learned from R3's serial interfaces are tagged with the EIGRP stub site attribute. R4 was selected for this output to demonstrate that the attribute is passed to other downstream routers.

Example 3-14 *Verification of Routes Learned from the WAN Interface*

```
R4# show ip eigrp topology 10.1.1.0/24
EIGRP-IPv4 VR(EIGRP-NAMED) Topology Entry for AS(100)/ID(192.168.4.4) for 10.1.1.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 8519680, RIB is
66560
  Descriptor Blocks:
  10.34.1.3 (GigabitEthernet0/1), from 10.34.1.3, Send flag is 0x0
      Composite metric is (8519680/7864320), route is Internal
      Vector metric:
        Minimum bandwidth is 100000 Kbit
        Total delay is 30000000 picoseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 2
        Originating router is 192.168.1.1
      Extended Community: StubSite:100:1
```

A major benefit to the EIGRP stub site feature is that the stub functionality can be passed to a branch site that has multiple edge routers. As long as each router is configured with the EIGRP stub site feature and maintains the same stub site identifier, the site does not become a transit routing site; however, it still allows for all the networks to be easily advertised to other routers in the EIGRP autonomous system.

Example 3-15 verifies that R1 recognizes R3 as an EIGRP stub router and does not send it any queries when a route becomes active.

Example 3-15 *EIGRP Stub Router Flags*

```
R1# show ip eigrp neighbors detail Serial1/0
EIGRP-IPv4 VR(EIGRP-NAMED) Address-Family Neighbors for AS(100)
H   Address                 Interface              Hold Uptime   SRTT   RTO  Q  Seq
                                                   (sec)         (ms)       Cnt Num
1   10.13.1.3               Serial                  11 00:04:39   13   100  0  71
   Time since Restart 00:04:35
   Version 23.0/2.0, Retrans: 0, Retries: 0, Prefixes: 3
   Topology-ids from peer - 0
   Topologies advertised to peer:   base

   Stub Peer Advertising (CONNECTED STATIC SUMMARY REDISTRIBUTED ) Routes
   Suppressing queries
Max Nbrs: 0, Current Nbrs: 0
```

NOTE Although not required, configuring the EIGRP stub site feature on all branch routers keeps the configuration consistent and makes possible additional nondisruptive deployment of routers at that site in the future. The same *as-number:identifier* could be used for all of the site's WAN interfaces because those networks would never be advertised to other EIGRP stub sites, with the exception of tunnels or backdoor network links, which helps prevent suboptimal routing.

IP Bandwidth Percentage

Key Topic

Routing Information Protocol (RIP) and other routing protocols can consume all the bandwidth on slow circuits. Although the routers may have accurate routing tables, a router is worthless if no bandwidth is available for sending data packets. EIGRP overcomes this deficiency by setting the maximum available bandwidth for all circuits to 50%. This allows EIGRP to use 50% of the bandwidth and reserves 50% of the bandwidth for data packets.

The interface parameter command **ip bandwidth-percent eigrp** *as-number percentage* changes the EIGRP available bandwidth for a link on EIGRP classic configuration. The available bandwidth for EIGRP is modified under the **af-interface default** submode or the **af-interface** *interface-id* submode with the command **bandwidth-percent** *percentage* in a named mode configuration.

Example 3-16 provides the configuration for setting the bandwidth available for EIGRP on R1 for classic and named mode configurations.

Example 3-16 *EIGRP Bandwidth Percentage Configuration*

```
R1 (Classic Configuration)
interface GigabitEthernet0/0
ip address 10.34.1.4 255.255.255.0
 ip bandwidth-percent eigrp 100 25
```

```
R1 (Named Mode Configuration)
router eigrp EIGRP-NAMED
 address-family ipv4 unicast autonomous-system 100
  af-interface GigabitEthernet0/0
   bandwidth-percent 25
```

You can see the EIGRP bandwidth settings by looking at the EIGRP interfaces with the **detail** option. Example 3-17 shows the EIGRP bandwidth settings.

Example 3-17 *Viewing the EIGRP Bandwidth Percentage*

```
R1# show ip eigrp interfaces detail
! Output omitted for brevity
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
Gi0/0       1         0/0        0/0           1       0/0            50           0
..
   Interface BW percentage is 25
   Authentication mode is not set
```

Split Horizon

The first distance vector routing protocols advertised network prefixes out all interfaces for all known routes. Figure 3-13 demonstrates this behavior, with three routers processing the advertisements:

Key Topic

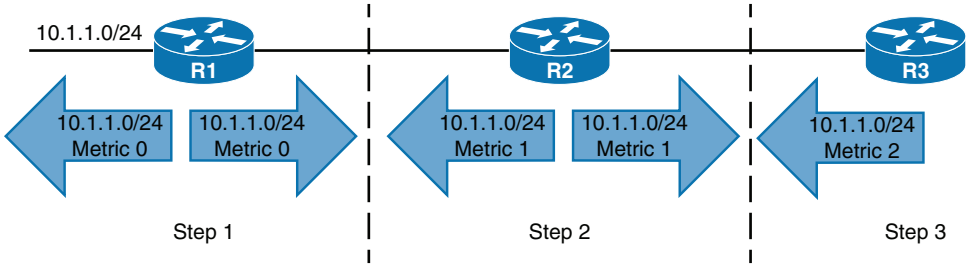

Figure 3-13 *Advertising All Routes Out All Interfaces*

Step 1. R1 advertises the 10.1.1.0/24 network out all of its interfaces.

Step 2. R2 adds to the metric and re-advertises the network to R1 and R3. Advertising a route (10.1.1.0/24) back to the originating router (R1) is known as a *reverse route*. Reverse routes waste network resources because R1 discards the route from R2 because 10.1.1.0/24 is the connected network and has a higher AD.

Step 3. R3 adds to the metric and advertises the reverse route to R2. R2 discards the route from R3 because it has a higher metric than the route from R1.

Figure 3-14 demonstrates a link failure between R1 and R2. R2 removes the 10.1.1.0/24 route learned from R1. It is possible that before R2 announces that the 10.1.1.0/24 network is unreachable, R3 advertises the 10.1.1.0/24 route with a metric of 2 out all interfaces

R2 installs the route advertised from R3, which has the next-hop IP address 10.23.1.3. R3 still maintains the original route advertised from R2 with the next-hop IP address 10.23.1.2. This causes a routing loop if a packet is sent from R2 or R3 to the 10.1.1.0/24 network. Eventually, the route entries time out and end the routing loop.

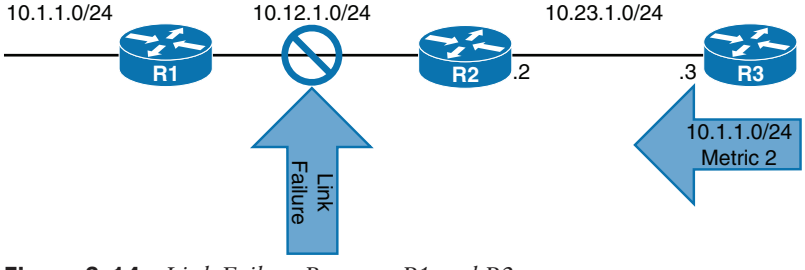

Figure 3-14 *Link Failure Between R1 and R2*

Split horizon prevents the advertisement of reverse routes and prevents scenarios like the one shown in Figure 3-14 from happening. Figure 3-15 shows the same scenario with split horizon.

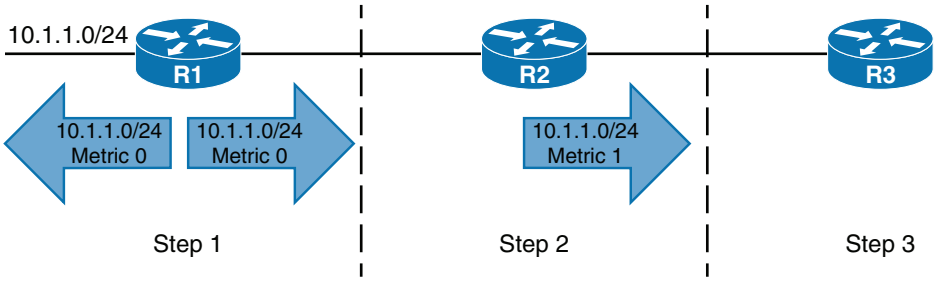

Figure 3-15 *Routing Updates with Split Horizon Enabled*

The following steps occur as R1 advertises the 10.1.10/24 prefix with split horizon enabled:

Step 1. R1 advertises the 10.1.1.0/24 network out all of its interfaces.

Step 2. R2 adds to the metric and re-advertises the network to R3 but does not advertise the route back to R1 because of split horizon.

Step 3. R3 receives the route from R2 but does not advertise the route back to R2 because of split horizon.

EIGRP enables split horizon on all interfaces by default. When an interface connects to a multi-access medium that does not support full-mesh connectivity for all nodes, split horizon needs to be disabled. This scenario is commonly found on hub-and-spoke topologies such as Frame Relay, Dynamic Multipoint Virtual Private Network (DMVPN), or Layer 2 Virtual Private Network (L2VPN).

Figure 3-16 shows a hub-and-spoke topology where R1 is the hub, and R2 and R3 are spoke routers that can only communicate with the hub router. R1 uses the same interface for establishing the DMVPN tunnel, and split horizon prevents routes received from one spoke (R2) from being advertised to the other spoke (R3).

Notice that the EIGRP routing table is not complete for all the routers. R2 only has a remote route for R1's 10.1.1.0/24 network, and R3 only has a remote route for R1's 10.1.1.0/24 network. Split horizon on R1 prevents routes received from one spoke from being advertised to the other spoke.

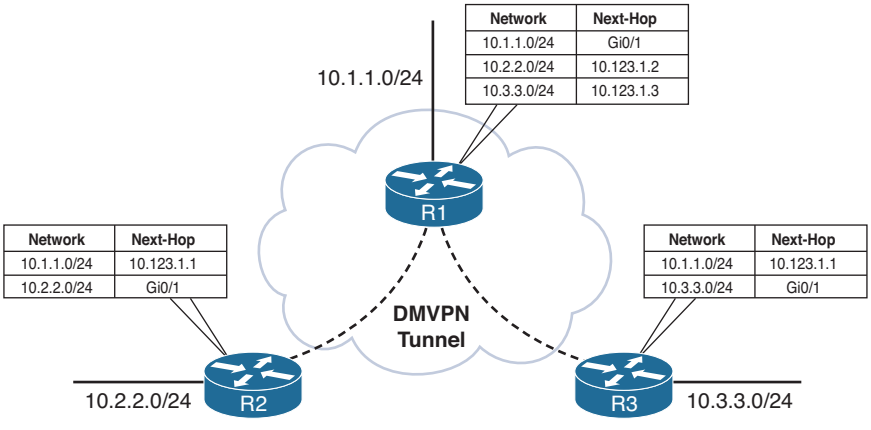

Figure 3-16 *Hub-and-Spoke Topology with Split Horizon*

You disable split horizon on a specific interface by using the interface parameter command **no ip split-horizon eigrp** *as-number* with EIGRP classic configuration. You disable split horizon on EIGRP named mode configuration under the **af-interface default** or **af-interface** *interface-id*, using the command **no split-horizon**. Example 3-18 shows a configuration to disable split horizon on the tunnel 100 interface.

Example 3-18 *Configuration to Disable Split Horizon*

```
R1 (Classic Configuration)
interface tunnel 100
  ip address 10.123.1.1 255.255.255.0
  no ip split-horizon eigrp 100
```

```
R1 (Named Mode Configuration)
router eigrp EIGRP-NAMED
 address-family ipv4 unicast autonomous-system 100
  af-interface tunnel 100
   no split-horizon
```

Figure 3-17 shows the routing table of all the routers after split horizon is disabled on R1. Notice that all routers have complete EIGRP routes.

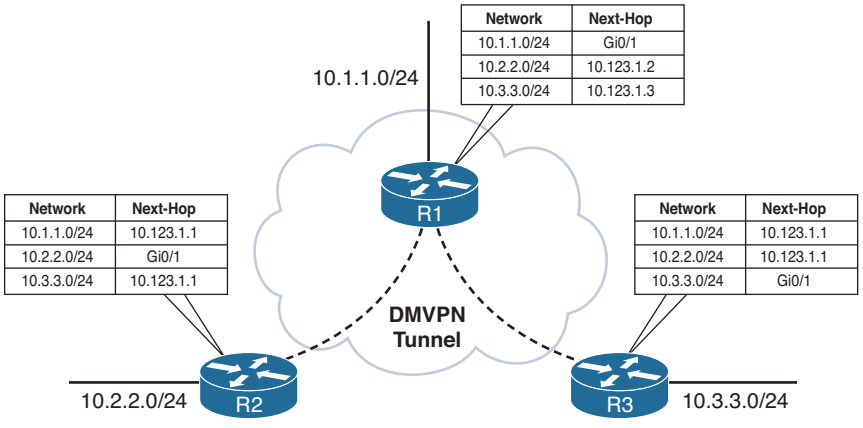

Figure 3-17 *Hub-and-Spoke Topology with Split Horizon Disabled*

Route Manipulation

Route manipulation involves selectively identifying routes that are advertised or received from neighbor routers. The routes can be modified to alter traffic patterns or removed to reduce memory utilization or to improve security. The following sections explain how routes are removed with filtering or modified with an EIGRP offset list.

Route Filtering

EIGRP supports filtering of routes as they are received or advertised from an interface. Filtering of routes can be matched against:

- Access control lists (ACLs) (named or numbered)
- IP prefix lists
- Route maps
- Gateway IP addresses

As shown in Figure 3-18, inbound filtering drops routes prior to the DUAL processing, which results in the routes not being installed into the RIB because they are not known. However, if the filtering occurs during outbound route advertisement, the routes are processed by DUAL and are installed into the local RIB of the advertising router.

Key Topic

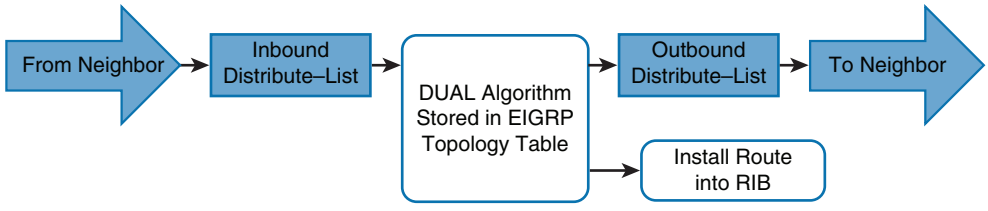

Figure 3-18 *EIGRP Distribute List Filtering Logic*

Filtering is accomplished with the command **distribute-list** {*acl-number* | *acl-name* | **prefix** *prefix-list-name* | **route-map** *route-map-name* | **gateway** *prefix-list-name*} {**in** | **out**} [*interface-id*]. EIGRP classic configuration places the command under the EIGRP process, while named mode configuration places the command under the topology base.

Prefixes that match against **deny** statements are filtered, and prefixes that match against a **permit** are passed. The **gateway** command can be used by itself or combined with a prefix list, an ACL, or a route map to restrict prefixes based on the next-hop forwarding address. Specifying an interface restricts the filtering to the interface that the route was received or advertised out of.

Figure 3-19 illustrates an EIGRP network for demonstrating inbound and outbound route filtering on R2.

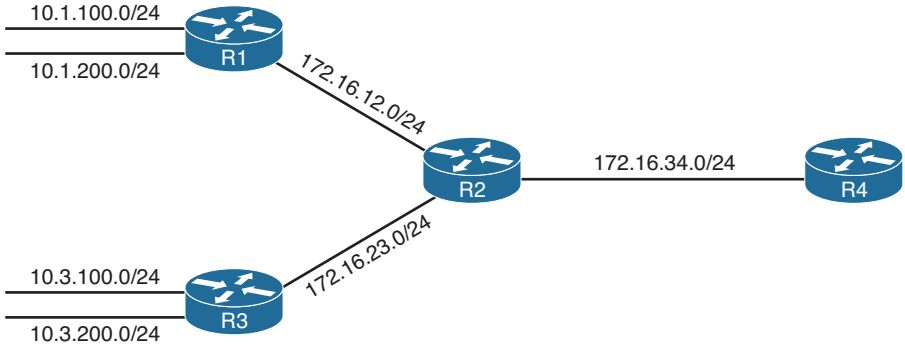

Figure 3-19 *EIGRP Distribution List Filtering Topology*

Example 3-19 shows the routing tables of R2 and R4 before the route filtering is applied. Notice that all the routes in the 10.1.0.0/16 and 10.3.0.0/16 range are present on both R2 and R4.

Example 3-19 *R2 and R4 Routing Tables*

```
R2# show ip route eigrp | begin Gateway
Gateway of last resort is not set

      10.0.0.0/24 is subnetted, 4 subnets
D        10.1.100.0 [90/15360] via 172.16.12.1, 00:05:45, GigabitEthernet0/1
D        10.1.200.0 [90/15360] via 172.16.12.1, 00:05:36, GigabitEthernet0/1
D        10.3.100.0 [90/15360] via 172.16.23.3, 00:06:26, GigabitEthernet0/3
D        10.3.200.0 [90/15360] via 172.16.23.3, 00:06:14, GigabitEthernet0/3
```

```
R4# show ip route eigrp | begin Gateway
Gateway of last resort is not set

      10.0.0.0/24 is subnetted, 4 subnets
D        10.1.100.0 [90/3328] via 172.16.24.2, 00:05:41, GigabitEthernet0/2
D        10.1.200.0 [90/3328] via 172.16.24.2, 00:05:31, GigabitEthernet0/2
D        10.3.100.0 [90/3328] via 172.16.24.2, 00:06:22, GigabitEthernet0/2
D        10.3.200.0 [90/3328] via 172.16.24.2, 00:06:10, GigabitEthernet0/2
      172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
D        172.16.12.0/24
           [90/3072] via 172.16.24.2, 00:07:04, GigabitEthernet0/2
D        172.16.23.0/24
           [90/3072] via 172.16.24.2, 00:07:04, GigabitEthernet0/2
```

Example 3-20 shows the configuration of R2 to demonstrate inbound filtering of 10.1.100.0/24 and outbound filtering of 10.3.100.0/24. The inbound filter uses a standard ACL to filter inbound routes and a prefix list to filter outbound advertisements. The **prefix** keyword must be used when referencing a prefix list.

Example 3-20 *EIGRP Route Filtering Configuration*

```
R2 (Classic Configuration)
ip access-list standard FILTER-R1-10.1.100.X
 deny 10.1.100.0
 permit any
!
ip prefix-list FILTER-R3-10.3.100.X deny 10.3.100.0/24
ip prefix-list FILTER-R3-10.3.100.X permit 0.0.0.0/0 le 32
!
router eigrp 100
 distribute-list FILTER-R1-10.1.100.X in
 distribute-list prefix FILTER-R3-10.3.100.X out
```

```
R2 (Named Mode Configuration)
ip access-list standard FILTER-R1-10.1.100.X
 deny 10.1.100.0
 permit any
!
ip prefix-list FILTER-R3-10.3.100.X deny 10.3.100.0/24
ip prefix-list FILTER-R3-10.3.100.X permit 0.0.0.0/0 le 32
!
router eigrp EIGRP-NAMED
 address-family ipv4 unicast autonomous-system 100
  topology base
   distribute-list FILTER-R1-10.1.100.X in
   distribute-list prefix FILTER-R3-10.3.100.X out
```


NOTE Conditional matching using ACLs, prefix lists, and route maps is covered in more detail in Chapter 15, "Route Maps and Conditional Forwarding."

Example 3-21 shows the routing table on R2 and R4 after EIGRP filtering is enabled on the routers. The 10.1.100.0/24 prefix is filtered upon receipt by R2, and it is not present in the EIGRP topology to advertise to R4. R2 still has the 10.3.100.0/24 prefix installed in the RIB, but the route is not advertised to R4. R4 does not have the 10.1.100.0/24 prefix or the 10.3.100.0/24 prefix in the routing table.

Example 3-21 *EIGRP Route Filtering Verification*

```
R2# show ip route eigrp | begin Gateway
Gateway of last resort is not set

      10.0.0.0/24 is subnetted, 4 subnets
D        10.1.200.0 [90/15360] via 172.16.12.1, 00:06:58, GigabitEthernet0/1
D        10.3.100.0 [90/15360] via 172.16.23.3, 00:06:15, GigabitEthernet0/3
D        10.3.200.0 [90/15360] via 172.16.23.3, 00:06:15, GigabitEthernet0/3
```

```
R4# show ip route eigrp | begin Gateway
Gateway of last resort is not set

      10.0.0.0/24 is subnetted, 2 subnets
D        10.1.200.0 [90/3328] via 172.16.24.2, 00:00:31, GigabitEthernet0/2
D        10.3.200.0 [90/3328] via 172.16.24.2, 00:00:31, GigabitEthernet0/2
      172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
D        172.16.12.0/24
           [90/3072] via 172.16.24.2, 00:00:31, GigabitEthernet0/2
D        172.16.23.0/24
           [90/3072] via 172.16.24.2, 00:00:31, GigabitEthernet0/2
```

Traffic Steering with EIGRP Offset Lists

Key Topic

Modifying the EIGRP path metric provides traffic engineering in EIGRP. Modifying the delay setting for an interface modifies all routes that are received and advertised from that router's interface. *Offset lists* allow for the modification of route attributes based on the direction of the update, a specific prefix, or a combination of direction and prefix.

An offset list is configured with the command **offset-list** *offset-value* {*acl-number* | *acl-name*} {**in** | **out**} [*interface-id*] to modify the metric value of a route. Specifying an interface restricts the conditional match for the offset list to the interface that the route is received or advertised out of. EIGRP classic configuration places the command under the EIGRP process, while named mode configuration places the command under the topology base.

On the downstream neighbor, the path metric increases by the offset value specified in the offset list. The offset value is calculated from an additional delay value that was added to the existing delay in the EIGRP path attribute. Figure 3-20 shows the modified path metric formula when an offset delay is included.

$$\text{Metric} + \text{offset} = 256 * \left(\left(\frac{10^7}{\text{Min. Bandwidth}} + \frac{\text{Total Delay}}{10} \right) + \text{Offset Delay} \right)$$

Equals

$$\text{Offset} = 256 * \text{Offset Delay}$$

Figure 3-20 *EIGRP Offset Value Calculation*

Figure 3-21 shows an EIGRP topology that helps demonstrate EIGRP offset lists. R1 is advertising the 10.1.100.0/24 and 10.1.200.0/24 networks, and R3 is advertising the 10.3.100.0/24 and 10.3.200.0/24 networks.

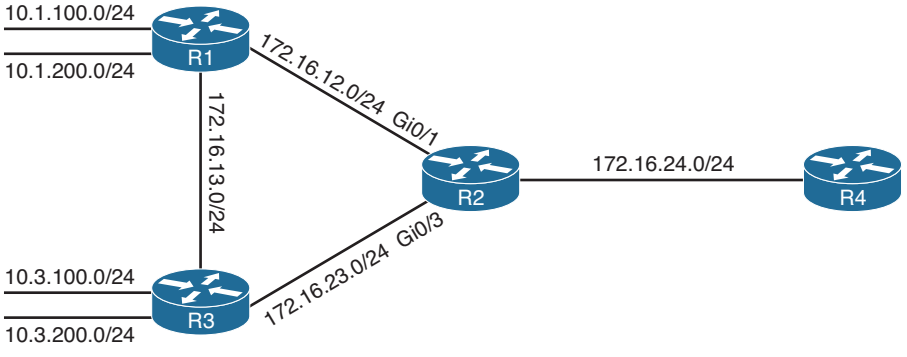

Figure 3-21 *EIGRP Offset List Topology*

Example 3-22 shows the EIGRP routing tables for R2 and R4 before any path metric manipulation is performed.

Example 3-22 *R2 and R4 Routing Tables Before Offset*

```
R2# show ip route eigrp | begin Gateway
Gateway of last resort is not set

      10.0.0.0/24 is subnetted, 4 subnets
D        10.1.100.0 [90/15360] via 172.16.12.1, 00:00:35, GigabitEthernet0/1
D        10.1.200.0 [90/15360] via 172.16.12.1, 00:00:35, GigabitEthernet0/1
D        10.3.100.0 [90/15360] via 172.16.23.3, 00:00:40, GigabitEthernet0/3
D        10.3.200.0 [90/15360] via 172.16.23.3, 00:00:40, GigabitEthernet0/3
      172.16.0.0/16 is variably subnetted, 7 subnets, 2 masks
D        172.16.13.0/24
           [90/15360] via 172.16.23.3, 00:00:42, GigabitEthernet0/3
           [90/15360] via 172.16.12.1, 00:00:42, GigabitEthernet0/1
```

```
R4# show ip route eigrp | b Gateway
Gateway of last resort is not set
      10.0.0.0/24 is subnetted, 4 subnets
D        10.1.100.0 [90/3328] via 172.16.24.2, 01:22:01, GigabitEthernet0/2
D        10.1.200.0 [90/3328] via 172.16.24.2, 01:22:01, GigabitEthernet0/2
D        10.3.100.0 [90/3328] via 172.16.24.2, 01:21:57, GigabitEthernet0/2
D        10.3.200.0 [90/3328] via 172.16.24.2, 01:21:57, GigabitEthernet0/2
      172.16.0.0/16 is variably subnetted, 5 subnets, 2 masks
D        172.16.12.0/24
           [90/3072] via 172.16.24.2, 01:22:01, GigabitEthernet0/2
D        172.16.13.0/24
           [90/3328] via 172.16.24.2, 00:00:34, GigabitEthernet0/2
D        172.16.23.0/24
           [90/3072] via 172.16.24.2, 01:22:01, GigabitEthernet0/2
```

To demonstrate how an offset list is used to steer traffic, the path metric for the 10.1.100.0/24 network is incremented on R2's Gi0/1 interface so that R2 forwards packets toward R3 for that network. In addition, the 10.3.100.0/24 network is incremented on R2's Gi0/1 interface so that R2 forwards packets toward R1 for that network.

Example 3-23 displays the configuration of R2 for classic and named modes.

Example 3-23 *EIGRP Offset List Configuration*

```
R2 (Classic Configuration)
ip access-list standard R1
 permit 10.1.100.0
ip access-list standard R3
 permit 10.3.100.0
!
router eigrp 100
```

```
offset-list R1 in 200000 GigabitEthernet0/1
 offset-list R3 in 200000 GigabitEthernet0/3
```

```
R2 (Named Mode Configuration
ip access-list standard R1
 permit 10.1.100.0
ip access-list standard R3
 permit 10.3.100.0
!
router eigrp EIGRP-NAMED
 address-family ipv4 unicast autonomous-system 100
  topology base
   offset-list R1 in 200000 GigabitEthernet0/1
   offset-list R3 in 200000 GigabitEthernet0/3
```

Example 3-24 shows R2's routing table after the offset list is implemented. Notice how the path metrics and next-hop IP address changed for the 10.1.100.0/24 and 10.3.100.0/24 networks, while the metrics for the other routes remained the same.

Example 3-24 *EIGRP Offset List Verification*

```
R2# show ip route eigrp | begin Gateway
Gateway of last resort is not set

      10.0.0.0/24 is subnetted, 4 subnets
D        10.1.100.0 [90/20480] via 172.16.23.3, 00:05:09, GigabitEthernet0/3
D        10.1.200.0 [90/15360] via 172.16.12.1, 00:05:09, GigabitEthernet0/1
D        10.3.100.0 [90/20480] via 172.16.12.1, 00:05:09, GigabitEthernet0/1
D        10.3.200.0 [90/15360] via 172.16.23.3, 00:05:09, GigabitEthernet0/3
      172.16.0.0/16 is variably subnetted, 7 subnets, 2 masks
D        172.16.13.0/24
           [90/15360] via 172.16.23.3, 00:05:09, GigabitEthernet0/3
           [90/15360] via 172.16.12.1, 00:05:09, GigabitEthernet0/1
```

References in This Chapter

Edgeworth, Brad, Foss, Aaron, and Garza Rios, Ramiro. *IP Routing on Cisco IOS, IOS XE, and IOS XR*. Cisco Press: 2014.

RFC 7838, *Cisco's Enhanced Interior Gateway Routing Protocol (EIGRP)*, D. Savage, J. Ng, S. Moore, D. Slice, P. Paluch, R. White. http://tools.ietf.org/html/rfc7868, May 2016.

Cisco. *Cisco IOS Software Configuration Guides*. http://www.cisco.com.

Exam Preparation Tasks

As mentioned in the section "How to Use This Book" in the Introduction, you have a couple choices for exam preparation: the exercises here, Chapter 24, "Final Preparation," and the exam simulation questions in the Pearson Test Prep software.

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 3-2 lists these key topics and the page number on which each is found.

Table 3-2 Key Topics

Key Topic Element	Description	Page Number
Section	Failure detection and timers	108
Paragraph	Convergence	109
Paragraph	Routes going active	110
Paragraph	Stuck in active	112
Paragraph	Summary routes	115
Paragraph	Summary discard routes	116
Paragraph	Summarization metrics	116
Paragraph	EIGRP stub router	118
Paragraph	EIGRP stub router configuration	120
Figure 3-11	EIGRP stub router constraints	121
Paragraph	EIGRP stub site	122
Figure 3-12	EIGRP stub site feature	123
Paragraph	IP bandwidth percentage	125
Figure 3-13	Split horizon	126
Figure 3-18	EIGRP distribution list filtering logic	129
Paragraph	EIGRP offset lists	132

Complete Tables and Lists from Memory

There are no memory tables in this chapter.

Define Key Terms

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

hello packets, hello timer, hold timer, stuck in active (SIA), summarization, EIGRP stub router, EIGPR stub site router, split horizon, offset list

Use the Command Reference to Check Your Memory

This section includes the most important configuration and verification 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 3-3 with a piece of paper, read the description on the left side, and then see how much of the command you can remember.

The ENARSI 300-410 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, verify, and troubleshoot the topics covered in this chapter.

Table 3-3 Command Reference

Task	Command Syntax

Modify the EIGRP hello interval and hold time per interface	Classic: (EIGRP Process)
	`ip hello-interval eigrp as-number seconds`
	`ip hold-time eigrp as-number seconds`
	
	Named Mode: `af-interface {default | interface-id}`
	`hello-interval seconds`
	`hold-time seconds`

Configure EIGRP network summarization	Classic: (EIGRP Process)
	`ip summary-address eigrp as-number network subnet-mask [leak-map route-map-name]`
	
	Named Mode: `(af-interface interface-id)`
	`summary-address network subnet-mask [leak-map route-map-name]`

Statically set the EIGRP metrics for a specific network summary aggregate	`summary-metric network {/prefix-length | subnet-mask} bandwidth delay reliability load MTU`
Configure an EIGRP router as a stub router	`eigrp stub {connected | receive-only | redistributed | static | summary}`

Configure an EIGRP router as a stub site router	Named Mode: `(af-interface interface-id)`
	`stub-site wan-interface`
	And
	`eigrp stub-site as-number:identifier`

Disable EIGRP split horizon on an interface.	Classic: (EIGRP Process)
	`no ip split-horizon eigrp as-number`
	
	Named Mode: `af-interface {default | interface-id}`
	`no split-horizon`

Filter routes for an EIGRP neighbor	`distribute-list {acl-number | acl-name | prefix prefix-list-name | route-map route-map-name | gateway prefix-list-name} {in | out} [interface-id]`

Task	Command Syntax
Modify/increase path cost for routes	**offset-list** *off-set-value* {*acl-number* | *acl-name*} {**in** | **out**} [*interface-id*]
Display the EIGRP-enabled interfaces	**show ip eigrp interface** [{*interface-id* **[detail]** | **detail**}]
Display the EIGRP topology table	**show ip eigrp topology [all-links]**
Display the IP routing protocol information configured on the router	**show ip protocols**

CHAPTER 4

Troubleshooting EIGRP for IPv4

This chapter covers the following topics:

- **Troubleshooting EIGRP for IPv4 Neighbor Adjacencies:** This section covers the reasons EIGRP for IPv4 neighbor relationships might not be formed and how to identify them.
- **Troubleshooting EIGRP for IPv4 Routes:** This section explores the reasons EIGRP for IPv4 routes might be missing from a router's EIGRP table or routing table and how to determine why they are missing.
- **Troubleshooting Miscellaneous EIGRP for IPv4 Issues:** This section identifies some additional issues you might face while using EIGRP, how to identify them, and how to solve them.
- **EIGRP for IPv4 Trouble Tickets:** This section provides three trouble tickets that demonstrate how to use a structured troubleshooting process to solve a reported problem.

This chapter focuses on troubleshooting EIGRP for IPv4. Chapter 5, "EIGRPv6," covers EIGRP for IPv6 and named EIGRP.

Before any routes can be exchanged between EIGRP routers on the same LAN or across a WAN, an EIGRP neighbor relationship must be formed. Neighbor relationships may not form for many reasons, 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 route-related issues.

In this chapter, you will also learn how to troubleshoot issues related to load balancing, summarization, discontiguous 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 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?' Quiz Questions."

Table 4-1 "Do I Know This Already?" Foundation Topics Section-to-Question Mapping

Foundation Topics Section	Questions
Troubleshooting EIGRP for IPv4 Neighbor Adjacencies	1–4
Troubleshooting EIGRP for IPv4 Routes	5, 6, 8
Troubleshooting Miscellaneous EIGRP for IPv4 Issues	7, 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 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 EIGRP adjacencies with the local router, how long they have been neighbors, and the current sequence numbers of EIGRP packets?
   a. **show ip eigrp interfaces**
   b. **show ip eigrp neighbors**
   c. **show ip route eigrp**
   d. **show ip protocols**
2. Which of the following are reasons EIGRP neighbor relationships might not form? (Choose three.)
   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 routes might be missing in an EIGRP autonomous system? (Choose three.)
   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. You have a DMVPN network that has a hub and three spokes. The spokes are not learning the routes of the other spokes. Of the following options, which is most likely the reason for this?
   a. Split horizon is enabled on the GRE interfaces of the spokes
   b. Split horizon is enabled on the hub's mGRE interface
   c. Split horizon is disabled on the hub's mGRE interface
   d. Split horizon is disabled on the GRE interfaces of the spokes
9. An EIGRP summary route is not showing up on the expected routes in the AS. Which of the following questions should you answer while troubleshooting? (Choose three.)
   a. Did you enable route summarization on the correct interface?
   b. Did you associate the summary route with the correct EIGRP autonomous system?
   c. Did you create the appropriate summary route?
   d. Did you create a route to NULL0?
10. The IP addressing scheme for your routing domain is discontiguous. What command should you use in EIGRP configuration mode to make sure that you do not have any routing issues in your EIGRP autonomous system?
    a. **no auto-summary**
    b. **auto-summary**
    c. **passive-interface**
    d. **network** ***ip_address wildcard_mask***

Foundation Topics

Troubleshooting EIGRP for IPv4 Neighbor Adjacencies

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 command **network 10.1.1.0 0.0.0.255** enables EIGRP on all interfaces with an IP address from 10.1.1.0 through 10.1.1.255. The command **network 10.1.1.65 0.0.0.0** enables the EIGRP process on only the interface with the IP address 10.1.1.65. It seems rather simple, and it is; however, for various reasons, neighbor relationships may not form, and you need to be aware of all of them if you plan on successfully troubleshooting EIGRP-related problems. This section focuses on the reasons EIGRP neighbor relationships might not form and how you can identify them during the troubleshooting process.

To verify EIGRP neighbors, you use the **show ip eigrp neighbors** command. Example 4-1 provides 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, 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 you 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 4-1 *Verifying EIGRP Neighbors with* ***show ip eigrp neighbors***

```
R2# show ip eigrp neighbors
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
```

Key Topic

EIGRP neighbor relationships might not form for a variety of reasons, including the following:

- **Interface is down:** The interface must be up/up.
- **Mismatched autonomous system numbers:** Both routers need to be using the same autonomous system number.
- **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 exactly the same K values.
- **Passive interface:** The passive interface feature suppresses the sending and receiving of hello packets while still allowing the interface's network to be advertised.
- **Different subnets:** The exchange of hello packets must be done on the same subnet; if it isn't, the hello packets are ignored.

- **Authentication:** If authentication is being used, the key ID and key string must match, and the key must be valid (if valid times have been configured).
- **ACLs:** An access control list (ACL) may be 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, neighbor adjacencies could flap.

When an EIGRP neighbor relationship does not form, the neighbor is not listed in the neighbor table. In such a case, you need the assistance of an accurate physical and logical 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 of the reasons listed here. Let's look at them individually.

Interface Is Down

The interface must be up if you plan on forming an EIGRP neighbor adjacency. You can verify the status of an interface with the **show ip interface brief** command. The status should be listed as **up**, and the protocol should be listed as **up**.

Mismatched Autonomous System Numbers

For an EIGRP neighbor relationship to be formed, both routers need to be in the same autonomous system. You specify the autonomous system number when you issue the **router eigrp** *autonomous_system_number* command in global configuration mode. If the two routers are in different autonomous systems, they will not form an EIGRP neighbor relationship. Most EIGRP **show** commands 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 4-2. In this example, you can see that R1 is participating in EIGRP autonomous system 100. Using the *spot-the-difference* troubleshooting method, you can compare the autonomous system value listed to the value on a neighboring router to determine whether they differ.

Key Topic

Example 4-2 *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
```


The output of the **debug eigrp packets** command shown in Example 4-3 indicates 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 Gi0/0 and Gi1/0. 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 4-3 *Sample Output of* ***debug eigrp packets*** *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# l
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# l
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 determine which interfaces are participating in the EIGRP process with the command **show ip eigrp interfaces**. In Example 4-4, for instance, 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 no other routers can be reached out Gi0/0 for this scenario. However, if you expect an EIGRP peer out the interface based on your documentation, you 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 (such as congestion) is preventing the interface from sending the necessary updates to the neighbor.

NOTE Remember that EIGRP passive interfaces do not show up in this output. Therefore, you shouldn't jump to the conclusion that the network command is incorrect or missing if the interface does not show up in this output. It is possible that the interface is passive.

Key Topic

Example 4-4 *Verifying EIGRP Interfaces with* ***show ip eigrp interfaces***

```
R2# show ip eigrp interfaces
EIGRP-IPv4 Interfaces for AS(100)
                        Xmit Queue   Mean   Pacing Time   Multicast    Pending
Interface     Peers     Un/Reliable  SRTT   Un/Reliable   Flow Timer   Routes
Gi0/0           0           0/0       0         0/0           0          0
Gi1/0           1           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. The reason it's not obvious is that it's not displayed properly. Focus on the highlighted text in Example 4-5. Notice that it states *Routing for Networks*. Those are *not* the networks you are routing for. Rather, you are routing for the networks associated with the interface on which EIGRP will be enabled, 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 to use the **show run | section router eigrp** command, as displayed in Example 4-6.

Example 4-5 *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 4-6 *Verifying* ***network*** *Statements with* **show run | section router eigrp**

```
R1# show run | section router eigrp
router eigrp 100
 network 10.1.1.1 0.0.0.0
 network 10.1.12.1 0.0.0.0
```

Notice that 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 packets** command on the router with the misconfigured or missing **network** statement, you will notice that hello packets are not being sent out the interface properly. For example, if you expect hello packets to be sent out Gig1/0, but the **debug eigrp packets** command is not indicating that this is happening, it is possible that the interface is not participating in the EIGRP process because of a bad **network** statement or the interface is passive and suppressing hello packets.

Mismatched K Values

The K values that are used for metric calculation must match between neighbors in order for an adjacency to form. You can verify whether K values match by using **show ip protocols**, as shown in Example 4-7. The default K values are highlighted in Example 4-7. Usually there is no need to change the K values. However, if they are changed, you need to 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 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
```

Key Topic

Example 4-7 *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 interface's 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 in Example 4-8, 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.

Key Topic

Example 4-8 *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 that for EIGRP, passive interfaces do not appear in the EIGRP interface table. Therefore, before you jump to the conclusion that the wrong network command was used and the interface was not enabled for EIGRP, you need to check to see whether the interface is passive.

When using the **debug eigrp packets** command on the router with the passive interface, 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 packets** command is not indicating so, it is possible that the 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. You can also use the **show ip interface** *interface_type interface_number* command or the **show interface** *interface_type interface_number* command. Example 4-9 shows 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 are both in the 10.1.12.0/24 subnet. However, if they are not in the same subnet and you have syslog set up for a severity level of 6, you get a message similar to the following:

```
%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 4-9 *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
```

Key Topic

Authentication

Authentication is used to ensure that EIGRP routers form neighbor relationships only 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 4-10 shows the output of the commands **show run interface** *interface_type interface_number* and **show ip eigrp interfaces detail** *interface_type interface_number*, which identify whether EIGRP authentication is enabled on the interface. According to the highlighted text, it is. Note that the authentication must be configured 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 keychain that will be used for the Message Digest 5 (MD5) authentication hash. You can verify the keychain with the command **show key chain**, as shown in Example 4-11. 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 4-10 *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 4-11 *Verifying the Keychain Used for EIGRP Authentication*

```
R1# show key chain
Key-chain EIGRP_AUTH:
    key 1 -- text "ENARSI"
        accept lifetime (always valid) - (always valid) [valid now]
        send lifetime (always valid) - (always valid) [valid now]
```

Inside the keychain, you find the key ID (1 in this case) and the key string (ENARSI 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 receive output based on the authentication issue. Example 4-12 shows the message that 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 4-13.

Example 4-12 *Debug Output When Authentication Is Missing on the 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 4-13 *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 (ACLs) are extremely powerful. How they are implemented determines what they are controlling in a network. If there is an ACL applied to an interface and the ACL is denying EIGRP packets, or if an EIGRP packet falls victim to the implicit deny all at the end of the ACL, a neighbor relationship does 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 4-14. Notice that ACL 100 is applied inbound on interface Gig1/0. To verify the ACL 100 entries, issue the command **show access-lists 100**, as shown in Example 4-15. In this case, you can see that ACL 100 is denying EIGRP traffic; this prevents a neighbor relationship from forming. Note that outbound ACLs do not affect EIGRP packets; only inbound ACLs do. Therefore, any outbound ACLs that deny EIGRP packets have no effect on your EIGRP troubleshooting efforts.

Example 4-14 *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 4-15 *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, an 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; this terminates the neighbor relationship. Five seconds later, the hello packet arrives, and the neighbor relationship is formed, but it is then terminated again 15 seconds later.

Although timers do not have to match, it is important that routers send hello packets at a rate that is faster than the hold timer. You verify the configured timers with the **show ip eigrp interfaces detail** command, as shown in Example 4-10.

Troubleshooting EIGRP for IPv4 Routes

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 EIGRP routes might be missing from either the topology table or the routing table, and you need to be aware of them if you plan on successfully troubleshooting EIGRP route-related problems. This section examines the reasons EIGRP routes might be missing and how to determine why they are missing.

EIGRP only learns from directly connected neighbors, which makes 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 probably something wrong between the neighbors. However, if the neighbor does not know about it either, you can focus on the neighbor's neighbor and so on.

As discussed earlier, neighbor relationships are the foundation of EIGRP information sharing. If there are no neighbors, you do not learn any routes. So, besides the lack of a neighbor, what would be reasons for missing routes in an EIGRP network? The following are some common reasons EIGRP routes might be missing either from 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 prefix of the network the interface is part of into the EIGRP process.
- **Better source of information:** If exactly the same network prefix is learned from a more reliable source, it is used instead of the EIGRP learned information.
- **Route filtering:** A filter might be preventing a network prefix from being advertised or learned.
- **Stub configuration:** If the wrong setting is chosen during the stub router configuration, or if the wrong router is chosen as the stub router, it might prevent a network prefix 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:** Split horizon is a loop-prevention feature that prevents a router from advertising routes out the same interface on which they were learned.

This section looks at each of these reasons individually and explores 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 do not form neighbor relationships with other routers need a valid **network** statement that enables EIGRP on those interfaces so the networks the interfaces belong to are injected into the EIGRP process and advertised. If the **network** statement is missing or configured incorrectly, EIGRP is not enabled on the interface, and the network the interface belongs to is never advertised and is therefore unreachable by other routers.

As discussed earlier in this chapter, the output of **show ip protocols** displays the **network** statements in a nonintuitive way. Focus on the highlighted text in Example 4-16. Notice that it states *Routing for Networks*. Those are *not* the networks you are routing for. You are routing for the networks associated with the interface on which EIGRP will be enabled, based on the **network** statement. 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**.

Example 4-16 *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 you actually routing for then? You are routing for the networks associated with the interfaces that are now enabled for EIGRP. In Example 4-17, you can see the output of the **show ip interface** command on R1 for Gig0/0 and Gig1/0, which was piped to include only the **Internet** address. Notice that that these two interfaces 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 you are routing for.

Example 4-17 *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 gi1/0 | 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 by using the **show ip eigrp interfaces** command, as shown earlier in Example 4-4.

Better Source of Information

For an EIGRP-learned route to be installed in the routing table, it must be the most trusted routing source. Recall that the trustworthiness of a source 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 exactly the same network and that source has a better AD, the source with the better AD wins, and its information is installed in the routing table. Compare Example 4-18, which is an EIGRP topology table, and Example 4-19, 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 4-18 *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 4-19 *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 4-18 appear in the routing table as EIGRP routes. In this case, there is a better source for the same information. Example 4-20, 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 to Example 4-18 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, and the static default route would be.

Example 4-20 *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

Using a suboptimal 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. However, it may cause suboptimal routing in the network. Figure 4-1 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.

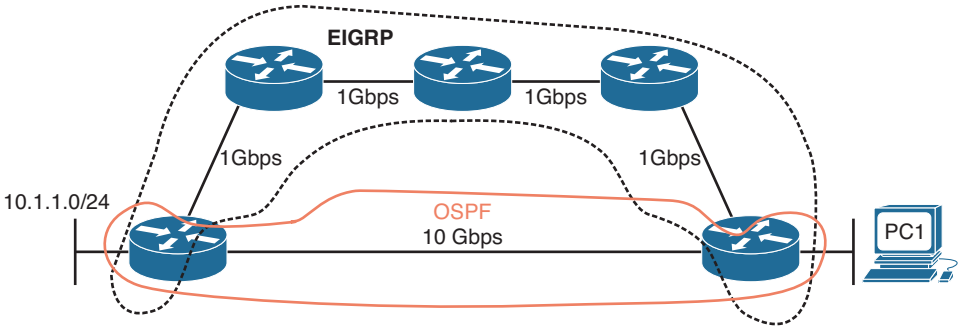

Figure 4-1 *Using the Suboptimal EIGRP Path*

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 being perceived as slow. In this case, you 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 and 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:

Key Topic

- 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 to an individual interface, as shown in Example 4-21. This example indicates that there are no outbound filters and that there is an inbound filter on Gig1/0.

Example 4-21 *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 4-21 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 shown in Example 4-22, you verify the command that was used to apply the distribute list in the running configuration by reviewing the EIGRP configuration section.

Example 4-22 *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
```

Key Topic

Stub Configuration

The EIGRP stub feature allows you to control the scope of EIGRP queries in the network. Figure 4-2 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 to be 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 and slows convergence. As shown in Figure 4-3, configuring the EIGRP stub feature on R3 with the **eigrp stub** command ensures that R2 never sends a query to R3.

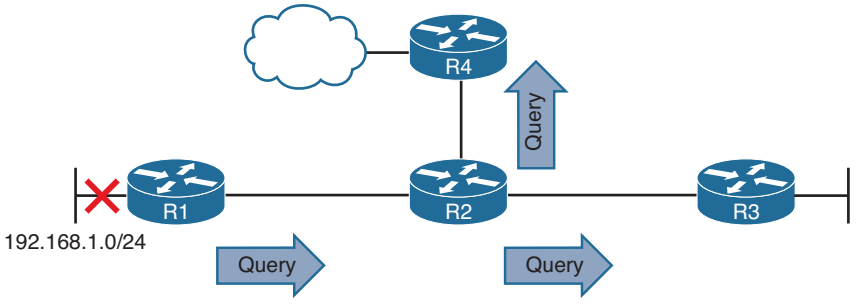

Figure 4-2 *Query Scope Without the EIGRP Stub Feature*

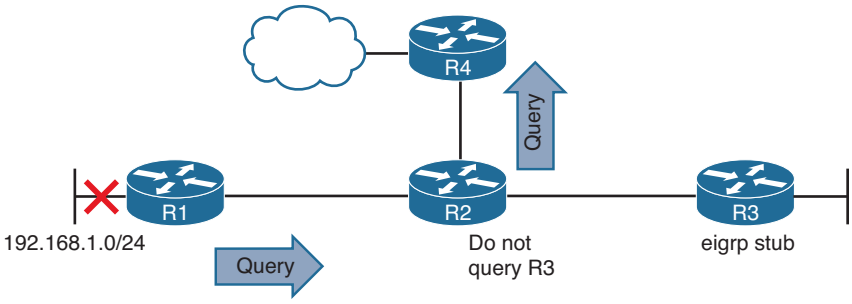

Figure 4-3 *Query Scope with the EIGRP Stub Feature*

This feature comes in handy over slow hub-and-spoke WAN links, as shown in Figure 4-4. The stub feature 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 it can result in the reestablishment of neighbor relationships, causing convergence and generating even more EIGRP traffic. Therefore, if you do not query the hubs, you do not have to worry about these issues.

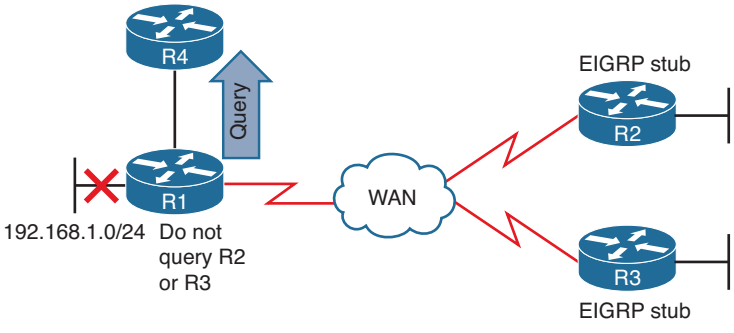

Figure 4-4 *EIGRP Stub Feature over WAN Links*

When configuring the EIGRP stub feature, you can control what routes the stub router advertises to its neighbor. By default, it advertises connected and summary routes. However, you have the option of advertising connected, summary, redistributed, or static—or a combination of these. The other option is to send no routes (called *receive only*). If the wrong option is chosen, the stub routers do not advertise the correct routes to their 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 4-4), R1 never fully shares all routes it knows about to R4, R2, and R3, resulting in missing routes in the topology. To verify whether a router is a stub router and determine the routes it will advertise, issue the **show ip protocols** command, as shown in Example 4-23.

Example 4-23 *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 4-24 shows the output of **show ip eigrp neighbors detail** on R1, which indicates that the neighbor is a stub router advertising connected and summary routes and suppressing queries.

Example 4-24 *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               Se1/0             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 (that is, 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 must 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 4-5 shows a nonbroadcast

multi-access (NBMA) Frame Relay hub-and-spoke topology or a Dynamic Multipoint Virtual Private Network (DMVPN) network, which both use multipoint interfaces on the hub. The multipoint interface (a single physical interface or a mGRE tunnel interface) provides connectivity to multiple routers in the same subnet out the single interface, as does 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 Se1/0 interface or the multipoint GRE tunnel interface on R1, R1 does not advertise the 10.1.2.0/24 network back out that interface. Therefore, R3 never learns 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 4-25. In this case, you can see that split horizon is enabled.

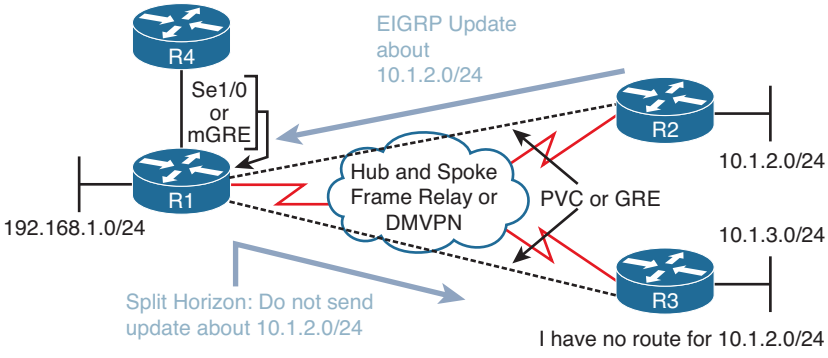

Figure 4-5 *EIGRP Split Horizon Issue*

Example 4-25 *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 completely disable split horizon on an interface, 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 still shows as enabled in the output of **show ip interface** (refer to Example 4-25). 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 4-26 shows that it is disabled for EIGRP on interface tunnel 0.

Example 4-26 *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   Mean   Pacing Time   Multicast    Pending
Interface        Peers  Un/Reliable  SRTT   Un/Reliable   Flow Timer   Routes
Tu0                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 in this chapter, the focus has been on troubleshooting EIGRP neighbor relationships and routes. In this section, the focus is on troubleshooting issues related to feasible successors, discontiguous networks and autosummarization, route summarization, and equal- and unequal-metric load balancing.

Feasible Successors

The best route (based on the 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. (The term *candidate* is used because even though it is the best EIGRP route, a better source of the same information might 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 4-27 shows 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 path is being used as the best path. It is the one with the lowest FD, 2174976, which is the path through 172.16.33.5, reachable out interface Serial 1/0.

Example 4-27 *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 10.1.13.0/24, 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):

- **Feasible distance:** The RD plus the metric to reach the neighbor at the next-hop address that is advertising the RD
- **Reported distance:** The distance from the neighbor at the next-hop address to the destination network

The successor is the path with the lowest FD. However, EIGRP also pre-calculates 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 4-27. The path through 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 4-28 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. In the output of **show ip eigrp topology**, notice that there is only one path listed; in the output of **show ip eigrp topology all-links**, notice that there are two paths listed. This is because the next hop 172.16.33.13 has an RD greater than the FD of the successor and therefore cannot be a feasible successor.

Example 4-28 *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 contains not only the routes learned from other routers but also routes that have been redistributed into the EIGRP process and the local connected networks whose interfaces are participating in the EIGRP process, as highlighted in Example 4-29.

Example 4-29 *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 Cisco IOS (before release 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 by using the **no auto-summary** command in router configuration mode for an EIGRP autonomous system. However, from Cisco IOS 15.0 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 auto-summarization in your EIGRP autonomous system, routing would be broken.

Figure 4-6 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, you would see an entry for 172.16.0.0 with two next hops (if everything else is equal): one through R3 using Fa0/1 and the other through R1 using Fa0/0.

Now picture a packet arriving at R2 from R4 with the destination IP address 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 discontiguous 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 4-30.

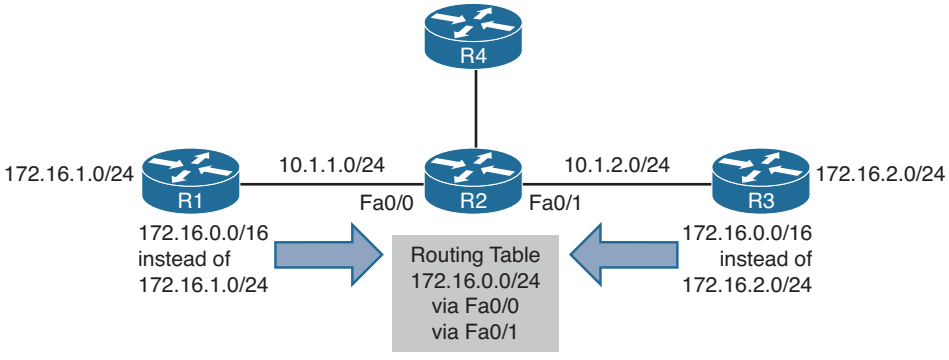

Figure 4-6 *Discontiguous Network Example*

Example 4-30 *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 (which is not recommended) or perform manual route summarization (which is recommended). With EIGRP, manual route summarization is enabled on an interface-by-interface basis. Therefore, when troubleshooting route summarization, keep in mind the following:

- 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 determine answers to all these questions by using the **show ip protocols** command, as shown in Example 4-30. In this example, autosummarization is disabled, and manual summarization is enabled for EIGRP autonomous system 100 on interface Gi2/0 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 the following snippet:

```
Router# show ip route | include Null

D 10.1.0.0/20 is a summary, 00:12:03, Null0
```

This route to null 0 is created to prevent routing loops. It is imperative that this route exists in the table. It ensures that when a packet is received by the router with a destination address that falls within the summary, the packet 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 using the default route. The next-hop router would then end up forwarding the packet back to this router because it is using the summary route. The local router would then forward it based on the default route again, and then it would come back. This is a routing loop.

The route to null 0 has an AD of 5, as shown in the following snippet, to ensure that it is more trustworthy than most of the other sources of routing information:

```
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
```

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, if someone created a static route for the same summary network and pointed it to a next-hop IP address instead of null 0). This would cause a routing loop.

Load Balancing

By default, EIGRP load balances 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, which means the load balancing will occur only over equal-metric paths. You 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 has a metric of 200000, and you configure the **variance 2** command for the EIGRP routing process. This causes load balancing to occur over any route with a metric in the range of 200000 through 400000 (that is, 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 configure the correct variance multiplier, but **maximum-paths** is set to 2, you use only two of the five paths. To use all five, you would also need to make sure that **maximum-paths** is set to 5.

Also, remember that the feasibility condition plays a huge role in unequal-path load balancing to prevent routing loops. If the path is not a feasible successor, it cannot be used for unequal-path load balancing. There is no exception to this rule. Recall the feasibility condition: *To be a feasible successor, the RD must be less than the FD of the successor.*

To verify the configured maximum paths and variance, you use the **show ip protocols** command, as shown in Example 4-31.

Key Topic

Example 4-31 *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
  10.1.12.2        90    10:26:36
Distance: internal 90 external 170
```

EIGRP for IPv4 Trouble Tickets

This section presents various trouble tickets related to the topics discussed earlier in the chapter. The purpose of these trouble tickets is to show 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 shown in Figure 4-7.

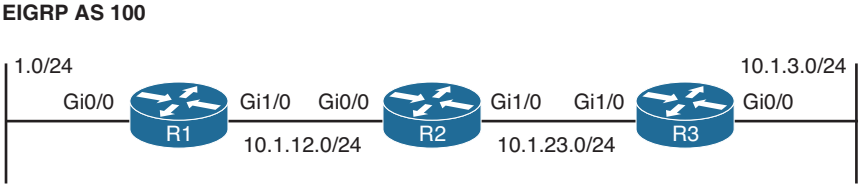

Figure 4-7 *EIGRP for IPv4 Trouble Tickets Topology*

Trouble Ticket 4-1

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 to 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 4-32. 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 4-32 *Destination Unreachable Result from the* ***ping*** *Command on a 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 you 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, you can focus your attention on R1 and work from there.

On R1, you issue the same ping, but it fails, as shown in Example 4-33.

Example 4-33 *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 4-34. You conclude that R1 is not learning any routes from R2.

Example 4-34 *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 4-7, EIGRP is the routing protocol in use. Therefore, you issue the **show ip protocols** command to verify that EIGRP is using the correct autonomous system number. Example 4-35 displays the **show ip protocols** output, which confirms that EIGRP 100 is in operation on R1.

Example 4-35 *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 the following snippet:

```
R1# show ip eigrp neighbors

EIGRP-IPv4 Neighbors for AS(100)
```

According to the output, R1 has no neighbors.

Next, you verify whether there are any interfaces participating in the EIGRP process by using the **show ip eigrp interfaces** command. Example 4-36 indicates that there are two interfaces participating in the EIGRP process: Gi0/0 and Gi1/0.

Example 4-36 *show ip eigrp interfaces Output on R1*

```
R1# show ip eigrp interfaces
EIGRP-IPv4 Interfaces for AS(100)
                        Xmit Queue   Mean   Pacing Time   Multicast    Pending
Interface         Peers  Un/Reliable  SRTT   Un/Reliable   Flow Timer   Routes
Gi0/0               0       0/0        0       0/0           0           0
Gi1/0               0       0/0        0       0/0           304         0
```

The output of **show cdp neighbors**, as shown in Example 4-37, indicates that R1 is connected to R2 using Gig 1/0 and that R2 is using Gig 0/0. Therefore, you expect a peering between the two, using these interfaces.

Example 4-37 *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 Gi0/0 on R2 is participating in the EIGRP process. On R2, you issue the **show ip eigrp interfaces** command, as shown in Example 4-38.

Example 4-38 *show ip eigrp interfaces Output on R2*

```
R2# show ip eigrp interfaces
EIGRP-IPv4 Interfaces for AS(100)
                        Xmit Queue   Mean   Pacing Time   Multicast    Pending
Interface        Peers  Un/Reliable  SRTT   Un/Reliable   Flow Timer   Routes
Gi1/0              0      0/0        0        0/0          448          0
```

Example 4-38 confirms that R2's interface Gi0/0 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 4-39, 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**, based on the IP address 10.1.12.2 of interface GigabitEthernet0/0.

Example 4-39 *show run | section router eigrp Output on R2 and Verifying the 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
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 4-40.

Example 4-40 *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 that the problem is solved, and you receive the same result, as shown in Example 4-41. R1 still does not know about the 10.1.3.0/24 network.

Example 4-41 *Destination Unreachable from the* ***ping*** *Command on a 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 4-42. R1 is receiving EIGRP routes because there is now an EIGRP route in the routing table (as indicated by D). However, R1 still does not know about the 10.1.3.0/24 network.

Example 4-42 *show ip route Output After the Neighbor Relationship with R2 Is 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 4-43 shows R2's routing table, which is missing 10.1.3.0/24 as well.

Example 4-43 *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. The R2 output of **show ip eigrp neighbors** in Example 4-44 indicates that R3 is not a neighbor; only R1 is.

Example 4-44 *show ip eigrp neighbors on R2*

```
R2# show ip eigrp neighbors
EIGRP-IPv4 Neighbors for AS(100)
H   Address                 Interface       Hold   Uptime    SRTT   RTO  Q  Seq
                                            (sec)             (ms)       Cnt Num
0   10.1.12.1               Gi0/0            11    00:17:28   65    390  0   7
```

Previously, Example 4-38 indicated that Gig1/0 on R2 is participating in the EIGRP process. Therefore, you should look at the interfaces on R3. According to the output in Example 4-45, both interfaces on R3 are participating in the EIGRP process for autonomous system 10.

Example 4-45 *show ip eigrp interfaces on R3*

```
R3# show ip eigrp interfaces
EIGRP-IPv4 Interfaces for AS(10)
                        Xmit Queue   Mean   Pacing Time   Multicast    Pending
Interface         Peers Un/Reliable  SRTT   Un/Reliable   Flow Timer   Routes
Gi0/0                 0       0/0       0        0/0           0          0
Gi1/0                 0       0/0       0        0/0           0          0
```

Can you see the issue? If not, look again at Example 4-45. If you need to compare it to Example 4-44, 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 configurations that are not needed, such as the EIGRP autonomous system 10 configurations. Example 4-46 shows the commands needed to accomplish this.

Example 4-46 *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 4-46 that the neighbor relationship with R2 is now successful. Now it is time to verify that all the issues have been solved. On R2, you issue the **show ip route** command, as shown in Example 4-47, 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 4-48. You then ping from the PC again, and the ping is truly successful, as shown in Example 4-49.

Example 4-47 *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 4-48 *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 4-49 *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 4-2

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 4-50, 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 is technically not successful.

Example 4-50 *Destination Unreachable Result from the* ***ping*** *Command on a 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 you 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, you can focus your attention on R1 and work from there.

On R1, you issue the same ping, but it fails, as shown in Example 4-51.

Example 4-51 *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 the following snippet:

```
R1# show ip route 10.1.3.0 255.255.255.0
```

This is the result:

```
% Subnet not in table
```

Does R2 know about it? You go to R2 and issue the same command, as shown in the following snippet:

```
R2# show ip route 10.1.3.0 255.255.255.0
```

The result is the same as on R1:

```
% 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 4-52.

Example 4-52 *Determining Whether a 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 prevents a connected route from being advertised using EIGRP to a neighbor? As we learned earlier, the interface not participating in the EIGRP process. You can check the EIGRP interface table on R3 with the **show ip eigrp interfaces** command. Example 4-53 indicates that only Gi1/0 is participating in the EIGRP process.

Example 4-53 *Determining Whether an Interface Is Participating in the EIGRP Process*

```
R3# show ip eigrp interfaces
EIGRP-IPv4 Interfaces for AS(100)
                        Xmit Queue   Mean   Pacing Time   Multicast    Pending
Interface        Peers  Un/Reliable  SRTT   Un/Reliable   Flow Timer   Routes
Gi1/0              1        0/0      821       0/0           4080         0
```

However, you should not jump to the conclusion that Gi0/0 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 4-54, you can see that there are no passive interfaces.

Example 4-54 *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
```


Next, 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 4-54, the output of **show ip protocols** indicates that R3 is routing for the network 10.1.3.0/32. Remember from earlier in this chapter that this really means network 10.1.3.0 0.0.0.0. As a result, EIGRP is enabled on the interface with the IP address 10.1.3.0. Example 4-55, which displays the output of **show ip interface brief**, shows that there are no interfaces with that IP address. Interface GigabitEthernet0/0 has the IP address 10.1.3.3. Therefore, the **network** statement is incorrect, as shown in the output of **show run | section router eigrp** in Example 4-56.

Example 4-55 *Reviewing the Interface IP Addresses*

```
R3# show ip interface brief
Interface          IP-Address OK? Method Status Protocol
GigabitEthernet0/0 10.1.3.3   YES NVRAM  up     up
GigabitEthernet1/0 10.1.23.3  YES NVRAM  up     up
```

Example 4-56 *Reviewing the **network** Statements in the Running Configuration*

```
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 4-57, 10.1.3.0/24 is now in the routing table and can be reached using the next hop 10.1.12.2.

Example 4-57 *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 4-58.

Example 4-58 *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 4-3

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 4-59, it fails. Notice that the reply is from the default gateway at 10.1.1.1, and it states Destination host unreachable.

Example 4-59 *Destination Unreachable Result from the* ***ping*** *Command on a 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 you 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, you can focus your attention on R1 and work from there.

On R1, you issue the same ping, but it fails, as shown in Example 4-60.

Example 4-60 *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 the following configuration:

```
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 4-61. R2 does know about it.

Example 4-61 *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 4-62 indicates that it is not.

Example 4-62 *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
```

It's time to hypothesize! Why would R2 know about 10.1.3.0/24 and R1 not know about it? Consider these possibilities:

- 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 4-63, and it shows that R2 is a neighbor. However, if you look closely at the topology table of R1, you might notice that R1 is learning about 10.1.23.0/24 from R2, meaning that they are neighbors, and routes are being learned. Therefore, you hypothesize that there must be a filter in place.

Example 4-63 *Determining Whether R2 Is a Neighbor*

```
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        12     01:20:27     72    432  0  18
```

Next, you issue the **show ip protocols** command, as shown in Example 4-64, to determine whether there are any route filters on R1. The output indicates that there is an inbound route filter on R1's GigabitEthernet 1/0 interface. The route filter is filtering based on a prefix list called DENY_10.1.3.0/24.

Example 4-64 *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...
```

Next, you issue the **show ip prefix-list** command on R1, as shown in Example 4-65, and it indicates that 10.1.3.0/24 is being denied.

Example 4-65 *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. The choice depends on the requirements of the organization or scenario. In this case, remove the distribute list from R1 with the command **no distribute-list prefix DENY_10.1.3.0/24 in GigabitEthernet1/0**. 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 4-66, 10.1.3.0/24 is now in the routing table and can be reached through the next hop 10.1.12.2.

Example 4-66 *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 4-67.

Example 4-67 *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
```

Exam Preparation Tasks

As mentioned in the section "How to Use This Book" in the Introduction, you have a couple choices for exam preparation: the exercises here, Chapter 24, "Final Preparation," and the exam simulation questions in the Pearson Test Prep software. The questions that follow present a bigger challenge than the exam itself because they use an open-ended question format. By using this more difficult format, you can exercise your memory better and prove your conceptual and factual knowledge of this chapter. You can find the answers to these questions in the appendix.

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 these key topics and the page number on which each is found.

Table 4-2 Key Topics

Key Topic Element	Description	Page Number
List	Possible reasons an EIGRP neighbor relationship might not form	141
Example 4-2	Verifying the autonomous system number with **show ip protocols**	142
Example 4-4	Verifying EIGRP interfaces with **show ip eigrp interfaces**	144
Example 4-7	Verifying K values with **show ip protocols**	146
Example 4-8	Verifying passive interfaces with **show ip protocols**	147
Section	Authentication	148
List	Possible reasons EIGRP for IPv4 routes may be missing from the routing table	152
Paragraph	How a better source of routing information could cause suboptimal routing	156
List	Considerations when troubleshooting route filters	157
Section	Stub configuration	158
Section	Split horizon	160
List	Considerations when troubleshooting route summarization	167
Example 4-31	Verifying variance and maximum paths	168

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 value, passive interface, key ID, key string, keychain, stub, split horizon, successor, feasible successor, reported distance, feasible distance, discontiguous network, autosummarization, classful, classless, maximum paths, variance

Use the Command Reference to Check Your Memory

This section includes the most important 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 in Table 4-3, go to the companion web site and download the Command Reference Exercises document. Fill in the missing command in the tables based on the command description. You can check your work by downloading the Command Reference Exercise Answer Key Appendix also on the companion website.

The ENARSI 300-410 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, verify, and troubleshoot the topics covered in this chapter.

Table 4-3 Command Reference

Task	Command Syntax
Display the IPv4 routing protocols enabled on the router; for EIGRP, display 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**
Show a router's EIGRP neighbors	**show ip eigrp neighbors**
Show 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	**show ip eigrp neighbors detail**
Display all of a router's interfaces that are configured to participate in an EIGRP routing process (with the exception of passive interfaces)	**show ip eigrp interfaces**
Display 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	**show ip eigrp interfaces detail**
Display the EIGRP configuration in the running configuration	**show run | section router eigrp**
Display the configuration of a specific interface in the running configuration (This is valuable when you are trying to troubleshoot EIGRP interface commands.)	**show run interface** *interface_type interface_number*
Display the keychains and associated keys and key strings	**show key chain**
Display IPv4 interface parameters; for EIGRP, verify whether the interface has joined the correct multicast group (224.0.0.10) and whether any ACLs applied to the interface might be preventing an EIGRP adjacency from forming	**show ip interface** *interface_type interface_number*

Task	Command Syntax
Display routes known to a router's EIGRP routing process, which 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.)	**show ip eigrp topology** [**all-links**]
Show routes known to a router's IP routing table that were injected by the router's EIGRP routing process	**show ip route eigrp**
Display all EIGRP packets exchanged with a router's EIGRP neighbors or display only specific EIGRP packet types (for example, EIGRP hello packets)	**debug eigrp packets**

CHAPTER 5

EIGRPv6

This chapter covers the following topics:

- **EIGRPv6 Fundamentals:** This section provides an overview of EIGRPv6 and the correlation to EIGRP for routing IPv4 networks.
- **Troubleshooting EIGRPv6 Neighbor Issues:** This section discusses the reasons EIGRPv6 neighbor relationships may not be formed and how to identify them.
- **Troubleshooting EIGRPv6 Routes:** This section explores the reasons EIGRPv6 routes might be missing and how to determine why they are missing.
- **Troubleshooting Named EIGRP:** This section introduces the **show** commands that you can use to troubleshoot named EIGRP configurations.
- **EIGRPv6 and Named EIGRP Trouble Tickets:** This section provides trouble tickets that demonstrate how to use a structured troubleshooting process to solve a reported problem.

The original EIGRP routing protocol supports multiple protocol suites. Protocol-dependent modules (PDMs) provide unique neighbor and topology tables for each protocol. When the IPv6 address family is enabled, the routing protocol is commonly referred to as EIGRPv6.

This chapter reviews the fundamentals of EIGRPv6 and guides you through configuring and verification. In addition, it examines how to troubleshoot common EIGRPv6 neighbor and route issues. It also explores named EIGRP and wraps up by providing a look at two trouble tickets.

"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?' Quiz Questions."

Table 5-1 "Do I Know This Already?" Foundation Topics Section-to-Question Mapping

Foundation Topics Section	Questions
EIGRPv6 Fundamentals	1–3
Troubleshooting EIGRPv6 Neighbor Issues	5, 9
Troubleshooting EIGRPv6 Routes	6, 7
Troubleshooting Named EIGRP	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 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 address does the EIGRPv6 hello packet use for the destination address?
   a. MAC address 00:C1:00:5C:00:FF
   b. MAC address E0:00:00:06:00:AA
   c. IP address 224.0.0.8
   d. IP address 224.0.0.10
   e. IPv6 address FF02::A
   f. IPv6 address FF02::8
2. Enabling EIGRPv6 on an interface with EIGRPv6 classic configuration requires _____.
   a. the command **network** *prefix*/*prefix-length* under the EIGRP process
   b. the command **network** *interface-id* under the EIGRP process
   c. the command **ipv6 eigrp** *as-number* under the interface
   d. nothing; EIGRPv6 is enabled on all IPv6 interfaces upon initialization of the EIGRP process
3. Enabling EIGRPv6 on an interface with EIGRPv6 named mode configuration requires _____.
   a. the command **network** *prefix*/*prefix-length* under the EIGRP process
   b. the command **network** *interface-id* under the EIGRP process
   c. the command **ipv6 eigrp** *as-number* under the interface
   d. nothing; EIGRPv6 is enabled on all IPv6 interfaces upon initialization of the EIGRP process
4. Which EIGRPv6 command is used to verify whether any interfaces have been configured as passive interfaces?
   a. **show ipv6 protocols**
   b. **show ipv6 eigrp interfaces detail**
   c. **show ipv6 eigrp neighbors detail**
   d. **show ipv6 eigrp topology**

5. Which EIGRPv6 command enables you to verify whether the local router is a stub router?
   a. **show ipv6 protocols**
   b. **show ipv6 eigrp interfaces detail**
   c. **show ipv6 eigrp neighbors detail**
   d. **show ipv6 eigrp topology**
6. Which EIGRPv6 command enables you to verify whether a neighboring router is a stub router?
   a. **show ipv6 protocols**
   b. **show ipv6 eigrp interfaces detail**
   c. **show ipv6 eigrp neighbors detail**
   d. **show ipv6 eigrp topology**
7. Which of these commands can you use to verify which interfaces are participating in the named EIGRP IPv4 address family? (Choose two.)
   a. **show ip eigrp interfaces**
   b. **show eigrp address-family ipv4 interfaces**
   c. **show ipv6 eigrp interfaces**
   d. **show eigrp address-family ipv6 interfaces**
8. Which of the following must match to form an EIGRPv6 neighborship? (Choose two.)
   a. The subnet the interfaces belong to
   b. The autonomous system number
   c. The passive interfaces
   d. The K values
9. What must be permitted within an IPv6 ACL for an EIGRPv6 neighbor adjacency to be formed?
   a. FF02::A
   b. FF02::10
   c. The link-local address of the neighboring device
   d. The global address of the neighboring device

Foundation Topics

EIGRPv6 Fundamentals

EIGRP's functional behavior is unchanged between IPv4 and IPv6. The same administrative distance, metrics, timers, and DUAL mechanisms are in place to build the routing table. This chapter provides a detailed overview of the EIGRP protocol operation along with its common features. This section is devoted to discussing the components of the routing protocol that are unique to IPv6.

EIGRPv6 Inter-Router Communication

EIGRP packets are identified using the well-known protocol ID 88 for both IPv4 and IPv6. When EIGRPv6 is enabled, the routers communicate with each other using the interface's IPv6 link-local address as the source, and depending on the EIGRP packet type, the destination address may be either a unicast link-local address or the multicast link-local scoped address FF02::A.

Table 5-2 shows the source and destination addresses for the EIGRP packet types.

Key Topic

Table 5-2 EIGRPv6 Packets

EIGRP Packet	Source	Destination	Purpose
Hello	Link-local address	FF02::A	Neighbor discovery and keepalive
Acknowledgment	Link-local address	Link-Local address	Acknowledges receipt of an update
Query	Link-local address	FF02::A	Request for route information during a topology change event
Reply	Link-local address	Link-Local address	A response to a query message
Update	Link-local address	Link-Local address	Adjacency forming
Update	Link-local address	FF02::A	Topology change

5

EIGRPv6 Configuration

There are two methods for configuring IPv6 for EIGRP on IOS and IOS XE routers:

- Classic AS mode
- Named mode

EIGRPv6 Classic Mode Configuration

Classic mode is the original IOS method for enabling IPv6 on EIGRP. In this mode, the routing process is configured using an autonomous system number.

Key Topic

The steps for configuring EIGRPv6 on an IOS router are as follows:

Step 1. Configure the EIGRPv6 process by using the global configuration command **ipv6 router eigrp** *as-number*.

Step 2. Assign the router ID by using the IPv6 address family command **eigrp router-id** *id*. The router ID should be manually assigned to ensure proper operation of the routing process. The default behavior for EIGRP is to locally assign a router ID based on the highest IPv4 loopback address or, if that is not available, the highest IPv4 address. The router ID does not need to map to an IPv4 address; the ID value could be any 32-bit unique dotted-decimal identifier. If an IPv4 address is not defined or if the router ID is not manually configured, the routing process does not initiate.

Step 3. Enable the process on the interface by using the interface parameter command **ipv6 eigrp** *as-number*.

Nearly all EIGRP IPv6 features are configured in the same manner in IPv4 EIGRP classic mode. The primary difference is that the **ipv6** keyword precedes most of the commands in lieu of the **ip** keyword. One noticeable exception is the familiar IPv4 network statement in the EIGRP routing configuration mode. The network statement does not exist within EIGRPv6. The protocol must be enabled directly on the interface when using the classic IPv6 EIGRP AS configuration method.

EIGRPv6 Named Mode Configuration

EIGRP named mode configuration is a newer method for configuring the protocol on IOS routers. Named mode provides support for IPv4, IPv6, and virtual routing and forwarding (VRF), all within a single EIGRP instance.

The steps for configuring EIGRP named mode are as follows:

Step 1. Configure the EIGRPv6 routing process in global configuration mode by using the command **router eigrp** *process-name*. Unlike in classic mode, you specify a name instead of an autonomous system number.

Step 2. Define the address family and autonomous system number (ASN) to the routing process by using the command **address-family ipv6 autonomous-system** *as-number*.

Step 3. Assign the router ID by using the IPv6 address family command **eigrp router-id** *router-id*.

EIGRP named mode uses a hierarchical configuration. Most of the command structure is identical to that of EIGRP IPv4 named mode; this mode simplifies configuration and improve CLI usability. All of the EIGRP-specific interface parameters are configured in the **af-interface default** or **af-interface** *interface-id* submode within the IPv6 address family of the named EIGRP process.

When the IPv6 address family is configured for the EIGRP named process, all the IPv6-enabled interfaces immediately start participating in routing. To disable the routing process on the interface, the interface needs to be shut down in **af-interface** configuration mode.

EIGRPv6 Verification

IPv6 uses the same EIGRP verification commands described in Chapter 3, "Advanced EIGRP," and Chapter 4, " Troubleshooting EIGRP for IPv4." The only modification is that the **ipv6** keyword is included in the command syntax.

Table 5-3 lists the IPv6 versions of the **show** commands that are covered in this chapter.

Table 5-3 EIGRP Display Commands

Command	Description
show ipv6 eigrp interfaces [*interface-id*] [detail]	Displays the EIGRPv6 interfaces.
show ipv6 eigrp neighbors	Displays the EIGRPv6 neighbors.
show ipv6 route eigrp	Displays only EIGRP IPv6 routes in the routing table.
show ipv6 protocols	Displays the current state of the active routing protocol processes.

Figure 5-1 illustrates a simple EIGRP topology in which EIGRPv6 AS 100 is enabled on routers R1 and R2 to provide connectivity between the networks.

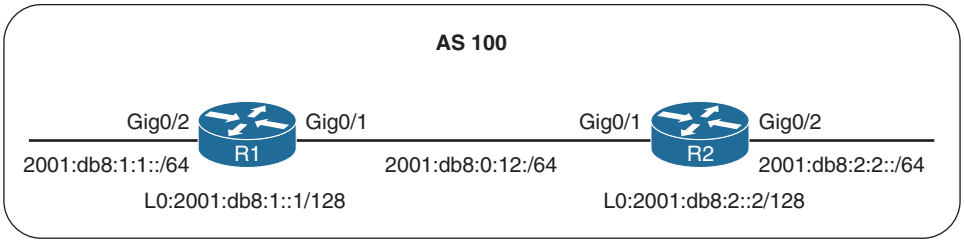

Figure 5-1 *Simple EIGRPv6 Topology*

Example 5-1 shows the full EIGRPv6 configuration for the sample topology. Both EIGRPv6 classic AS and named mode configurations are provided. Notice in IOS classic mode that the routing protocol is applied to each physical interface. In named mode, the protocol is automatically enabled on all interfaces.

Example 5-1 *EIGRPv6 Base Configuration*

```
R1 (Classic Configuration)
interface GigabitEthernet0/1
 ipv6 address 2001:DB8:0:12::1/64
 ipv6 address fe80::1 link-local
 ipv6 eigrp 100
!
interface GigabitEthernet0/2
 ipv6 address 2001:DB8:1:1::1/64
 ipv6 address fe80::1 link-local
 ipv6 eigrp 100
!
interface Loopback0
 ipv6 address 2001:DB8:1::1/128
 ipv6 eigrp 100
!
ipv6 unicast-routing
!
```

```
ipv6 router eigrp 100
 passive-interface Loopback0
 eigrp router-id 192.168.1.1
```

```
R2 (Named Mode Configuration)
interface GigabitEthernet0/1
 ipv6 address 2001:DB8:0:12::2/64
 ipv6 address fe80::2 link-local
!
interface GigabitEthernet0/2
 ipv6 address 2001:DB8:2:2::2/64
 ipv6 address fe80::2 link-local
!
interface Loopback0
 ipv6 address 2001:DB8:2::2/128
!
ipv6 unicast-routing
!
router eigrp NAMED-MODE
 address-family ipv6 unicast autonomous-system 100
  eigrp router-id 192.168.2.2
```

Example 5-2 provides verification of the EIGRPv6 neighbor adjacency. Notice that the adjacency uses link-local addressing.

Example 5-2 *EIGRPv6 Neighbor Adjacency*

```
R1# show ipv6 eigrp neighbors
EIGRP-IPv6 Neighbors for AS(100)
H   Address                 Interface              Hold Uptime   SRTT   RTO  Q  Seq
                                                   (sec)         (ms)       Cnt Num
0   Link-local address:     Gi0/1                    13 00:01:14 1593  5000  0  7
    FE80::2
```

```
R2# show ipv6 eigrp neighbors
EIGRP-IPv6 VR(NAMED-MODE) Address-Family Neighbors for AS(100)
H   Address                 Interface              Hold Uptime   SRTT   RTO  Q  Seq
                                                   (sec)         (ms)       Cnt Num
0   Link-local address:     Gi0/1                    11 00:01:07   21   126  0  5
    FE80::1
```

Example 5-3 shows routing table entries for R1 and R2. Notice that the IPv6 next-hop forwarding address also uses the link-local address rather than the global unicast address of the peer.

Example 5-3 *EIGRPv6 Routing Table Entries*

```
R1# show ipv6 route eigrp
! Output omitted for brevity
D   2001:DB8:2::2/128 [90/2848]
     via FE80::2, GigabitEthernet0/1
D   2001:DB8:2:2::/64 [90/3072]
     via FE80::2, GigabitEthernet0/1
```

```
R2# show ipv6 route eigrp
! Output omitted for brevity
D   2001:DB8:1:1::/64 [90/15360]
     via FE80::1, GigabitEthernet0/1
D   2001:DB8:1::1/128 [90/10752]
     via FE80::1, GigabitEthernet0/1
```

Key Topic

IPv6 Route Summarization

There is no concept of classful or classless routing in IPv6, and therefore, autosummarization is not possible. EIGRPv6 summarization for IPv6 is manually configured on a per-interface basis, using the same rules as for IPv4:

- The summary aggregate prefix is not advertised until a prefix matches it.
- More specific prefixes are suppressed.
- A Null0 route with an administrative distance of 5 is added to the routing table as a loop-prevention mechanism.
- A leak map can be used to advertise more specific prefixes while advertising a summary address.

Network summarization is configured at the interface level in classic mode using the command **ipv6 summary-address eigrp** *as-number ipv6-prefix/prefix-length* or in named mode with the command **summary-address** *ipv6-prefix/prefix-length* under **af-interface**.

Example 5-4 demonstrates how to configure R1 to advertise a 2001:db8:1::/48 summary route to R2 and how to configure R2 to advertise a 2001:DB8:2::/48 summary route to R1. It shows both classic and named mode summary configurations.

Example 5-4 *EIGRPv6 Summary Configuration*

```
R1 (Classic Mode Configuration)
interface GigabitEthernet0/1
 ipv6 summary-address eigrp 100 2001:DB8:1::/48
```

```
R2 (Named Mode Configuration)
router eigrp NAMED-MODE
 address-family ipv6 unicast autonomous-system 100
  af-interface GigabitEthernet0/1
   summary-address 2001:DB8:2::/48
```

5

Example 5-5 shows the routing tables for R1 and R2. Notice that only the /48 summary prefix is received from the neighbor router and that the more specific /64 and /128 route entries are suppressed. A Null0 route is populated on the router for the local /48 summary route advertisement.

Example 5-5 *EIGRPv6 Routing Table Entries*

```
R1# show ipv6 route eigrp
! Output omitted for brevity
D   2001:DB8:1::/48 [5/2816]
     via Null0, directly connected
D   2001:DB8:2::/48 [90/2848]
     via FE80::2, GigabitEthernet0/1
```

```
R2# show ipv6 route eigrp
! Output omitted for brevity
D   2001:DB8:1::/48 [90/2841]
     via FE80::1, GigabitEthernet0/1
D   2001:DB8:2::/48 [5/2816]
     via Null0, directly connected
```

Default Route Advertising

You advertise a default route into the EIGRPv6 topology by placing the default prefix (::/0) as a summary address at the interface level. When you use the summary method, all prefix advertisements are suppressed by the router, except for the ::/0 default route entry.

Example 5-6 demonstrates the two configuration methods for injecting a default route into EIGRPv6.

Example 5-6 *EIGRPv6 Default Route Injection*

```
R2 (Classic Configuration)
interface GigabitEthernet0/1
 ipv6 eigrp 100
 ipv6 summary-address eigrp 100 ::/0
```

```
R2 (Named Mode Configuration)
router eigrp CISCO
 address-family ipv6 unicast autonomous-system 100
  af-interface GigabitEthernet0/1
   summary-address ::/0
```

Route Filtering

In IOS and IOS XE, you use prefix lists to match IPv6 routes in route maps and distribution lists.

Example 5-7 demonstrates how to use a distribution list for filtering the default route ::/0 advertisements from an upstream neighbor connected to interface GigabitEthernet0/1. The associated prefix list BLOCK-DEFAULT with sequence 5 is a deny statement that filters the exact match for the default route prefix ::/0. Sequence 10 is a permit-any match statement that allows a prefix of any length to be received.

Example 5-7 *IOS Distribute List to Filter the Default Route*

```
R1 (Classic Configuration)
ipv6 router eigrp 100
 distribute-list prefix-list BLOCK-DEFAULT in GigabitEthernet0/1
!
ipv6 prefix-list BLOCK-DEFAULT seq 5 deny ::/0
ipv6 prefix-list BLOCK-DEFAULT seq 10 permit ::/0 le 128
```

```
R2 (Named Mode Configuration)
router eigrp CISCO
 address-family ipv6 unicast autonomous-system 100
  topology base
   distribute-list prefix-list BLOCK-DEFAULT in GigabitEthernet0/1
  exit-af-topology
 exit-address-family
!
ipv6 prefix-list BLOCK-DEFAULT seq 5 deny ::/0
ipv6 prefix-list BLOCK-DEFAULT seq 10 permit ::/0 le 128
```


Key Topic

Troubleshooting EIGRPv6 Neighbor Issues

Because EIGRPv6 is based on EIGRP for IPv4, it involves similar issues when it comes to troubleshooting, although there are a few differences for IPv6. Although you do not have to learn a large amount of new information for EIGRPv6, you do need to know the **show** commands that will display the information you need to troubleshoot any given EIGRPv6-related issue.

This section explores the same issues presented in Chapter 4; however, the focus here is on the **show** commands that are used when troubleshooting EIGRPv6-related issues.

The neighbor issues are mostly the same, except for a few differences based on the way EIGRPv6 is enabled on an interface. To verify EIGRPv6 neighbors, use the **show ipv6 eigrp neighbors** command, as shown in Example 5-8. Notice how EIGRPv6 neighbors are identified by their link-local address. In this case, R2 is a neighbor of two different routers. One is reachable out Gi1/0, and the other is reachable out Gi0/0.

Example 5-8 *Verifying EIGRPv6 Neighbors*

```
R2# show ipv6 eigrp neighbors
EIGRP-IPv6 Neighbors for AS(100)
H   Address                 Interface          Hold  Uptime    SRTT   RTO   Q  Seq
                                               (sec)            (ms)       Cnt Num
1 Link-local address:       Gi1/0              10    00:17:59   320   2880  0   4
  FE80::C823:17FF:FEEC:1C
0 Link-local address:       Gi0/0              12    00:18:01   148    888  0   3
  FE80::C820:17FF:FE04:1C
```

Interface Is Down

To verify that an interface is up, you use the **show ipv6 interface brief** command, as shown in Example 5-9. In this example, GigabitEthernet0/0 and GigabitEthernet1/0 are up/up, and GigabitEthernet2/0 is administratively down/down. This indicates that GigabitEthernet2/0 has been configured with the **shutdown** command.

Example 5-9 *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 5-10. In this example, the EIGRP autonomous system is 100.

Mismatched K Values

You verify the EIGRPv6 K values with the command **show ipv6 protocols**, as shown in Example 5-10. In this example, the K values are 1, 0, 1, 0, and 0, which are the defaults.

Passive Interfaces

To verify the router interfaces participating in the EIGRPv6 autonomous system that are passive, you use the **show ipv6 protocols** command, as shown in Example 5-10. In this example, GigabitEthernet 0/0 is a passive interface.

Example 5-10 *Verifying EIGRPv6 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, and if the valid times are configured they must match as well between neighbors. Example 5-11 shows how to verify whether an interface is enabled for EIGRPv6 authentication with the **show ipv6 eigrp interfaces detail** command and how to verify the configuration of the keychain that is being used with the **show key chain** command. In this example, the authentication mode is MD5, and the keychain TEST is being used.

Example 5-11 *Verifying EIGRPv6 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 by using the **show ipv6 eigrp interfaces detail** command, as shown in Example 5-11. In that example, the hello interval is configured as 5, and the hold interval is 15; these are the defaults.

Interface Not Participating in Routing Process

With EIGRPv6, the interfaces are enabled for the routing process with the **ipv6 eigrp** *autonomous_system_number* interface configuration command. You can use two **show** commands, **show ipv6 eigrp interfaces** and **show ipv6 protocols**, to verify the interfaces that are participating in the routing process, as shown in Example 5-12. As with EIGRP for IPv4, the **show ipv6 eigrp interfaces** command does not show passive interfaces. However, **show ipv6 protocols** does.

Example 5-12 *Verifying EIGRPv6 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

Key Topic

EIGRPv6 uses the IPv6 multicast address FF02::A to form neighbor adjacencies. If an IPv6 access control list (ACL) is denying packets destined to the multicast address FF02::A, neighbor adjacencies do not form. In addition, because neighbor adjacencies are formed with link-local addresses, if the link-local address range is denied based on the source or destination IPv6 address in an interface with an IPv6 ACL, neighbor relationships do not form.

Troubleshooting EIGRPv6 Routes

The reasons a route might be missing and the steps used to troubleshoot them with EIGRPv6 are similar to those listed in Chapter 4 for EIGRP for IPv4. This section identifies some of the most common issues and the **show** commands you should use to detect them.

Interface Not Participating in the Routing Process

For a network to be advertised by the EIGRPv6 process, the interface associated with that network must be participating in the routing process. As shown earlier in the chapter, in Example 5-12, 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

Key Topic

If exactly the same network is learned from a more reliable source, it is used instead of the EIGRPv6-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 5-13, the 2001:db8:0:1::/64 network has an AD of 90, and it was learned from EIGRP autonomous system 100.

Example 5-13 *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

Key Topic

A filter might be preventing a route from being advertised or learned. With EIGRPv6, 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 5-14, a distribute list is using a prefix list called ENARSI_EIGRP to filter routes inbound on GigabitEthernet1/0. To successfully troubleshoot route filtering issues, you also need to verify the IPv6 prefix list by using the **show ipv6 prefix-list** command.

Example 5-14 *Verifying EIGRPv6 Distribute List*

```
R1# show run | section ipv6 router eigrp
ipv6 router eigrp 100
 distribute-list prefix-list ENARSI_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 if the wrong setting is chosen during stub router configuration, it might prevent a network from being advertised when it should be advertised. When troubleshooting EIGRPv6 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 5-15. On a remote router, you can issue the **show ipv6 eigrp neighbors detail** command, as shown in Example 5-16. In this case, R1 is a stub router advertising connected and summary routes.

Example 5-15 *Verifying the 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 5-16 *Verifying the EIGRP Stub Configuration of a Neighbor Router*

```
R2# show ipv6 eigrp neighbors detail
EIGRP-IPv6 Neighbors for AS(100)
H   Address                 Interface              Hold   Uptime   SRTT   RTO  Q  Seq
                                                   (sec)           (ms)       Cnt Num
0 Link-local address:     Gi0/0                    11     00:03:35  68    408  0   10
  FE80::C820:17FF:FE04:1C
 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:     Gi1/0                    13     00:14:16  252  1512  0   7
  FE80::C823:17FF:FEEC:1C
 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 on which they were learned. As shown in Example 5-17, you can verify whether split horizon is enabled or disabled by using the **show ipv6 eigrp interfaces detail** command.

Example 5-17 *Verifying the EIGRP Split-horizon Configuration*

```
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 EIGRPv6 network designs that need routes to be advertised out interfaces on which they were learned—either a nonbroadcast multi-access (NBMA) Frame Relay hub-and-spoke topology or a Dynamic Multipoint Virtual Private Network (DMVPN) network, which both use multipoint interfaces on the hub. Therefore, split horizon needs to be disabled on the hub in these networks.

Troubleshooting Named EIGRP

The purpose of EIGRP named configuration is to provide a central location on the local router to perform all EIGRP for IPv4 and IPv6 configuration. Example 5-18 provides a sample named EIGRP configuration called ENARSI_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 and does not cause conflict as these are separate routing processes.

Example 5-18 *Sample Named EIGRP Configuration*

```
Branch# show run | section router eigrp
router eigrp ENARSI_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 EIGRPv6 apply here as well. However, now you need to know which **show** commands can help you successfully troubleshoot named EIGRP deployments. This section covers the **show** commands that you can use to troubleshoot named EIGRP configurations.

With named EIGRP, you can use all the same EIGRP **show** commands that you use for classic EIGRP for IPv4 and classic EIGRPv6, as discussed in Chapter 4 and earlier in this chapter. However, there is also a new set of **show** commands for named EIGRP that you might want to learn.

The command **show eigrp protocols** (see Example 5-19) shows both the EIGRP for IPv4 address family and the EIGRPv6 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 5-19 *Output of show eigrp protocols*

```
Branch# show eigrp protocols
EIGRP-IPv4 VR(ENARSI_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(ENARSI_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
```

5

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** are better options, at least for now.

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 5-20. Note that passive interfaces do not show up in this output. Using the classic **show ip protocols** and **show ipv6 protocols** commands, you would be able to verify the passive interfaces.

Example 5-20 *Verifying Interfaces Participating in the Named EIGRP Process*

```
Branch# show eigrp address-family ipv4 interfaces
EIGRP-IPv4 VR(ENARSI_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(ENARSI_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 5-21, 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).

Key Topic

Example 5-21 *Verifying Details of Interfaces Participating in the Named EIGRP Process*

```
Branch# show eigrp address-family ipv4 interfaces detail
EIGRP-IPv4 VR(ENARSI_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(ENARSI_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
 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 5-22. Just as you saw with the classic commands, if you want to verify whether a neighbor is a stub router, you can add the **detail** keyword to these commands.

Key Topic

Example 5-22 *Verifying Named EIGRP Neighbors*

```
Branch# show eigrp address-family ipv4 neighbors
EIGRP-IPv4 VR(ENARSI_EIGRP) 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                14   00:31:08    88    528  0   8
Branch# show eigrp address-family ipv6 neighbors
EIGRP-IPv6 VR(ENARSI_EIGRP) Address-Family Neighbors for AS(100)
H   Address                 Interface           Hold  Uptime     SRTT   RTO  Q  Seq
                                                (sec)            (ms)       Cnt Num
0 Link-local address:     Fa1/0                14   00:50:33    73    438  0   40
  FE80::C820:17FF:FE04:54
```

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 5-23.

Example 5-23 *Verifying Named EIGRP Topology Tables*

```
Branch# show eigrp address-family ipv4 topology
EIGRP-IPv4 VR(ENARSI_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(ENARSI_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
```

EIGRPv6 and Named EIGRP Trouble Tickets

This section presents various trouble tickets related to the topics discussed earlier in this chapter. These trouble tickets show a process that you can follow when troubleshooting in the real world or in an exam environment.

Trouble Ticket 5-1 is based on the topology shown in Figure 5-2.

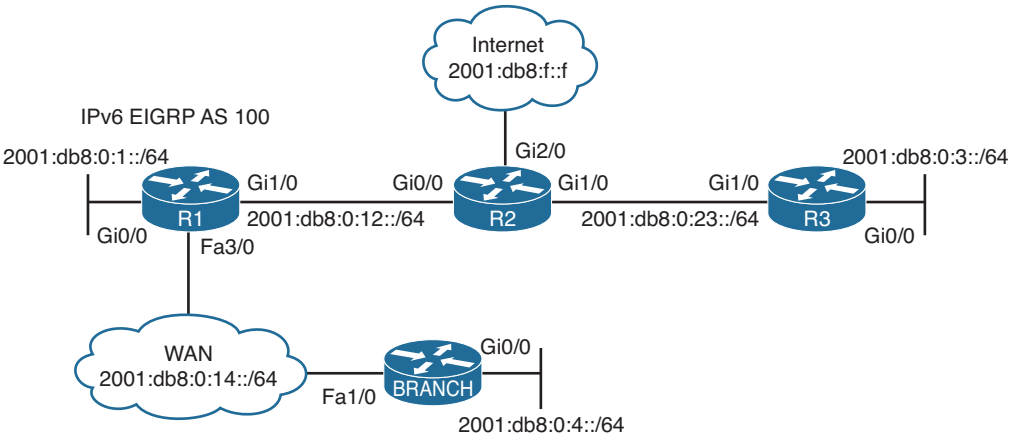

Figure 5-2 *EIGRPv6 Trouble Tickets Topology*

Trouble Ticket 5-1

Problem: Users in the Branch network 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 the source address 2001:db8:0:4::4, as shown in Example 5-24. The ping fails.

Example 5-24 *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 the following snippet, the route is not found:

```
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 5-25, you can see that the Internet address is reachable using a default route (::/0) that was learned through EIGRP.

Example 5-25 *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 output 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 5-26, 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 5-27. The link-local address from Example 5-26 matches the address in Example 5-27.

Example 5-26 *Verifying EIGRPv6 Neighbor Adjacencies*

```
Branch# show ipv6 eigrp neighbors
EIGRP-IPv6 Neighbors for AS(100)
H   Address                 Interface              Hold Uptime   SRTT   RTO  Q  Seq
                                                   (sec)         (ms)       Cnt Num 0
Link-local address:   Fa1/0                  12 00:16:01    63    378  0   16
    FE80::C820:17FF:FE04:54
```

Example 5-27 *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
```